ATMANIRBHAR BHARAT
SWAYAMPURNA GOA

गोंय विद्यापीठ
ताळगांव पठार,
गोंय – ४०३ २०६
फोन : +९१–८६६९६०९०४८

(Accredited by NAAC)

Goa University
Taleigao Plateau, Goa-403 206
Tel : +91-8669609048
Email : registrar@unigoa.ac.in
Website : www.unigoa.ac.in

GU/Acad –PG/BoS -NEP/2024/157 Date: 28.05.2024

Ref: GU/Acad –PG/BoS -NEP/2023/102/28 dated 15.06.2023

CIRCULAR

In supersession to the above referred Circular, the Syllabus of Semester III to VIII of the **Bachelor of Science in Computer Science** Programme approved by the Standing Committee of the Academic Council in its meeting held on 06th, 07th and 21st March 2024 is enclosed. The syllabus of Semester I and II approved earlier is also attached.

The Dean/ Vice-Deans of the Goa Business School and Principals of the Affiliated Colleges offering the **Bachelor of Science in Computer Science** Programme are requested to take note of the above and bring the contents of the Circular to the notice of all concerned.

ASHWIN VYAS LAWANDE
Digitally signed by ASHWIN VYAS LAWANDE
Date: 2024.05.28 15:24:58 +05'30'

(Ashwin Lawande)
Assistant Registrar – Academic-PG

To,

The Principals of Affiliated Colleges offering the Bachelor of Science in Computer Science Programme.

Copy to:

1. The Director, Directorate of Higher Education, Govt. of Goa
2. The Dean, Goa Business School, Goa University.
3. The Vice-Deans, Goa Business School, Goa University.
4. The Chairperson, BOS in Computer Science and Technology.
5. The Controller of Examinations, Goa University.
6. The Assistant Registrar, UG Examinations, Goa University.
7. Directorate of Internal Quality Assurance, Goa University for uploading the Syllabus on the University website.

Programme Structure for Semester I to VIII – Under Graduate Programme – Computer Science										
Semester	Major -Core	Minor	MC	AEC	SEC	I	D	VAC	Total Credits	Exit
I	CSC-100 Computer Organisation (3T + 1P)	CSC-111 E-Commerce (4T) OR CSC-112 Computer Software Fundamentals (4T)	CSC-131 Emerging Trends in Computers (3T) OR CSC-132 Computer Applications (3T) OR CSC-133 Cyber Security Essentials (3T)		CSC-141 PC Troubleshooting (1T + 2P) OR CSC-142 Multimedia and Web Design (1T + 2P) OR CSC-143 Data Analytics using Spreadsheets -I (1T + 2P) OR CSC-144 Desktop Publishing (1T + 2P) OR					

					CSC-145 Basic Computer Applications (1T + 2P)					
II		CSC-113 Digital Marketing (4T) OR CSC-114 Social Media Marketing (4T)			CSC-146 Basics of Python Programming (1T + 2P) OR CSC-147 Graphical User Interface Design (1T + 2P) OR CSC-148 Data Analytics using Spreadsheets -II (1T + 2P) OR CSC-149 Data and Business Analytics (1T + 2P)					CSC-161 Network Creation, Main-tenance and Troubleshooting (4)

					OR CSC-150 Database Management and Analysis (1T + 2P)					
III	CSC-200 Programming using C ++ (3T + 1P) CSC-201 Mathematical Foundations for Computer Science (3T + 1P)	CSC-211 Problem Solving and Programming Concepts (3T + 1P) OR CSC-212 Office Administration (3T + 1P) OR CSC-213 (Computer Organization) (3T + 1P)	CSC-231 Web Designing (3T) OR CSC-232 Application Software for Social Science (3T) OR CSC-233 Application Software for Science (3T) OR CSC-234 Application Software for Literature (3T)		CSC-241 Mobile App Development (1T + 2P) OR CSC-242 Digital Marketing (1T + 2P) OR CSC-243 Embedded Systems (1T + 2P) OR CSC-244 3D Modelling and Animation (1T + 2P)					

			OR CSC-235 Latex (3T) OR CSC-236 3D Graphics Design (3T)		OR CSC-245 Data Management Essentials (1T + 2P)					
IV	CSC-202 Data Structures and Algorithms (3T+1P) CSC-203 Object Oriented Technologies (3T+1P) CSC-204 Operating System (4T)	CSC-221 Introduction to Python Programming (3T + 1P) VET OR CSC-222 Visual Computing (3T + 1P) OR CSC-223 Multimedia Technologies (3T + 1P)								CSC-261 Digital Media Marketing & Analytics (4)

	CSC-205 Operating System Lab (2P)									
V	CSC-300 Database Management System (4T) CSC-301 Computer Networks (3T +1P) CSC-302 Software Engineering (3T+1P) CSC-303 DBMS Lab (2P)	CSC-321 Python for Data Science (3T + 1P) VET OR CSC-322 Image Processing (3T + 1P) OR CSC-323 Statistical Package (3T + 1P)								

VI	CSC-304 Cloud Computing (3T+1P) CSC-305 Foundations of Data Science (3T+1P) CSC-306 Software Quality Assurance (3T+1P) CSC-307 Project (4P)	CSC-324 R Programming for Data Science (3T + 1P) VET OR CSC-325 Computer Graphics (3T + 1P) OR CSC-326 Business Intelligence (3T + 1P)								
VII	CSC-400 Design and Analysis of Algorithms (3T+1P) CSC-401 Artificial Intelligence (3T + 1P)	CSC-411 Ethical Hacking (3T + 1P)								

	CSC-402 Formal Language and Automata Theory (3T + 1P) CSC-403 Network Security (3T + 1P)									
VIII	CSC-404 Machine Learning (4) CSC-405 Internet of Things (4) CSC-406 Introduction to Parallel Computing (4) CSC-407 Research Methodology (3T+1P)	CSC-412 Operations Research (3T + 1P) OR CSC-413 Natural Language Processing OR CSC-414 Introduction to Quantum Computing								

<u>Programme Specific Outcomes</u>

<u>Course: B.Sc. Computer Science</u>

PSO1 – Demonstrate understanding of principles and concepts of various computer science aspects.

PSO2 - Apply problem-solving skills and the knowledge of computer science to solve real world problems.

PSO3 - To enhance skills and adapt to new computing technologies for attaining professional excellence.

Name of the Programme : B.Sc. Computer Science
Course Code : CSC-100
Title of the Course : Computer Organization
Number of Credits : 4 (3T+1P)
Effective from AY : 2024-25 (Revised version of 2023-24)

Pre-requisites for the Course:	Nil	
Course Objectives:	1. Conceptualize the basics of Computer Organizational and Architectural issues and classify the computers based on performance and machine instructions. 2. Learn various data transfer techniques and the I/O interfaces 3. Estimate and compare performances of various classes of memory 4. Understand the basics of ALU implementation, hardwired and microprogrammed control units, pipelining and parallel architectures	
Content:		**No of Hours**
	Unit I: Introduction to Logic Gates and Boolean Algebra: Logic Gates, Boolean Algebra, Combinational circuits, Karnaugh Map Data representation: Data Type Representation, Number System, Signed number, fixed, floating point, character representation, Addition, Subtraction, Multiplication - Shift and Add, Booth's Algorithm, DivisionPseudo-code: Definition and its attributes, constructs, and Examples Memory Hierarchy: Hierarchical memory organization, Types of Memory-internal and external, Cache memory, Memory interleaving,	**15**
	Unit II: Introduction to Computer Architecture: Introduction to Computer Architecture, Flynn's Classification of Computers, Performance Metrics (like Latency, throughput), Fundamental Blocks of Computer (like CPU, I/O subsystems, memory, control unit), computer function, interconnection structures, Bus interconnections Peripheral devices: Types of Peripheral Devices, I/O subsystem,programmed I/O, Interrupt-driven I/O, DMA, I/O channels and processors	**15**
	Unit III: Instruction Set Architecture (ISA): Introduction to Instruction Set, Types of ISA; RISC, CISC; Processor Organization, Registers organization, Instruction Execution Cycle, Instruction formats, Addressing Modes; Register Transfer Language (RTL), Assembly Language Programming, X86-Architecture, ARM Architecture	**15**
	Unit IV: Practical Work Writing assembly language programs in 8086 using MASM or compatible assembler either in Windows or Linux.	

	Week 1 & 2 a. Introduction to 8086 architecture and instruction set b. Find the sum of 1 + 2 + 3 + ……………+ n	**04**
	Week 3 & 4 c. Display the multiplication table of a number d. Store and retrieve numbers from memory	**04**
	Week 5 & 6 e. Block Transfer f. Block Transfer in reverse order	**04**
	Week 7, 8 & 9 g. Sort the numbers stored in the memory (Any two.methods) h. Searching methods	**06**
	Week 10 & 11 i. Masking of bits j. Counting of number of bits	**04**
	Week 12 & 13 k. Count the number of even or odd numbers from a given set of numbers l. Check if the number is a palindrome	**04**
	Week 14 & 15 m. Count the number of positive and negative numbers from a given set of numbers n. Generate a series like 1,3,5,7,…….. up to n terms	**04**
Pedagogy:	Suggested strategies for use to accelerate the attainment of the various course outcomes. Lectures, Tutorials, Collaborative/peer learning, Hands-on assignments	
References/ Readings:	Main Reading: 1. William Stallings(2022), *“Computer Organization and Architecture: Designing for performance”*, Pearson Education India. Additional Reading: 2. John L. Hennessy & David Patterson(2017)*“Computer Architecture: A Quantitative Approach”*, Morgan Kaufmann, 6thEdition,	
Course Outcomes:	On completion of the course, students will be able to: 1. Explain the theory and architecture of central processing unit, I/O and memory organization 2. Understand the concepts of parallel processing, pipelining and inter-processor communication. 3. Represent different number systems, and perform various binary operations. 4. Analyze some of the design issues in terms of speed, technology, cost, performance, CPU architecture.	

Name of the Programme : **B.Sc. Computer Science**
Course Code : **CSC-111**
Title of the Course : **E-Commerce**
Number of Credits : **4T**
Effective from AY : **2023-24**

Pre-requisites for the Course:	Nil	
Course Objectives:	This Course aims - 1. To develop an understanding of Web-based Commerce 2. To equip students to assess-commerce requirements of a business 3. To enable students to develop - business plans and e-commerce applications	
Content:		**No. of Hours**
	1. Introduction to Electronic Commerce & E-Commerce Web-sites Meaning, Nature and scope of e-commerce,History of e-commerce, Business applications of e-commerce, E-CommerceModels:-(B2B,B2C,C2C,B2G),Advantages and Disadvantages of e-commerce, Applications of M-Commerce, Websites as marketplace, Role of website in B2C e-commerce, Website design principles, Alternative methods of customer communication such as e-mail, Email etiquette and e-mail security	**15**
	2. Online Marketing & Applications of E-commerce Online marketing and advertising, Push and pull approaches, Web counters, Web advertisements, Content marketing, Need of Digital Marketing for an e-commerce Business, Search Engine Optimization(SEO),Search Engine Marketing(SEM),Social Media Marketing(SMM), Web Analytics Applications of e-commerce to Supply chain management Applications of e-commerce to Customer Relationship Management, Product and service digitization, Remote servicing	**15**
	3. Business to Consumer, Business to Business E-Commerce Applications Cataloging, Order planning and order generation, Cost estimation and pricing, Order receipt and accounting, Order selection and prioritization, Order scheduling, Order fulfilling, Order delivery, Order billing, Post sales service. Need and Models of B2B e-commerce, Using public and private computer networks for B2B trading; EDI and paperless trading, Characteristic features of EDI service arrangement, EDI architecture and standards, Reasons for slow acceptability of EDI, Value Added Networks	**15**
	4. Electronic Payment System & Security Issues in E-Commerce	**15**

	Types of payment systems, credit cards, debit cards, mobile all etc., Electronic Fund TransferEFT),Operational credit and legal risk of e-payment, Risk management options for e-payment systems. Risks of e-commerce, Types and sources of threats to e-commerce; Protecting electronic commerce assets and intellectual property, Firewalls, Client server network security, Security Protocols–SSL,SET,S-HTTP, Data and message security, Security tools, Digital identity and electronic signature, Encryption and concept of public and private key infrastructure; Risk management approach to ecommerce security	
Pedagogy:	PowerPoint presentations, Case studies	
References/ Readings:	Main Reading: 1. Kamalesh N. Agarwala, Amit Lal, Deeksha Agarwala(2000),*“Business on the Net: An Introduction to the Whats’ and Hows’ of E Commerce”*, Macmillan India Ltd. 2. Parag Diwan, Sunil Sharma(2002), *“Electronic Commerce- A Manager's Guide to E Business”*, Excel Books, India Additional Reading: 1. Jerry FitzGerald, Alan Denis, Alexandra Durcikova(2001), “*Business Data Communication and Networking”*, Wiley Publisher. 2. Ravi Kalakota, Andrew B. Whinston(1996), “*Frontiers of Electronic Commerce*”, Pearson Education India. 3. Dishek J. Mankad(2019), “*Understanding Digital Marketing: Strategies for online success*”, BPB Publications.	
Course Outcomes:	On completion of the course students will be able to:- 1. Describe the basics of e-commerce. 2. Explain the design principles of e-commerce websites. different models of e-commerce. 3. Describe the different electronic payment systems. 4. Explain the security issues, security mechanism and threats to e-commerce applications.	

Name of the Programme : B.Sc. Computer Science
Course Code : CSC-112
Title of the Course : Computer Software Fundamentals
Number of Credits : 4T
Effective from AY : 2023-24

Pre-requisites for the Course:	Nil	
Course Objectives:	This course will enable the student to learn 1. To understand the fundamentals of IT, software and Networking 2. To learn the usage of IT, software and networking 3. To familiarize with latest trends in IT	
Content:		**No. of Hours**
	Unit I: Introduction of IT • **Data:** Definition, Types of data. Data Representation: Character formats-ASCII, Unicode (Definition, Adding regional languages, Phonetic keyboards. • Number system:Binary, decimal, Conversion. Data Organization Directory structure, File formats and Compression (Text, Audio, Image,Video). DataBackup:Techniques,Scheduler,Onlinebackup,Advantages. Device Interfaces and Data Storage: Data device Interface access methods (USB, IDE /SATA). Optical memory (Blue ray), Flash memory (USB Sticks, Memory Cards, SD, MMC, Micro SD), Magnetic Memory (External disks), New Devices (Solid state drives) • **Information**:Prerequisites of Information, Need for Information Technology and its advantages, QualitiesofInformation • **Information Technology:**Definition and components. Basic computerOrganization Application of IT (Scienceand Engineering,Business & Commerce, Education, Governance, Medicine, Entertainment)	**15**
	Unit II: Software:System and Applications • RelationshipbetweenHardwareandSoftware • ProgrammingLanguages:Low level, High level,Translators • SystemPrograms:Operatingsystems:Operating systems • Definition and functions of operating system, Examples of operating System (Windows, Linux, Online OS, Virtual OS, Comparison), Multi boot systems (disk partitions and logical drives) Directory Structure: System directories, Users (administrator, limited rights user and guest), User directories (directory permission) • Services,drives and hardware interfaces • ApplicationPrograms:Definition,ExamplesIntroductionto Mathematical Computation Packages Human computer interaction (HCI)	**15**

	Unit III: Basics of Computer Networking and data communication • Networking basics, why networking of computers is needed, • Types of networks-LAN, MAN, WAN, • Network Components – H/W, Software, Network Devices, Network topologies. • Communication channels, Communication Process, Data Transmission speed, Communication Types (modes) • Internet – role and importance, IP Addressing – public Vs Private, Static VsDynamic; WWW & related protocols;	**15**
	Unit IV: Future IT trends • Artificial Intelligence (AI) and Automation (Definition, Applications) • IoT and Edge Computing • Cloud Infrastructure • Virtual Reality & Augmented Reality • Business Intelligence	**15**
Pedagogy:	Lecture method using ICT tools	
References/ Readings:	Main Reading: 1. Anita Goel (2010), "*Computer Fundamentals*",Pearson Education. Additional Reading: 1. P.Aksoy, L.DeNardis(2006), "*Introduction to Information Technology*", CengageLearning, 2006 2. Pradeep K.Sinha,Priti Sinha(2007), "*Computers Fundamentals*", BPB Publishers	
Course Outcomes:	At the end of the course,learner will be able to: 1. Remember basics of IT, software, networking, trends in IT 2. Understand various I/O devices, systems, networking devices, IT uses 3. Apply the concepts in systems, devices, networking for IT 4. Analyse the applications of IT, Software, Networking and trends in IT	

Name of the Programme	**: B.Sc. Computer Science**
Course Code	**: CSC-113**
Title of the Course	**: Digital Marketing**
Number of Credits	**: 4 (4 Theory)**
Effective from AY	**: 2023-24**

Pre-requisites for the Course:	Nil	
Course Objectives:	1. To acquaint the students with basic principles and concepts of digital marketing & advertising 2. To understand and familiarize the students with the concept of Digital Marketing techniques like Adwords, search advertising, display advertising. 3. To understand the concept of Search Engine Optimization (SEO)	
Content:		**No. of Hours**
	1. Fundamentals of Digital Marketing & Ad Words Fundamentals Marketing in the digital world; Integrated marketing- The Phygital; Global trends inDigital Marketing; Digital channels- Paid, Owned and Earn; Fundamentals on the primary asset- yourwebsite; Careers in digital marketing; Skill development in Digital marketing **Ad Words Fundamentals:** Understanding Pay-per-click Advertisement; Significance and evolution of Ad Words in PPC Bing Ads V/s Google Ads- overview; AdWords Certification- Overview, Benefits and Preparation; Google Ad Networks; Different Ad Formats; Keywords - significance and planning; Using Keyword Planner and other tools; Keyword matches and their usage; Campaign Structure and Organization Quality, Rank and Relevance of Ads; Bidding and budget; Targeting Setting Extensions and their usage; Ad policies and approvals; Reports and Analysis Metrics; Conversion Tracking; Campaign Optimization	**15**
	2. Search & Display Advertising with Adwords Search with Adwords Keywords - planning, matching and combination; Specifications of an Ad and how to put it to good use; Managing Invalid Clicks; Ad extensions and usage; Dynamicsearch ads; Landing page - your virtual front; Campaign Experiment; Opportunities Tab; AdWords APIs; Ad Words editor-Benefits and usage; Managing multiple accounts Display with Adwords Google Display Network and Partnerships; Double click Ad Exchange and AdSense Campaign Creation and Structuring for display; Keyword and targeting through display network; Campaign Metrics, Analysis and optimization	**15**
	3. SEO Basics How search engines work; Different Search results and significance; Query types and significance; What is SEO and key	**15**

	factors determining the same; Components on SEO-onsite and off page; Keyword Planning; Using tools to get effective keywords; Long tail keywords-the hidden gems; Art and science of tags-URL, title, meta, H1, alt text, etc.; Write a good meta description; Page speed - its impact and improvement areas; All about links- broken, internal et al; Dealing with duplicate content; Robot. Txt and Sitemap; Structured data and schema.org	
	4. SEO Advance Concepts Link building basics; Avoiding harmful links; Finding and leveraging link building opportunities; Creating a link building plan; Major Google updates and their implication son SEO; Using Search Console for SEO; KPIs of SEO; Tools for SEO; Moz SEO Products; SEM rush Competitive Research and Business Intelligence Software; Competition Analysis for SEO; Overall planning for SEO; Understanding nuances of local and international SEO; Accelerated mobile pages and SEO; Artificial Intelligence,Voice search and SEO–what to look forward	**15**
Pedagogy:	PowerPoint presentations, Case studies	
References/ Readings:	Main Reading: 1. Dave Chaffey & Fiona Ellis-Chadwick(2015), “*DigitalMarketing: Strategy, Implementation and Practice*”, 6th Edition, Pearson Education Additional Reading: 1. Ekaterina Walter(2014), “*The Power of Visual Storytelling*”, 1st Edition, McGraw-Hill Education 2. Ben Hunt(2011), “*Convert!: Designing Websites For Traffic and Conversions*”, 1st Edition, John Wiley &Sons 3. Lon Safko(2014), “*The Social Media Bible: Tactics, Tools, & Strategies for Business Success*”, Brilliance Audio; Unabridged Edition 4. Pam Didner(2014), “*Global Content Marketing*”, McGraw-Hill Professional, Illustrated Edition 5. Joe Pulizzi(2015), “*Content Inc.: How Entrepreneurs Use Content to Build Massive Audiences and Create Radically Successful Businesses*”, 1st Edition, McGraw-Hill Education 6. Corey Rabazinski(2015), “*Google Adwords for Beginners: A Do-It-Yourself Guide to PPC Advertising*”, 1st Edition, Create Space Independent Publishing Platform	
Course Outcomes:	At the end of the course learner will be able to 1. Understanding of digital landscape and building a case to leverage online channels 2. Apply Strategize, implement and optimize online campaigns successfully 3. Develop and design Online Advertising campaigns, AdWords Campaign management and Campaign Basics across search. 4. Apply advance concept of Search Engine Optimization to capture the right intent and drive organic traffic.	

Name of the Programme	**: B.Sc. Computer Science**
Course Code	**: CSC-114**
Title of the Course	**: Social Media Marketing**
Number of Credits	**: 4 (4 Theory)**
Effective from AY	**: 2023-24**

Pre-requisites for the Course:	Nil	
Course Objectives:	1. To understand the concept of Social Media Marketing platform. 2. To understand video and mobile platform advertising. 3. To understand and apply the concept of web and google analytics. 4. To acquire understanding of LinkedIn, Twitter, Pinterest Marketing 5. To Measure, Analyze and Optimize Social Media Marketing Campaigns 6. To create an effective Digital Marketing Plan.	
Content:		**No. of Hours**
	1. Introduction to Social Media Marketing Evolution and importance of Social Media ;What social media can do for you?; Different social media platforms;Unwritten rules of Social Media ;Facebook for business; Using of Facebook groups, pages and events; Using of Facebook tabs and apps;Running Facebook ads;AdManager and Power Editor in Facebook;Targeting –the structured approach; Facebook page Insights	**15**
	2. YouTubeVideoandMobileAdvertising YouTube - why do you need to be there?; YouTube format, tools & targeting; Video Campaign Creation;Video Campaign track and optimization;Video Ad performance & best practices; YouTube Analytics.Importance of Mobile and Opportunities to Leverage; Key Objectives for Mobile Marketing;AdFormats and Networks for Mobile;Mobile Site: Key Considerations; Mobile App :Key Considerations;Mobile specific bidding and targeting;Apps Marketing,Mobile Analytics,Reporting and Optimization	**10**
	3. Media Marketing with Twitter, LinkedIn, Instagram & Snapchat Introduction to Twitter and its terminologies; Creating a good Twitter profile; Building followers on Twitter;Using Twitter Chats;Twitter as an influencer marketing tool; Twitter ads; Twitter Analytics; LinkedIn for Business; Profile, pages and Pulse in LinkedIn; LinkedIn Ad; LinkedIn Analytics; B2B marketing using LinkedIn; Introduction to Pinterest for Business; Pinterest strategies; Instagram for business; Instagram strategies;New kid on the block–Snapchat;Online Reputation Management ;Social media tool sand how to use them;Creating social media calendar and workflow	**15**
	4. WebAnalytics Introduction to web analytics; How web analytics works, Analytics	**20**

	Framework; Goals, Objectives and KPIs; Contextualizing of Data; Segmentation of Data; Making analytics actionable; Attribution Modeling; URL tracking and UTM builder; Clickstream, HeatMap and other forms of WebAnalytics; A/B testing **Google Analytics** How Google Analytics (GA) work; Dimensions, metrics and other common terminologies; Setting up Google analytics; Tracking, Reports and Dashboards; Acquisition, Behaviour and Conversion; Visitors Analysis; Source and Medium analytics; Conversion tracking; Content Performance Analytics; Userflow; Leveraging real time analytics; Content Experiment; Linking Search Console and AdWords with Google Analytics; Intro to Google Data Studio	
Pedagogy:	PowerPoint presentations, Case studies	
References/ Readings:	Main Reading: 1. Dave Chaffey & Fiona Ellis-Chadwick(2015), “*Digital Marketing: Strategy, Implementation and Practice*”, 6th Edition, Pearson Education Additional Reading: 2. Ekaterina Walter(2014), “*The Power of Visual Storytelling*”, 1st Edition, McGraw-Hill Education 3. Ben Hunt(2011), “*Convert!: Designing Websites For Traffic and Conversions*”, 1st Edition, John Wiley &Sons 4. Lon Safko(2014), “*The Social Media Bible: Tactics, Tools, & Strategies for Business Success*”, Brilliance Audio; Unabridged Edition 5. Pam Didner(2014), “*Global Content Marketing*”, McGraw-Hill Professional, Illustrated Edition 6. Joe Pulizzi(2015), “*Content Inc.: How Entrepreneurs Use Content to Build Massive Audiences and Create Radically Successful Businesses*”, 1st Edition, McGraw-Hill Education 7. Corey Rabazinski(2015), “*Google Adwords for Beginners: A Do-It-Yourself Guide to PPC Advertising*”, 1st Edition, Create Space Independent Publishing Platform 8. Seth Godin(2012), “*All Marketers Are Liars*”, Portfolio, Reprint Edition 9. JayBaer(2013), “*Youtility: Why Smart Marketing Is About Help Not Hype*”, Portfolio, Edition Unstated 10. Russell Glass & Sean Callahan(2015), “*The Big Data- Driven Business*”, 1st Edition, Wiley 11. Damian Ryanand Calvin Jones(2008), “*Understanding Digital Marketing: Marketing Strategies for Engaging the Digital Generation*”, 1st Edition, KoganPage 12. Ryan Deiss and Russ Henneberry(2017), “Digital Marketing for Dummies”, 1st Edition, John Wiley and Sons	
Course Outcomes:	On completion of the course learner will be able to: 1. Remember the basics of Social Media Marketing. 2. Understand the use of mobile and video media for online advertising, & AdWords campaign management. 3. Apply Twitter, LinkedIn, Instagram & similar media for promotion. Tools and concepts to execute measure and monitor an annual online marketing plan and use analytics to drive action able improvements 4. Design digital marketing techniques into strategic marketing plan	

Name of the Programme : **B.Sc. Computer Science**
Course Code : **CSC-131**
Title of the Course : **Emerging Trends in Computer**
Number of Credits : **3T**
Effective from AY : **2023-24**

Pre-requisites for the Course:	NIL	
Course Objectives:	This course will – 1. enable students to explore current breakthrough technologies in the areas of Artificial Intelligence (AI), Big data and Business Intelligence, IOT, Blockchain that have emerged over the past few years. 2. prepare the students to use technology in their respective professional preparations.	
Content:		**No. of Hours**
	Unit 1: Artificial Intelligence & Business Intelligence (BI) and Big data AI Concept, Scope of AI, Components of AI, Types of AI, Machine Learning (ML) and Natural Language Processing (NLP), Applications of AI, the state of art AI today BI- Definition, Importance, Benefits of Business Intelligence, How BI process works, Stages of Business Intelligence. Big data – Definition, Characteristics, Challenges with Big Data, Traditional Business Intelligence (BI) versus Big Data. Big Data Applications in Business	**15**
	Unit2: Internet of Things (IoT) and Embedded Systems Definition, Characteristics of Embedded System, Real time systems, Real time tasks. Processor basics: General Processors in Computer Vs Embedded Processors, Microcontrollers, Microcontroller Properties, Components of Microcontrollers, Components of Embedded Systems, Introduction to embedded processor Definition, Characteristics of IoT, Trends in Adoption of IoT, IoT Devices, IoT Devices Vs Computers, Societal Benefits of IoT, Technical Building Blocks. IoT functional blocks, IoT enabling technologies, IoT levels and deployment templates, Applications in IoT.	**15**
	Unit 3: Cloud Computing & Blockchain and Cryptocurrency Importance of Cloud Computing, Characteristics, Pros and Cons of Cloud Computing, Migrating into the Cloud, Seven-step model of migration into a Cloud, Trends in Computing. Cloud Service Models: SaaS, PaaS, IaaS, Storage, Cloud Architecture: Cloud Computing Logical Architecture, Developing Holistic Cloud	**15**

	Computing Reference Model, Cloud System Architecture, Cloud Deployment Models. Introduction to Blockchain Technology and its Importance, Evolution of the Blockchain Technology, Elements of a Blockchain A basic crypto currency, Creation of coins, Payments and double spending, Bitcoin –Digital Signatures as Identities – eWallets – Personal Crypto security - Bitcoin Mining	
Pedagogy:	PowerPoint, YouTube Videos	
References/ Readings:	Main Reading: 1. Stuart Russel and Peter Norvig (2015), "*Artificial Intelligence: A Modern Approach*", 3rd Edition, Pearson 2. V.K Jain (2018), "*Big Data and Hadoop*", 2nd Edition, Khanna Publishing 3. Tejaswini N and Yathish R(2019), "*Blockchain for Beginners: The Art of Decentralization & Cryptography*", 1st Edition, Shroff/X-team 4. Cuno Pfister(2011), "*Getting Started with the Internet of Things*", 1st Edition, Make Community 5. ArsheepBahga, Vijay MAdisetti(2015), "Internet of Things: A Hands-On Approach", 1st Edition, Orient Blackswan Private Limited - New Delhi 6. Anandamurugan, T.Priyaa, M.C. Arvind Babu(2017), "Cloud Computing", 1st Edition, Laxmi Publications Pvt. Ltd.	
Course Outcomes:	At the end of the course, students will be able to: 1. Remember different emerging technologies 2. Define emerging trends in Computer Science 3. Select appropriate technology for a given task 4. Identify necessary inputs for applications of emerging technologies	

Name of the Programme	**: B.Sc. Computer Science**
Course Code	**: CSC-132**
Title of the Course	**: Computer Applications**
Number of Credits	**: 3T**
Effective from AY	**: 2023-24**

Pre-requisites for the Course:	Nil	
Course Objectives:	1. To provide an understanding of essential Information Technology concepts 2. To familiarize and learn use of various types of IT tools	
Content:		**No. of Hours**
	Unit 1: (Computer Basics) Introduction to computers – Definition, Characteristics, Classification of computers, Components of a Computer System –Hardware Components - Central Processing Unit, Input devices, Output devices, Computer Memory. Categories of Software - System Software and Application Software, Operating Systems - definition and functions. Data - Definition, Types, Data Representation, Types of Number system- Binary, Octal, Hexadecimal Conversion between number bases	**8**
	Unit 2: (Word Processor) Word processing concepts: Use of Templates, Working with word document: Editing text, Find and replace text. Formatting- Text, Paragraphs, Styles, Columns. Bullets and numbering, Tabs, Indent, Page Formatting. Design Themes, Page Background. Page setup Insert: Tables, Illustrations, Links, Comments, Header and Footer, Symbols. Tables: Inserting, filling and formatting a table, Changing cell width and height, Alignment of Text in cell, Delete / Insertion of Row, Column and Merging & Splitting of Cells, Border and Shading. Referencing- Captions, Footnotes and Endnotes Citations and Bibliography, Reference Tables and Indexes, Bookmarks and Cross-References.	**10**
	Unit 3: Spreadsheets Spreadsheet concepts: Managing worksheets; Formatting, Conditional formatting, Entering data, Editing, Handling operators in formula, Project involving multiple spreadsheets, Organizing Charts and graphs, Generally used Spreadsheet functions: Mathematical, Statistical, Financial, Logical, Date and Time, Lookup and reference, Database, and Text functions, Summarizing data using filter. Pivot tables to analyze data. Using What-If Scenario Manager, Goal Seek. Printing a worksheet-working with page breaks, adding headers	**10**

	or footers, choosing what to print.	
	Unit 4: Presentation Software Creating a presentation, creating a Presentation Using a Template, Creating a Blank Presentation, Inserting & Editing Text on Slides, Inserting and Deleting Slides in a Presentation, Saving a Presentation, Manipulating Slides, Inserting Table, Adding ClipArt Pictures, Inserting Other Objects, Resizing and Scaling an Object, Creating & using Master Slide, Presentation of Slides, Choosing a Set Up for Presentation, Running a Slide Show, Transition and Slide Timings, Automating a Slide Show, Providing Aesthetics to Slides & Printing, Enhancing Text Presentation, Working with Color and Line Style, Adding Movie and Sound, Adding Headers, Footers and Notes, Printing Slides and Handouts.	**10**
	Unit 5: User Generated Content Blogs and Wikis. Online Data Capture Tools: Types of data capture form templates (Personal, Work and Education). Question Formats for data capture (short answer, paragraph, multiple choice, check- box, drop-down, linear-scale, multiple choice grid). Data form design (Add new question, add section, add title/description/image/video). Data form distribution techniques (Send via email, publish on social media, send as link). Response management (Print responses, Export to spreadsheet, View analysis, Include analysis in word processing reports)	**7**
Pedagogy:	PowerPoint, Tutorials	
References/ Readings:	Main Reading: 1. Dennis Curtin, Kim Foley, Kunal Sen, Cathy Morin(2017), "*Information Technology The breaking wave*", Indian Edition, McGraw-hillEducation Additional Reading: 1. ITL Education Solutions Limited(2012), "*Introduction to Information Technology*", second edition, Pearson Education India. 2. Satish Jain, Shashank Jain, Shashi Singh & M. Geetha Iyer (2010), *"O*" *Level made simple "Introduction to ICT resources*", BPB publication. 3. Pradeep K. Sinha and Priti Sinha(2004), "*Computer fundamentals*", 4th Edition, BPB publications	
Course Outcomes:	At the end of the course the learner will be able to: 1. Understand the essential of Information Technology Concepts 2. Develop practical skills in data capture, analysis and presentation, report formatting 3. Use a range of current, standard, Office Productivity software applications 4. Apply the basic concepts of a word processing package, electronic spreadsheet and PowerPoint tool	

Name of the Programme : B.Sc. Computer Science
Course Code : CSC-133
Title of the Course : Cyber Security Essentials
Number of Credits : 03 T
Effective from AY : 2023-24

Pre-requisites for the Course:	The student should have basic knowledge on how to use computers and internet technology.	
Course Objectives:	1. To introduce principles of cyber security and have an understanding on the cyber-crimes taking place. 2. To have an understanding of the existing legal framework and laws on cyber security. 3. To enable students to adopt safe practices when using social media platforms and digital payment systems.	
Content:		**No. of Hours**
	1. Introduction to Cyber security& Cyber crime and Cyber law Defining Cyberspace and Overview of Computer and Web-technology, Architecture of cyberspace, Communication and web technology, Internet, World Wide Web, Advent of internet, Internet infrastructure for data transfer and governance,Internet society,Regulation of cyberspace, Concept of cyber security, Issues and challenges of cyber security. Classification of cyber-crimes, Common cyber-crimes, cyber-crime targeting computers and mobiles, cyber-crime against women and children, financial frauds, social engineering attacks, malware and ransomware attacks, zero day and zero click attacks, Cybercriminals modus-operandi, reporting of cyber-crimes, Remedial and mitigation measures, Legal perspective of cyber-crime, IT Act 2000 and its amendments, Cyber-crime and offenses, Organizations dealing with Cybercrime and Cyber security in India, Case studies.	**15**
	2. Social Media Overview and Security Introduction to Social networks. Types of Socialmedia, Social media platforms, Social media monitoring, Hashtag, Viralcontent, Social media marketing, Social media privacy, Challenges, Opportunities and pitfalls in online social network, Security issues related to social media, Flagging and reporting of inappropriate content, Laws regarding posting of inappropriate content,Best practices for the use of Social media,Case studies.	**10**
	3. E-Commerce and Digital Payments Definition of E- Commerce, Main components of E-Commerce, Elements of E-Commerce security, E-Commerce threats, E-Commerce security best practices, Introduction to digital payments,Components of digital payment and stakeholders, Modes of digital payments- BankingCards, Unified Payment Interface (UPI),e-Wallets, Unstructured SupplementaryService	**10**

	Data (USSD), Aadhar enabled payments, Digital payments related common frauds and preventive measures.RBI guidelines on digital payments and customer protection in unauthorized banking transactions. Relevant provisions ofPayment Settlement Act,2007.	
	4. Digital Devices Security, Tools and Technologies for Cyber Security End Point device and Mobile phone security, Password policy, Security patch management, Data backup, Downloading and management of third-party software, Device security policy, Cyber Security best practices, Significance of host firewall and Ant-virus, Management of host firewall andAnti-virus, Wi-Fi security, Configuration of basic security policy and permissions.	**10**
Pedagogy:	Lecture method, Case Studies, Hands-on Training, Group Discussions	
References/ Readings:	Main Reading: 1. R. C Mishra(2010), "*Cyber Crime Impact in the New Millennium*", Authors Press 2. Nina Godbole, SunitBelapure (2011), "*Cyber Security Understanding Cyber Crimes, Computer Forensics and Legal Perspectives*", 1st Edition, Wiley India Pvt. Ltd. Additional Reading: 1. Henry A. Oliver (2015), "*Security in the Digital Age: Social Media Security Threats and Vulnerabilities*", Create Space Independent Publishing Platform. 2. Elias M. Awad(2002), "*Electronic Commerce*", Prentice Hall of India Pvt Ltd. 3. Krishna Kumar(2011), "*Cyber Laws: Intellectual Property & E-Commerce Security*", Dominant Publishers and Distributors 4. Eric Cole, Ronald Krutz, James W. Conley (2005), "*Network Security Bible*", 2ndEdition, Wiley India Pvt. Ltd. 5. EricMaiwald(2017),"*Fundamentals of Network Security*", McGraw Hill Education.	
Course Outcomes:	At the end of the course, learner will be able to: 1. Remember the concept of Cyber Crime & Cyber security and issues and challenges associated with it. 2. Understand the nature of cyber crimes, legal remedies and as to how to report the crimes through available platforms and procedures. 3. Explain various privacy and security concerns on online social media and the reporting procedure of inappropriate content, underlying legal aspects and best practices for the use of Social media platforms. 4. Explain the basic concepts related to E-Commerce and digital payments, digital payment modes and related cyber security aspects, RBI guidelines and preventive measures against digital payment frauds.	

Name of the Programme : **UG Degree (Honors) with Computer Science**
Course Code : **CSC-141**
Title of the Course : **PC Troubleshooting**
Number of Credits : **3 (1 T + 2 P)**
Effective from AY : **2023-24**

Pre-requisites for the Course:	Nil	
Course Objectives:	To make the students capable of understanding the functioning of hardware parts and develop skills in diagnosing the faults and troubleshooting the computer system.	
		No. of Hours
Content:	1. **Hardware Basics** Basic terms, concepts, and functions of system modules (System board,firmware, storage devices, monitor, boot process, ports).CMOS and BIOS,Motherboard, SMPS	**3**
	2. **Memory Module and Hard disk** Different types of Memory Modules, Tips on installing memory chips.Disk structure: Cylinders, heads, platters, tracks and sectors, structure of a disk, hard disk controllers.Types of interface controller and drives.	**4**
	3. **Input/OutputDevices** Keyboard and Mouse, Scanner and its types, CD-ROM Drives, Monitors: Display basics, Display adapter cards, VGA and super VGA, Printer: Types, Interfaces, Connection to Computers.	**4**
	4. **TroubleshootingandPreventive Maintenance** Troubleshooting basics,Troubleshooting by visual Inspection,PreventativeMaintenance. POST: Functions, Test Sequence, Error messages, Troubleshooting Procedures andPreventative Maintenance. Power Supply and UPS.	**4**
Pedagogy:	PowerPoint Presentations, Hands on	
References/ Readings:	Main Reading: 1. B.Govindarajalu(2002), “*IBM PC & Clones: Hardware Troubleshooting and Maintenance*”, 2^{nd} Edition, Tata McGraw Hill Additional Reading: 1. Marcia Press, Barry Press(2004), “*PC Upgrade & Repair Bible*”, Wiley India. 2. D Balasubramanian (2005), “*Computer Installation and Servicing*”, 2^{nd} Edition, McGraw Hill Education	
Course Outcomes:	At the completion of this course, the student will be able to: 1. Explain Basics of Hardware Components. 2. AcquireknowledgeofFindingFaultsinComponents 3. Install, Configure and maintain various components in computer systems and peripherals.	

	4. Diagnose faults, repair and maintain computer systems and its peripherals & different components.

List of Experiments:
(Perform at least 10 experiments from the list given below)

Sr.No.	Name of the Experiment (60 hours)
1	Disassemble the PC carefully. Assemble the same PC you have disassembled and boot the system.
2	Observe various connectors, ports back and front side of the computer and write their purpose. (e.g. Power, PS/2keyboard and mouse, Serial and parallel, USB, VGA, LAN, Audio & microphone, Firewire, HDMI, games, SATA etc.)
3	Identify the on-board features of the motherboard like network capabilities, and gaming capabilities. Install the given driver and test the computer for proper functioning. Remove the drivers for some devices like sound, display, network etc. and again install them and check the proper functioning of the computer. Upgrade the given PC by adding RAM and additional HardDisk.
4	Observe the power supply (SMPS) and measure their voltage levels of a given SMPS. Measure various voltage levels, such as motherboard, storage devices and fan etc. using a multimeter. Do a detailed study on all the components and devices on the given power supply. Observe different types of Switch Mode Power Supply – AT, ATX, NLX. Record the different types of power connectors on the motherboard.
5	Identify BIOS settings, demonstrate starting BIOS, identify how to disable unused devices to decrease security risks. Change booting of computer with different secondary storage CD,HDD,USB, etc.
6	Perform low-level and high-level formatting of Hard Disk. Format the given Hard Disk using any one technique and create three partitions, two for operation systems and one for data.
7	Install OS of different types (Windows and Linux). Also,search for various data recovery software on pendrive/HDD.
8	Open different types of keyboards and mouse and observe the internal circuits. Observe and write steps to troubleshoot, maintain and clean the keyboard and mouse.
9	Observe different types of printers. Install driver and interface the printers with PC/Laptop on any operating system (connect the printer to one PC directly using USB/Serial/Parallel/Wi-Fi as per the availability; test the functioning of the printer.)
10	Learn the interfacing, installation and working of various devices such as scanner, projector, web cam etc. Connect all these devices with the given PC, install & test them.
11	Identify the problem in the given PC, using the given troubleshooting sequence, fix the issue, record the given problem, and produce proper documentation of your work.

12	Recognize common symptoms associated with diagnosing and troubleshooting PCs and utilize Windows built-in diagnostic tools. ● Identify general troubleshooting techniques and strategies ● Utilize scandisk, control panel, boot-up menu, and startup disk as diagnostic tools. ● Access Microsoft Knowledge Base on the Internet to solve common problems. ● Identify the common problems associated with shutdown, configuration, and cabling. ● Identify problems associated with heating and cooling of the internal components. ● Identify problems with installing internal devices such as hard drive, tape drives, or CD-ROM drive. ● Recognize and interpret the meaning of common error codes and start up messages. ● Recognize windows-specific printing problems and corrections.
13	Perform computer maintenance and preventative maintenance functions. ● Perform physical cleaning (internal and external) of a personal computer. ● Demonstrate how to adjust basic performance settings. ● Perform hard drive file system maintenance. ● Identify anti-virus software and applications
14	Utilize the Internet to download device drivers. Installation of drivers of various devices from the internet. Demonstrate to remove unwanted software applications.

Name of the Programme : B.Sc. Computer Science
Course Code : CSC-142
Title of the Course : Multimedia and Web Design
Number of Credits :3 (1 T + 2 P)
Effective from AY : 2023-24

Pre-requisites for the Course:	NIL	
Course Objectives:	1. To introduce the basic concepts of Multimedia and Web Designing 2. To develop skills and competencies in image, video editing. 3. To acquire and develop skills to create web pages using HTML, CSS, Bootstrap and JavaScript	
Content:		**No of hours**
	UNIT 1: Computer Graphics Difference between Raster and Vector Graphics , Raster graphics : resolution, image compression, file formats, manipulation, Geometrical transformations, Vector graphics – fundamentals, file formats, shapes, transforms and filters. **Sound :** Sound Design, Audio Codec & file formats, processing sound, compression **Video** Aspect Ratio Frame Size, Frame Rate, Regions, Video Codec & Formats, Processing. **Web Architecture, HTML** : Introduction to internet and web design. Basic concepts of web architecture. Introduction to hypertext mark-up language (html), creating web pages, lists, hyperlinks, tables, web forms, inserting images. **Cascading style sheet (CSS)** : Concept of CSS, creating style sheet, Importing style sheets, CSS properties, CSS styling (background, text format, controlling fonts), CSS rules, Style Types, CSS Selectors, working with block elements and objects, working with lists and tables, CSS id and class, box model. **Basics of JavaScript** : Document object model, data types and variables, functions, methods and events, controlling program flow, built-in objects and operators, validations.	**15**
Pedagogy:	PowerPoint presentations, Practical Assignments	
References/ Readings:	Main Reading: 1. Nigel Chapman, Jenny Chapman(2004),“*Digital Multimedia*”, 2nd Edition, Wiley India Edition 2. Laura Lemay ,Rafe Colburn , Jennifer Kyrnin(2016), “*Mastering HTML, CSS & JavaScript Web Publishing*”, 1st Edition, BPB Publications Additional Reading: 1. Roger Parker(1997),“*One-Minute Designer*”, Revised Edition, Hungry Minds Inc, U.S.	

	2. Ranjan Parekh(2017), “*Principles of Multimedia*”, 2nd Edition, McGraw Hill Education 3. Tay Vaughan(2011), “*Multimedia Making It Work*”, 8th Edition, McGraw Hill 4. Alex Libby, Gaurav Gupta, AsojTalesra(2016), “*Responsive Web Design with HTML5 and CSS3 Essentials*”, Packt Publishing
Course Outcomes:	On successful completion of this course, the student will be able to: 1. Create and edit images, audio and video 2. Build websites using the elements of HTML& interactive, stylish websites using the client-side programming techniques with CSS and JavaScript. 3. Learn to validate client-side data. 4. Define the structure and contents of the website using different features of CSS

List of Practical Experiments: (60 Hours)

Multimedia

i. Image compositing : Remove background and combine images to create a work of art. (GIMP)
ii. Learn to create images for Print, Web and Video
iii. Design a Logo for a company
iv. Design a Brochure for given Product and details. Learn about different file formats(SCRIBUS)
v. Design a poster with given information and learn about image compression(SCRIBUS)
vi. Edit the sound file and Learn about Effects and Filters of sound.(AUDACITY)
vii. Record your voice and learn about Audio Compression(AUDACITY)
viii. Learn Audio mixing and streaming of audio content(AUDACITY)
ix. Learn about Video editing – Prepare video with rough cut.(WINDOWS MOVIE MAKER)
x. Prepare video content with title and special effects. (WINDOWS MOVIE MAKER)
xi. Record video content and learn about video compressions. (WINDOWS MOVIE MAKER)
xii. Prepare Video content for vimeo / youtube. (WINDOWS MOVIE MAKER)

Note: Practical can be done using GIMP, Inkscape, Scribus, Blender, Audacity, Lightworks / Kdenlive

Web Design

HTML

i. Create a basic HTML document structure with the appropriate doctype, <html>, <head>, and <body> elements.
ii. Add a heading (<h1>) and a paragraph (<p>) to your HTML document.
iii. Create a hyperlink (<a>) that navigates to another web page when clicked.
iv. Insert an image (<img>) into your HTML document and provide an appropriate alt text.
v. Construct an ordered list (<ol>) with three list items (<li>).
vi. Design a table (<table>) with two columns and two rows, containing relevant data.
vii. Design table using colspan, rowspan
viii. Build a simple form (<form>) with text input fields (<input type="text">) for name and email.

ix. Implement a checkbox (<input type="checkbox">) and a radio button (<input type="radio">) , lists within a form.
x. Create a navigation menu using semantic HTML tags such as <header>, <nav>, and <ul>.
xi. Embed an audio or video file (<audio> or <video>) into your HTML document.

CSS

Write CSS to ,

i. Apply inline styles to a paragraph (<p>) to change its color and font size.
ii. Link an external CSS file to your HTML document using the <link> tag.
iii. Change the colour of text using CSS
iv. Change the font family and font size of text using CSS
v. Add a background color to an element using CSS
vi. Add padding and margin to elements using CSS
vii. Align text and elements using CSS properties
viii. Change the border color, width, and style of an element using CSS
ix. Style hyperlinks using CSS
x. Apply styles to specific elements using class selectors in CSS
xi. Create and style lists using CSS
xii. Change the display and visibility of elements using CSS
xiii. Control the size and position of elements using CSS properties
xiv. Add shadows and gradients to elements using CSS
xv. Apply styles to images using CSS
xvi. Create and style basic CSS animations and transitions
xvii. Override CSS styles using inline styles
xviii. Include and use external CSS stylesheets in HTML documents
xix. Use CSS pseudo-classes to style elements based on their state
xx. Create and style basic CSS layouts using floats and clear properties
xxi. Style form elements, such as input fields and buttons, using CSS

Bootstrap

i. Include Bootstrap in an HTML document using CDN , local files
ii. Create a responsive grid layout using Bootstrap's grid system
iii. Create and style navigation menus using Bootstrap's navbar component
iv. Create and style buttons using Bootstrap's button classes
v. Incorporate and style Bootstrap's predefined CSS classes for typography
vi. Create and style forms using Bootstrap's form components
vii. Create and style responsive images using Bootstrap's responsive image classes
viii. Incorporate and style Bootstrap's modal dialogs
ix. Use Bootstrap's responsive utilities to control visibility and alignment

JavaScript

i. Validate a user's input in a form using JavaScript
ii. Write a JavaScript function that reverses a given string.
iii. Implement a slideshow or image carousel using JavaScript.
iv. Write a JavaScript program that finds the largest number in an array.
v. Detect and handle browser events like clicks or keyboard input using JavaScript.
vi. Write a JavaScript function that calculates the factorial of a given number.
vii. use JavaScript to manipulate the CSS properties of an HTML element dynamically.

Note : Web design Practical can be done using VS code IDE

Name of the Programme	**: B.Sc. Computer Science**
Course Code	**: CSC-143**
Title of the Course	**: Data analytics using Spreadsheets I**
Number of Credits	**: 03 (1 T + 2 P)**
Effective from AY	**: 2023-24**

Pre-requisites for the Course:	Nil	
Course Objectives:	Introduce the basic concepts of data analytics; develop proficiency in students in using spreadsheets to format data, manipulate data using appropriate basic function and formulas; visualize data; filter data and generate basic reports using Pivot tables.	
Content **Content**		**No of Hours**
	Unit I: Introduction to Data Analytics and Spreadsheet Basics Definition of Data Analysis and Data Analytics Phases of Data Analysis. Methods of Data Analysis in Spreadsheets. Understanding Data: Data and types of data. Quantitative data – discrete data, continuous data. Qualitative data - categorical data, ordinal data. Understanding operators and functions essential for data analytics. Arithmetic operators and order of operations. Functions: Parts of a function, arguments to a function, function library and types of functions.	**5**
	Unit II: Data Collection and Manipulation. Data Collection using online data collection tools. Creating Spreadsheets online and collaboration. Introduction to data cleansing, data modification using data analysis functions. Sorting criteria and types of sorting, Filters and types of filters, Guidelines and examples for sorting and filtering data by colour: Overview of sorting and filtering data by colour and icon set, using colour effectively when analysing data, choosing the best colours.	**5**
	Unit III: Data Visualization and Summarization Visualizing data: Principles of charting, types of basic charts, Some practicalities in preparing charts. Conditional Formatting and its types. Functions used for data summarization. Pivot tables and its applications.	**5**
	Practical	
	Unit I: Spreadsheet Basics: ● Formatting Cells with font formats, alignment, borders etc. ● Number formats, currency formats, formatting dates, custom and special formats. ● Format painter ● Selection techniques ● Advanced paste special techniques: paste value, paste formulas, paste formats, paste validations, transpose tables	**20**

	● Formulas and Functions: ● Complex Formulas with arithmetic operators ● Relative, mixed and absolute cell reference ● Basic Functions such as sum, average, max, min, count, counta. ● Customization, Formatting and Protection: ● Customizing the ribbon, Using and customizing autocorrect ● Changing Excel default options ● Page Layout and printing options: Setting up print area, customizing headers and footer, print titles. ● File level protection, workbook, worksheet protection ● Working with named ranges, ● Commonly used shortcut keys **Essential Data Analysis Functions and Methods:** ● Text Functions: Upper, Lower, Proper, Left, Mid, Right, Trim, Len, Exact, Concatenate, Find, Search, Substitute ● Date and time Functions: today, now, day, month, year, date, date if, dateadd, EOMonth, weekday, days, networkdays ● Logical functions: TRUE, FALSE,IF, AND, OR ● Nested if, IF function together with AND, OR function ● Data cleaning and preparation using text functions and text to column.	
	Unit II **Data collection using online data collection tools such as Google Forms:** ● Creating data forms to collect data for different types of scenarios such as surveys, event registration, feedback etc. ● Creating forms with conditional data input workflows based on user choice. ● Online storage of spreadsheets: ● Creating online spreadsheets such as google sheets and online collaboration of the same. ● Working with multiple worksheets & spreadsheets: ● Scenarios which require creating a workbook with multiple sheets and cross referencing. ● Scenarios which require creating multiple workbooks with multiple sheets and cross referencing across workbooks. ● Methods used in data analytics: ● Freezing Rows and Columns ● Sorting Data ● Filtering Data ● Summarizing Data ● Formatting Data as Table	**20**
	Unit III **Data Analytics Methods:** ● Visualizing data with charts. ● Adding Conditional Formatting. ● Essential advanced Data Analysis Functions: sumif, sumifs, countif, countifs, averageif, averageifs, nested if, iferror statement, and, or, not	**20**

	● Introduction to simple pivot tables. ● Solving real life problems/scenarios in spreadsheets.	
Pedagogy:	● Blended learning: Concept learning through Lab assignments and online video resources followed by application of concept learnt to real life scenario provided. ● Practical skill development through Lab assignments.	
References/ Readings:	Main Reading: 1. DavidWhigham(2007), “*Business Data Analysis using Excel*”, Oxford University Press Additional Reading: 1. Wayne Winston(2019), “*Microsoft Excel 2019 – Data Analysis and Business Modelling*”, PHI Learning Pvt. Ltd.	
Course Outcomes:	**At the end of the course, learner will be able to:** 1. Format a given spreadsheet with various formatting features and use appropriate functions given relevant description of desired output. 2. Sort, filter, summarize data given in a spreadsheet as per given instructions 3. Visualize data using appropriate charts and conditional formatting. 4. Solve basic queries on a given data set by preparing basic pivot tables for a given data set.	

Name of the Programme : **UG Degree (Honors) with Computer Science**
Course Code : **CSC-144**
Title of the Course : **Desktop Publishing**
Number of Credits : **3 (1 T + 2 P)**
Effective from AY : **2023-24**

Pre-requisites for the Course:	Nil	
Course Objectives:	1. To introduce the basic concepts of Desktop Publishing and Page Layout. 2. To develop skills and competencies in image editing. 3. To acquire and develop skills for Digital Content Creation for various platforms.	
Content:		**No of hours**
	1. Introduction Definition of Desktop Publishing and Digital Publishing. Introduction to open source and proprietary software used in DTP. Print Media v/s Digital Media. Benefits of Desktop and Digital Publishing. Examples of Desktop Publishing and Digital Publishing. Digital Publishing Platforms. Branding and Identity.	**03**
	2. Typography and Color Definition of Typography; Common Types of Fonts; Choosing a Font; Kerning, Leading and Tracking. Color Basics; Hue, Saturation and Value; Color Wheel.	**04**
	3. Layout and Design Basics of page layout; page layout in pictures measurement units like inch, pica and points; features of good typography; Serif and sans serif fonts. Basic design principles: Proximity, White Space, Alignment, Contrast and Repetition. Fundamentals of Design: Line, Shape, Forms, Texture and Balance.	**04**
	4. Images Graphics: Raster v/s Vector, Lossy v/s Lossless Compression, Common Image Formats, Image Manipulation Techniques, Image Usage Rights	**04**
Pedagogy:	Practical assignments using open source software/platforms such as Gimp, Canva	
References/ Readings:	Main Reading: 1. Nigel Chapman, Jenny Chapman(2004), “*Digital Multimedia*”, 2nd Edition, Wiley India Edition	

Course Outcomes:	**At the completion of this course, the learner will be able to:** 1. Explain the basic concepts of Desktop Publishing and its relevance in e-content development. 2. Apply typographic and color schemes used for the layout and designing e-content. 3. Apply the editing features for given images. 4. Develop e-content for a given product for various platforms

Suggested Practical List: 60 Hours

(at least 10-12 Practicals from the following)

1. Image Editing and Graphic Manipulation
   a. Basic Transformation Tools
   b. Enhancing images (contrast and brightness changes)
   c. Image compression using different file formats
   d. Applying special effects and filters on images
2. To create a social media Post for any platform.
3. To create a social media Story for any platform.
4. To create Animated Social Media content such as Instagram Reels.
5. To create social media ads for any platform.
6. To design Covers for any social media platform.
7. To design a Logo for a given product.
8. To design a Poster with the given information.
9. To design a Flyer with the given information.
10. To design a Banner for a given product.
11. To design an Advertisement for a given brand.
12. To develop Infographics content on a given topic.
13. To design a Newsletter covering the given events for your department.
14. To design a Magazine Cover for your college.
15. To design a Brochure for a given product.

Name of the Programme : B.Sc. Computer Science
Course Code : CSC-145
Title of the Course : Basic Computer Applications
Number of Credits : 3(1T+2P)
Effective from AY : 2023-24

Pre-requisites for the Course:	Nil	
Course Objectives:	To Provide an Understanding of Essentials of Information Technology, Internet Applications and Emerging Technologies. Includes practical skills in data capture, presentation, report formatting, efficient search techniques and online collaboration tools.	
Content:		**No. of Hours**
	Unit I: Information Technology Basics Information : Prerequisites of Information, Need for Information Technology and its advantages; Information Technology: Definition and components; Data: Definition, Types, Data Representation, Number system and Coding Schemes (ASCII and UNICODE); Parts of a Computer: CPU, Memory, Input/ Output Devices, Auxiliary Memory; Software– Definition, Relationship between Hardware and Software, Categories of Software, OS - definition & functions Role of Information Technology in: Business, Mobile Computing, Health Services, Public Sector, Media, Defense Services, Education and Publication.	**10**
	Unit II: InternetApplicationsandEmergingTechnologies Internet – role and importance, Web Browser, IP Addressing– Public Vs Private, Static VsDynamic;WWW&relatedprotocols; InternetApplications. Cloud Computing: Meaning, Features, & Service models, Advantages and disadvantages, Mobile Computing: Meaning, Business Applications ofMobile computing, Virtual reality & Augmented Reality: Meaning and applications, IoT - InternetofThings: Meaning& Applications.	**5**
	Practicals	

	Lab1: Basic Computer Skills Surfing the Internet, Use of Email and Search Engines Securing your device Installation and Configuration of any free Antivirus Package eg. AVG/Avast etc., **Online Sharing and Collaboration** Create documents, spreadsheets and presentations online, Share and collaborate in real time, Safely store and organize your work, Control who can see your documents **Data capture using Google Forms** Create data forms to capture data for Event Registration, Event Feedback, Customer feedback/satisfaction on a product or service and Order Request **OS Basic** Installation of Operating System, Demonstrate features of any MS Windows based OS or any of the Linux flavor , Identification of Directories , Setting up computer, Add a printer, Check device drivers, Installation of software, Users and administrative rights for installation	**10**
	Lab2: Report Formatting using Word Processing (MS Word or any similar Open Source software) Draft an official letter for job interview invitation/ job appointment/ invitation to a business trade show event, use mail merge to input the recipients list linking with database. Given a project report in PDF format transfer to word processor software and format to include title page, specified Paragraph and Page Formatting (page size, orientation, line spacing, font type and font size, Indent, bullets, paragraph formatting) details, Acknowledgement page, Table of contents page, List of figures page, List of Tables page, bibliography, references, distinct headers for each chapter, page numbering in roman for initial pages and normal from first chapter. The document should be checked for spelling errors and corrected appropriately. Create / Upload a document in a collaboration software like Google docs. Share and collaborate in real time, Safely store and organize your work, Control who can see your documents.	**30**
	Lab 3: Presentation Software (MS- Powerpoint or any similar Open Source software) Preparing presentation in areas such as Customer satisfaction/ feedback, product analysis, job satisfaction using the data obtained through data capture tool, including appropriate slide animation, sound recording, slide timings, customer feedback video. Export the presentation as video or save as slide show. Prepare handouts for audience.	**20**
Pedagogy:	MS-Word, MS-Powerpoint or any similar open source software may be used	
References/ Readings:	Main Reading: 1. Pradeep K. Sinha and PritiSinha(2022), "*Computer Fundamentals*", BPB	

	Publications 2. ITL Education Solutions Limited(2005), "*Introduction to Information Technology*", Pearson Education Additional Reading: 1. M. Arvind Babu, Dr. S. Anandamurugan, T. Priyaa(2016), "*Cloud Computing*", First Edition,Laxmi Publications Pvt Ltd 2. ArshdeepBahga, Vijay Madisetti (2014), "*Internet of Things: A Hands-On Approach*", Vijay Madisetti Publications
Course Outcomes:	At the end of the course, learner will be able to: 1. Explain the basic Knowledge and Understanding of Information Technology, Internet Applications and Emerging Technologies. 2. Understand future technologies through foundational skills learnt. 3. Develop practical skills in Application software. 4. Apply technology and professional development in IT.

Name of the Programme : B.Sc. Computer Science
Course Code : CSC-146
Title of the course : Basics of Python Programming
Number of Credits : 3 (1T+2P)
Effective from AY : 2023-24

Pre-requisites for the Course:	Basic working knowledge of Computers and Internet	
Course Objectives:	1. To introduce programming concepts using Python. 2. To introduce object-oriented programming concepts.	
Content:		**No. of Hours**
	(Theory) 1. Python Interpreter, Python Shell, strings, relational operators, logical operators, precedence of operators, bitwise operators, variables and assignment statements, script mode, functions, modules, command line arguments, control structures- if conditional statements, iteration for and while statements, break, continue and pass statements.	**4**
	2. Data types- Boolean, numbers, coercing integers to floats and vice versa, numerical operations, lists, creating a list, slicing a list, adding and removing items from a list, searching for values in a list, tuples, immutability property, converting tuples into a list, sets, set operations, dictionaries, strings, Unicode, formatting strings, docString, modules, packages, scope, recursion	**4**
	3. Object Oriented Concepts- Classes, Objects, Abstract Data types, polymorphism, encapsulation, modifier, accessor methods, static method, adding methods dynamically, composition, inheritance, built-in functions for classes.	**4**
	4. File Handling, Exception Handling	**2**
	5. Applications of Python - use of Python libraries such as Matplotlib, Pandas, using databases with python, collecting information from Twitter etc. (at least three applications to be covered).	**1**
	(Practicals) (15x4=60hrs) List of Practicals : (at least 8 practicals from the following) 1) a) Write a function that returns the sum of digits of a number, passed to it as an argument. b) Write a function that returns True or False depending on whether the given number of a palindrome. c)Take the radius of circle as input from the user, passes it to another function that computes the area and the circumference of the circle and displays the values.	

d) Write a function that finds the sum of the n terms of the following series:

1 - x2 / 2! + x4 / 4! - x6 / 6! +. ….. xn/ n!

2) Perform following actions on a list :

a) Print the even-valued elements

b) Print the odd-valued elements

c) Calculate and print the sum and average of the elements of array

d) Print the maximum and minimum element of array.

e) Remove the duplicates from the array

f) Print the array in reverse order

3) a)Define a function which can generate and print a list where the values are square of numbers between 1 and 20 (both included). Then the function needs to print all values except the first 5 elements in the list.

b) Write a program which takes 2 digits, X,Y as input and generates a 2-dimensional array. The element value in the i-th row and j-th column of the array should be i*j.

4) a)Write a program that accepts sequence of lines as input and prints the lines after making all characters in the sentence capitalized.

b) Write a program that accepts a sentence and calculate the number of letters and digits.

c) Given an array of integers, find two numbers such that they add up to a specific target number.

5) a)Write a function that takes a list of values as input parameter and returns another list without any duplicates.

b) Write a program that takes a sentence as input from the user and computes the frequency of each letter. Use a variable of dictionary type to maintain the count.

6) a)Write a recursive function that multiplies two positive numbers a and b and return the result. Multiplication is to be achieved as a+a+a (b times).

b) Write a recursive function that inserts the element x at every n th position in the given list and returns the modified list.

7) a)Given a list of strings, return the count of the number of strings where the string length is 2 or more and the first and last characters of the string are the same

b) Given a list of strings, return a list with the strings in sorted order, except group all the strings that begin with 'x' first. e.g. ['mix', 'xyz', 'apple', 'xanadu', 'aardvark'] yields ['xanadu', 'xyz', 'aardvark', 'apple', mix‘]

8) Define a class Student that keeps track of academic record of students in a school. The class should contain the following data members:

- rollnum - roll number of the student
- name - name of the student
- marksList - List of marks in 5 subjects
- stream - A: Arts, C: Commerce, S: Science
- percentage - percentage computed using marks
- grade - grade in each subject computed using marks
- division - division computed on the basis of overall percentage

The class should support the following methods:

a. __init__ for initializing the data members

b. setMarks to take marks for five subjects as an input from the user

c. getStream for accessing the stream of the student.

d. Percentage for computing the overall percentage of for the student.

e. gradeGen that generates grades for each student in each course on the basis of marks.

Marks	Grade
>=90	A
<90 and >=80	B
<80 and >=65	C
<65 and >=40	D
<40	E

f. division for computing division on the basis of the following criteria based on overall percentage of marks scored:

Percentage	Division
>=60	I
<60 and >=50	II
<50 and >=35	III

g. __str__ that displays student information.

9) Define a base class Vehicle , having attributes registration number, make, model and color. Also, define classes Passenger Vehicle and Commercial Vehicle that derive the class Vehicle. The Passenger Vehicle class should have additional attribute for maximum passenger capacity. The Commercial Vehicle class should have an additional attribute for maximum load capacity. Define __init__ method for all these classes. Also, get and set methods to retrieve and set the value of the data attributes.

10) Define classes Car, Autorickshaw and Bus which derive from the Passenger Vehicle class mentioned in the previous question. The Car and Bus should have attributes for storing information about the number of doors, not shared by Autorickshaw. The Bus should have Boolean attribute doubleDecker not shared by Car and Autorickshaw. Define __init__ method for all these classes. Also define get and set methods to determine and set the value of the day attributes.

11) Develop a program to sort the employee data on the basis of pay of

	the employees using i) selection sort ii) bubble sort. iii) insertion sort. Consider a list L containing objects of class Employee having empNum, name and salary. 12) Write a function that takes two file names, file1 and file2 as input. The function should read the contents of the file file1 line by line and should write them to another file file2 after adding a newline at the end of each line. 13) Write a function that reads a file file1 and displays the number of words and the number of vowels in the file. 14) Write a function that reads the contents of the file Peom.txt and counts the number of alphabets, blank spaces, lowercase letters and uppercase letters, the number of words starting with a vowel and the number of occurrences of word —beautiful‖ in the file. 15) Write a function that takes two files of equal size as an input from the user. The first file contains weights of items and the second file contains corresponding prices. Create another file that should contain price per unit weight for each item. Note : Testing and Debugging tools to be used during the practical sessions.
Pedagogy:	● Powerpoint presentations ● Group Discussions
References/ Readings:	Main Reading : 1) Taneja Sheetal, Kumar Naveen(2017), “*Python Programming - A modular approach*”, 1st Edition, Pearson Additional Reading: 1) John V.Guttag(2016), “*Introduction to Computation and Programming using Python*”, 2nd Edition, MIT Press
Course Outcomes:	At the end of the course, learner will be able to: 1. Use the basic programming syntax with Python programming language, Python Interpreter and Command Line Arguments. 2. Describe the data types, various Control Structures, Packages, Recursion and File Handling concepts available in Python. 3. Explain and use Object Oriented Programming (OOPs) Concept and its features. 4. Develop simple Python Applications using various Python Libraries.

Name of the Programme : **B.Sc. Computer Science**
Course Code : **CSC-147**
Title of the Course : **Graphical User Interface Design**
Number of Credits : **03 (1 T + 2 P)**
Effective from AY : **2023-24**

Pre-requisites for the Course:	Nil	
Course Objectives:	This course is aim is to: 1. teach basic concepts of interface design. 2. train to create interface prototypes to test usability. 3. explain user personas and experiences. 4. teach to create user engaging interfaces	
Content:		**No. of Hours**
	I. Introduction to UI/UX UI and UX An overview of the user interface's history User experience (UX) Similarities and Differences between UX and UI The job of the user experience designer The UI designer's job description	**01**
	II. User Persona for UX Design User Flow in UX Design User flow in UX Tools to make a user experience flow UX Design Prototypes Creating a user experience prototype Test designs prototypes Wireframes in UX Design Benefits and use of wireframes	**02**
	III. Basic visual design principles in UI Design Creating attractive and functional interfaces Definition of Design thinking Design Thinking Phases The Science of Creativity in the Brain Advantages of intuition Importance of Intuitive Design Advantages and disadvantages of intuitive design Characteristics and Benefits of Using User-Friendly Software	**03**
	IV. Common tools for UI Designs Using Balsamiq Wireframes Creating Wireframes with Axure Use Axure RP to Create Wireframes Tips to achieve a good user interface and Experience	**03**
	V. Understanding what Typography is	**02**

	Understanding typefaces, fonts, and font selection Obey the laws of typography A glossary of typographic terms	
	VI. Way to test contrast in UI design Design alignment Significance of alignment in design Visual Hierarchy's Building Blocks Creating an Effective Visual Hierarchy UI and its relation with readability Advice on Improving Readability and Legibility	**02**
	VII. Font pairing Basics of Font Pairing Adding Meta information in UI/UX design SEO and UI Design Responsive Website Design SEO and UX Design	**02**
Pedagogy:	● Lectures to be conducted using computer and projector ● Hands on practice of all concepts covered in theory sessions	
References/ Readings:	Main Reading: 1) James Cabrera(2017), "*Modular Design Frameworks : A Projects-based Guide for UI/UX Designers*", 1st Edition, APress Additional Reading: 1) Westley Knight(2018), "*UX for Developers: How to Integrate User-Centered Design Principles Into Your Day-to-Day Development Work*", 1st Edition, APress 2) Laura Klein (2013), "*UX for Lean Startups: Faster, Smarter User Experience Research and Design*", 1st Edition Shroff/O'Reilly 3) Jesmond Allen & James Chudley(2012), "*Smashing UX Design: Foundations for Designing Online User Experiences*", John Wiley and Sons. 4) Edward Stull (2018), "*UX Fundamentals for Non-UX Professionals: User Experience Principles for Managers, Writers, Designers and Developers*", 1st Edition, APress	
Course Outcomes:	At the end of the Course, learner will be able to: 1. Understand the principles and concepts of Interface design 2. Explain UX 3. Apply better interfaces for effective UX 4. Create intuitive interfaces	

Practical Work

Sr. No	Topic	**No. of Hour**

1	Exercises to Identify interface connectivity and establish interface connectivity between two different program modules. ● Choose any of the programming languages (like HTML, JavaScript, Bootstrap etc.), do create two separate programming files and establish the interface connectivity between both.	**08**
2	Exercisesthat will facilitate Understanding frontend and backend interface and implementation of both interfacing. ● Front –end and back-end interfacing languages used for interface design. ● HTML: HTML stands for Hypertext Markup Language. It is used to design the front-end portion of web pages using a markup language.	**16**
3	Exercises to create wireframes designs:- ● Modelling wireframes designs ● Implementing wireframes	**14**
4	Exercises using font, color matching and typography:- ● Use of colors and contrasts ● Font suitability ● Matching content to target users	**10**
5	Exercises using responsive design on :- ● Web pages ● Desktop software screens ● Smartphones/ Tabs and other handheld devices	**12**

Name of the Programme	**: B.Sc. Computer Science**
Course Code	**: CSC-148**
Title of the Course	**: Data analytics using Spreadsheets II**
Number of Credits	**: 03 (1 T + 2 P)**
Effective from AY	**: 2023-24**

Pre-requisites for the Course:	Basic knowledge of Spreadsheets.	
Course Objectives:	Develop ability to use spreadsheets for conditional data summarization, financial calculations, advanced data visualization. Work with pivot tables and charts to obtain insights, use lookup functions for data manipulation, perform what-if analysis. Create a dashboard in Excel. Obtain basic descriptive statistics using analysis tools.	
Content:	**Theory**	**No of hours**
	Unit I : Financial Data Analysis and Advanced data Visualization: Data Analysis financial functions ● Financial arithmetic basics and Investment Appraisal functions- modeling financial data in Spreadsheets. ● Data Analytics advanced visualization methods ● Data Visualization with charts such as tree map, waterfall, sunburst, box and whisker, power maps.	**5**
	Unit II: Steps in data analytics: ● Preparation of data: Data collection, data cleansing and data validation ● Elementary data modeling – linear functions in business, expressions and functions involving logical tests, vertical lookup functions, combining conditional statements with lookup functions.	**5**
	Unit III: Statistical analysis of data using Spreadsheets: ● Collating and categorizing data, data description-central tendency and dispersion, descriptive statistics using Analysis Tool Pak.	**5**
	Practical **Unit I:Data Analysis advanced functions and methods** ● Financial Functions: FV, PV, NPV, IRR, PMT (loan amortization schedule) ● Scenarios for visualizing data using charts such as tree map, waterfall, sunburst, box and whisker, combo charts, power maps and 3D Maps ● Advanced Sorting option and Advanced Filters	**20**
	Unit II ● Data Validation: Number, Date and Time Validation, Text and List Validation, Custom validation based on formula for a cell, Dynamic dropdown list creation using data validation-dependency list.	**20**

	● What-if Analysis: Goal Seek, scenario analysis, data tables using PMT function, Solver tool ● Lookup Functions: Vlookup and Hlookup functions, Index and Match, Reverse Lookup using choose function.	
	Unit III ● Pivot Tables and Pivot Charts: Creating advanced pivot tables with advanced value field settings, filtering pivot tables, modifying pivot table data, grouping based, Pivot Charts and Slicers. Filter data with slicers, manage primary and secondary axis ● Creating Interactive Dashboard: Planning a Dashboard, Adding Tables and charts to dashboard, adding dynamic content to dashboard. ● Descriptive statistics using Analysis ToolPak. ● Introduction to Excel macros and VBA Basics.	**20**
Pedagogy:	Blended learning: Concept learning through Lab assignments and online video resources followed by application of concept learnt to real life scenario provided. Practical skill development through Lab assignments.	
References/ Readings:	Main Reading: 1. DavidWhigham(2007), "*Business Data Analysis using Excel*", Oxford University Press Additional Reading: 1. Wayne Winston(2019), "*Microsoft Excel 2019 – Data Analysis and Business Modelling*", PHI Learning Pvt. Ltd.	
Course Outcomes:	At the end of the course, learner will be able to: 1. Use conditional arithmetic functions to summarize data and use financial functions, given a spreadsheet with data and relevant description of desired output. 2. Perform what-if analysis and data validation on given data for a given scenario. 3. Summarize and analyze data using Pivot Tables and Pivot Charts. 4. Apply and visualize data using Dashboard and descriptive statistics using Analysis ToolPak.	

Name of the Programme : B.Sc. Computer Science
Course Code : CSC-149
Title of the Course : Data and Business Analytics
Number of Credits : 3(1L+2P)
Effective from AY : 2023-24

Pre-requisites for the Course:	Nil	
Course Objectives:	To understand data processing, data analysis, business analytics concepts, computer networking basics, e-commerce technology and business applications; To develop practical skills in data analytics and business analytics.	
Content:		**No of hours**
	Unit I: Data processing, Data Analysis and Business Analytics Data Processing – Steps involved in data processing, advantages of computers in data processing Data analysis and forecasting - importance of data analysis in business, Data forecasting, its need, benefits of data forecasting, Data Integration: concept and how it works Introduction to Business analytics – meaning and basic concepts,Visualization/ Data Issues: Organization/sources of data, Importance of data quality, Dealing with missing or incomplete data, Data Classification	**4**
	Unit II:E-Commerce Definition, E-commerce and Trade Cycle, Electronic Markets, Electronic Data Interchange andInternetCommerce,TypesofE-commerce:BusinesstoBusinessE-Commerce,BusinesstoConsumerE-Commerce.ConsumertoConsumer, Government to Consumer, Business to Government, ElectronicPaymentSystems:SmartCards–Credit Cards – Wallets, Safe practices, Risks, E-Retail, Concept and Examples, Onlineshopping–Introduction, Safety measures (Encryption ofdata authentication, SSL, Digital signatures, Digital Certificates), E-Banking, Features and services,M-Commerce,Productsand services	**5**
	Unit III : Basics of Computer Networking Networking basics, Need for computer networks, Types of networks-LAN, MAN, WAN, Network Components–H/W, Software, Communication channels, Network Devices, Network topologies.	**6**
	Practicals	
	Lab1: Spreadsheet (MS-Excel or any similar open source software) ● Working with worksheets -Entering data, Formatting, Editing, and Printing a worksheet,	**36**

	● Formulas and Functions in Excel, operators in formula ● Generally used Spreadsheet functions - Mathematical, Statistical, Financial, Logical, Date and Time, Database and Text functions ● Introduction to some more useful functions such as the IF, nested IF, VLOOKUP and HLOOKUP ● Data Sorting and Filtering ● Result representation of data using spreadsheet ● What-if analysis, Logical tests(nested if functions), Goal seek, ● Representing results graphically ● Filtering, advanced filters, sorting and conditional formatting data ● Data validation techniques, Hyperlinks ● Pivot table, Scenarios ● Summing through the sheets ● Getting external data files into Excel ● Macros - creation, editing and deletion of macros Assignments to be given on the following topics: to prepare and analyse Loan and Lease statement; Ratio Analysis; Payroll statements; Capital Budgeting; Depreciation Accounting; Graphical representation of data; Frequency distribution and its statistical parameters; Correlation and Regression	
	Lab2: Data Analytics ● Assignments to analyse data available from IndiaStat.com such as Analysis of demographic data, environment data, public expenditure ● Analyse data from annual reports of Companies and banks	**8**
	Lab3: E-commerce Website review Write a review of an E-Commerce Site visited include: Site description, Site Design, ease in navigation , process for purchasing items, security, privacy, customer service, best features of site, Target Audience, Revenue model, Marketing Strategies	**8**
	Lab4: Computer Networking Basic Networking Setup of PC, Network commands like ipconfig, ping, traceroute, nslookupetc, Setup of Home Router / Wifi Hotspot, Understanding of Firewall and Basic Firewall Setup, File and Printer Sharing, connecting to share, Finding out public address, connection speeds etc.	**8**
Pedagogy:	MS-Excel or any similar open source software may be used Field visits may be conducted to banks, corporate offices employing relevant software for business applications.	
References/ Readings:	Main Reading: 1. ITL Education Solutions Limited(2005), "*Introduction to Information Technology*", Pearson Education 2. Ravi Kalakota& Andrew B. Whinston(2009), "*Frontiers of Electronic Commerce*", Ninth Impression, Pearson Education. Additional Reading:	

	1. David Whiteley(2000), "*E-Commerce: Strategy, Technologies And Applications*", McGraw-Hill Education, ISBN-10 : 0077095529 2. Thomas H. Davenport, Jeanne G. Harris(2010), "*Competing on Analytics- The New Science of Winning*", Harvard Business Review Press. 3. LaValle et al.(2005), "*Analytics: The New Path to Value*", Taxmann Publishers. 4. Davenport and Harris(2007), "*The Dark Side of Customer Analytics*", Harvard Business Review Press. 5. Bartlett, R.(2013) , "*A Practitioner's Guide to Business Analytics*", McGraw-Hill, New York. 6. Bruice Schneier, "*Applied Cryptography-Protocols, Algorithms and Source code in C",* Second Edition, Wiley India Pvt Ltd, ISBN 978-81-265-1368-0
Course Outcomes:	At the end of the course, learner will be able to: 1. Explain the concepts of data processing, data analysis, business analytics, computer networking, e-commerce technology and its applications in business. 2. Develop skills of data analysis and business analytics using relevant Application software. 3. Apply the Spreadsheet tools to solve business problems. 4. Review an E-commerce Website

Name of the Programme : **B.Sc. Computer Science**
Course Code : **CSC-150**
Title of the Course : **Database Management and Analysis**
Number of Credits : **3 (1L+2P)**
Effective from AY : **2023-24**

Pre-requisites for the Course:	Nil	
Course Objectives:	To familiarize the student with various applications of Information and Communication technologies in business and to familiarize with the mechanism for conducting business transactions through electronic means.	
Content:		**No. of Hours**
	Unit I : Database Management System ● Database Designs for Accounting and Business Applications: Reality- Expressing the Application; Creating Initial design in Entity Relationship(ER) Model; Transforming ER Model to Relational data model concepts; Implementing RDM design using an appropriate DBMS. ● SQL and Retrieval of Information: Basic Queries in SQL; Embedded Queries in SQL; Insert, Delete and Update statements in SQL ● DBMS Software: Environment; Tables; Forms; Queries; Reports; Modules; ● Applying DBMS in the areas of Accounting, Inventory, HRM and its accounting, Managing the data records of Employees, Suppliers and Customers.	**10**
	Unit II: Enterprise Resource Planning ● Introduction: Traditional information model, Introduction to an enterprise, What is ERP?, Reasons for growth of ERP market, Advantages and Disadvantages of ERP , Introduction to business modules: finance, manufacturing, Human resource, materials management, sales and distribution, Limitations of ERP, ERP and eCommerce	**5**
	Practicals	
	Lab1: Database Management System Creating Database and Tables, Changing Table Contents, Adding and Editing Records, Changing Table Properties, Creating Relationships between Tables, Importing and Exporting Data with other Programs. Creating Queries: Using the Query Window, Using Criteria and Saving Queries, Criteria Expressions and Operators, Changing a Query and Totaling, Creating a Query to the above-made Databases Creating Forms: Putting List Box on the Form, Selecting and Redesigning Labels and Data, Moving Label and Data, Adding Data and using Data Validation	**50**

	Creating Reports: Creating a single Column Report, Creating a Grouped Data Report, Adding Graphs to Reports. Use of Macros for search and navigation filters.	
	Lab2: ERP Mini Project Case study – Studying ERP implementation in any business firm Report preparation and submission – report shall include ERP introduction, life cycle as followed by the Business firm under study – pre-evaluation screening, package evaluation, project planning phase, gap analysis, reengineering, configuration, implementation team training, testing, going live, end user training, post implementation.	**10**
Pedagogy:	MS-Access or any similar open source software Field visits may be conducted to understand and demonstrate ERP Software.	
References/ Readings:	Main Reading: 1. Bipin Desai(2010), "*An Introduction to Database systems*", Revised Edition, Galgotia Publications 2. S Sadagopan(1999), "*ERP a Management Prospective*", Tata McGraw Hill Publishing Company Limited, New Delhi Additional Reading: *1.* "MS-Access manual" *2.* Alexis Leon(2000) , "*ERP Demystified*", Tata McGraw Hill Publishing Company Limited, New Delhi	
Course Outcomes:	At the end of the course, learner will be able to: 1. Describe database designing in DBMS software, Query language and understand its applications. 2. Creation and management of Database tables, queries, forms, reports and also macros in DBMS. 3. Explain the application of Enterprise Resource Planning in Business 4. Explain Implementation of ERP as a case study in any business firm.	

Exit Course

Name of the Programme : B.Sc. Computer Science
Course Code : CSC-161
Title of the Course : Network Creation, Maintenance, and Troubleshooting
Number of Credits : 4 (2T+2P)
Effective from AY : 2024-25 (Revised version of 2023-24)

Pre-requisites for the Course:	None	
Course Objectives:	1. To understand the basics of Networking – design, architecture, devices and configuration 2. To understand network services 3. To learn network monitoring and management 4. To learn troubleshooting of network	
		No of Hours
Content:	**Unit I:** Introduction to Networking Overview of Computer Networks: Types of Networks (LAN, WAN, WLAN), Network Topologies (Star, Bus, Ring), OSI Model and TCP/IP Protocol Suite Network Design and Architecture Network Design Principles, IP Addressing and Subnetting, VLANs (Virtual LANs) and Inter-VLAN Routing Network Devices and Configuration Introduction to Network Devices (Routers, Switches, Access Points), Configuring Cisco IOS Devices (Router and Switch Configuration), DHCP (Dynamic Host Configuration Protocol) Configuration Wireless Networking Wireless Standards and Technologies (802.11a/b/g/n/ac), Wireless Security (WPA2, Encryption), Configuring Wireless Networks (Access Points, SSIDs) Network Services DNS (Domain Name System) and DHCP Services, NAT (Network Address Translation), Introduction to Firewalls and Access Control Lists (ACLs) Network Monitoring and Troubleshooting Tools Network Monitoring Tools (Wireshark, SNMP), Troubleshooting Network Connectivity Issues, Introduction to Packet Analysis	**20**
	Unit-2: Network Security Common Network Threats and Vulnerabilities, VPN (Virtual Private Network) Technologies, Implementing Security Policies and Best Practices Quality of Service (QoS) Understanding QoS Requirements for VoIP and Video Streaming, Implementing QoS Policies on Routers and Switches Advanced Routing and Switching Routing Protocols (OSPF, EIGRP), Switching Concepts (STP, EtherChannel), Troubleshooting Routing and Switching Issues Network Redundancy and High Availability Implementing Network Redundancy (HSRP, VRRP), Failover and Disaster Recovery Planning	**10**

	Network Performance Optimization Performance Monitoring and Tuning, Identifying and Resolving Network Bottlenecks	
	Practical	
	Week 1 & 2 Setting up basic LAN connections using Ethernet cables and switches	**08**
	Week 3 & 4 Designing a small-scale network with VLAN segmentation	**08**
	Week 5 & 6 Setting up and configuring routers and switches in a simulated environment Configuring and securing a wireless network with encryption and authentication	**08**
	Week 7, 8 & 9 Implementing DNS and DHCP services on a network server Configuring NAT on routers for internet access Using Wireshark to analyze network traffic and troubleshoot connectivity problems	**12**
	Week 10 & 11 Configuring a VPN tunnel between two network sites Setting up basic firewall rules and intrusion prevention measures Configuring QoS to prioritize traffic types on a network	**08**
	Week 12 & 13 Implementing OSPF routing protocol and troubleshooting routing issues Setting up redundancy protocols to ensure network high availability	**08**
	Week 14 & 15 Analyzing network performance metrics and optimizing network settings	**08**
Pedagogy:	Suggested strategies for use to accelerate the attainment of the various course outcomes. Lectures, Tutorials, Collaborative/peer learning, Hands-on assignments	
References/ Readings:	Main Reading: 1. James F. Kurose and Keith W. Ross(2016), "*Computer Networking: A Top-Down Approach*", 7th Edition, Pearson 2. Todd Lammle (2016), "*CCNA Routing and Switching Complete Study Guide*" , 2nd Edition, Sybex 3. William Stallings (2016), "*Network Security Essentials: Applications and Standards*", 6th Edition, Pearson 4. Richard Bejtlich (2013), "*The Practice of Network Security Monitoring: Understanding Incident Detection and Response*", 1st Edition, No Starch Press Additional Reading: 1. W. Richard Stevens(2011),"*TCP/IP Illustrated, Volume 1: The Protocols*", 2nd Edition, Addison Wesley Professional 2. Sherri Davidoff and Jonathan Ham(2012), "Network Forensics: Tracking Hackers through Cyberspace", 1st Edition, Pearson	
Course	On completion of the course, students will be able to:	

Outcomes:	1. Explain the basics of networking – design, architecture, devices and configuration 2. Set up network using ethernet cales, switches and routers. 3. Troubleshoot network for issues. 4. Analyze network performance metrics and optimize network settings

Semester III
Name of the Programme : B.Sc. Computer Science
Course Code : CSC-200
Title of the Course : Programming Using C++
Number of Credits : 4 (3T + 1P)
Effective from AY : 2024-25 (Revised version of 2023-24)

Pre-requisites for the Course:	Basic problem solving skills	
Course Objectives:	1. Describe the principles of Object Oriented Programming 2. Understand the basics of C++ programming language. 3. Analyse the Object oriented concepts applied to C++. 4. Implement the Object Oriented Concepts using C++.	
Contents:		**No of hours**
	Unit I: Principles of Object Oriented Programming Software evolution, Procedure Oriented Programming (POP) vs Object Oriented Programming (OOP), Benefits of OOP, Applications of OOP, Object based vs Object Oriented languages Objects, Classes, Data Abstraction, Encapsulation, Inheritance, Polymorphism, Dynamic binding, Message passing, What is C++, Applications of C++, Simple C++ program, Comments, I/O Operators, the iostream file, return type of main (), Structure of C++ Program Keywords, Identifiers, Constants, Basic data types, User defined datatypes-Structures, Classes, Enumerated data types, Variables, reference variables, scope resolution operator, memory management operators, typecast operator, operator overloading Control structures: if statement, if else statement, switch statement, do while statement, while statement, for statement Functions in C++- the main () function, function prototyping, call by reference, return by reference, inline functions, default arguments, constant arguments, function overloading, friend functions, virtual functions, library functions	**15**
	Unit II: Classes and Objects Structures, Limitations of structures, Classes, Class declaration, visibility modifiers-public and private, creating objects, accessing class members, defining member functions inside and outside class definition, making a class function inline Private member functions, memory allocation for objects, static data members, static member functions Array of objects, objects as function arguments, friendly functions, returning objects, pointers to members Constructors-default, parameterised, copy, constructor overloading, dynamic constructors Operator overloading-unary, binary operators, rules for overloading operators, type conversion	**15**

	Unit III: Inheritance, Templates and Exception handling Reusability, Extendibility, class, derived class, Inheritance-single, multiple, hierarchical, multilevel, hybrid, visibility modifiers in inheritance, making private member inheritable, protected visibility modifier, Virtual base classes, abstract classes, Constructors in derived class, Compile time vs runtime polymorphism, pointers to objects, this pointer, pointers to derived classes, virtual functions, rules for virtual functions, pure virtual functions, Templates and Exception Handling Generic programming, class template, function template, member function templates Errors and exceptions, basics of exception handling, exception handling mechanism, throwing mechanism, catching mechanism, multiple catch statements, catching all exceptions, rethrowing exception	**15**
	Practical Work WEEK WISE	
	Week 1 & 2: Basic programs in C++, saving, compiling, executing programs with main () function with simple Input, output, comments, iostream.h file, basic data types, variable declaration, basic keywords, basic program constructs (if else, while, do while, switch, for) Example: 1. Display Helloworld in C++ 2. Display size in bytes of basic datatypes using sizeof operator 3.Calculate sum and average of two numbers inputted 4. Convert temperature in Fahrenheit to Celcius 5. Display largest of two numbers inputted (if else) 6. Display if a number entered is even or odd (if else) 7. Display suitable message for choosing a favourite number from 1-5. (switch case) 8. Display entered digit in words (switch case) 9. Display a message “Well done” n number of times (while/do while/for) 10. Display factorial of a number entered (while/do while/for)	**4**
	Week 3 & 4: Programs implementing C++ functions, main () function, function prototyping, call by value, call by reference, return by reference, inline functions, default arguments, constant arguments, function overloading, library functions Example 1. Display simple message using Inline function 2. Swap two numbers with call by value, call by reference 3. Function with default arguments egrectarea () with default value of breadth. 4. Function with constant arguments eg.circlearea () with const value of pi 5. Overloading area () to calculate area of various shapes based on arguments. 6. Library functions eg.math.h	**4**

	Week 5 & 6: Programs implementing structure , class, data and functions, class instance, visibility modifiers, defining functions inside and outside class, inline functions, static data members, static member functions Example 1. Revise simple program with a structure. 2. Simple i/o program with a class, accessing class instance in main (). 3. Defining functions inside and outside class. 4. Testing visibility modifiers public and private 5. Making a class function inline 6. Display count as static data member to count instances of a class 7. Demonstrate static member functions to access static data members	**4**
	Week 7, 8 & 9: Programs implementing Array of objects, objects as function arguments, friendly functions, returning objects, pointers to members Constructors-default, parameterised, copy, constructor overloading, dynamic constructors, destructors Operator overloading-unary, binary operators, rules for overloading operators, type conversion Example 1. Demonstrate i/o for array of class objects eg employees 2. Passing object argument to a function eg salary function passing employee object as argument, Returning object from a function 3. Demonstrate friend function egaverage () to calculate average of students marks 4. Demonstrate pointers to members eg.update () marks of student 5. Demonstrate constructors: default, parameterised, copy 6. Program to overload class constructors 7. Demonstrate constructors with default arguments 8. Initialise objects dynamically with constructors 9. Demonstrate destructor 10. Overload unary operator – to negate the sign of the object number. 11. Overload unary operator ++ to increment an objects data. 12. Overload binary operator + to add two complex numbers 13. Overload binary operator - to add two find distance between two points p1 (x1, y1) and p2 (x2, y2) 14. Overload binary operator < to compare two objects 15. Overload the + operator to concatenate two string class objects	**6**
	Week 10 & 11: Programs implementingInheritance Reusability, Extendibility, class, derived class, Inheritance-single, multiple, hierarchical, multilevel, hybrid, visibility modifiers in inheritance, making private member inheritable, protected visibility modifier, Virtual base classes, abstract classes, Constructors in derived class Example	**4**

	1. Program with single inheritance eg. Class parent, child 2. Program with private, public visibility modifiers in single inheritance 3. Program with protected visibility modifier in single inheritance 4. Program with multiple inheritance eg. Class parent1, parent2, child 5. Program with multilevel inheritance eg. Class grandparent, parent, child 6. Program with hierarchical inheritance eg. Class parent, child1, child2 or employee, skilled, unskilled 7. Program with hybrid inheritance eg. Class grandparent, parent1, parent2, child 8. Program with virtual base class 9. Simple program with abstract class eg class figure, circle, rectangle 10. Single inheritance program with base and derived class constructors and destructors.	
	Week 12 & 13: Programs implementing Compile time vs runtime polymorphism, pointers to objects, this pointer, pointers to derived classes, virtual functions, pure virtual functions Example 1. Simple program with pointer to object 2. Program using this pointer to access class members 3. Program with pointer to derived class 4. Using base class pointer to access derived class object 5. Program with virtual function 6. Program with pure virtual function	**4**
	Week 14 & 15: Programs implementing class template, function template, member function templates, Errors and exceptions, basics of exception handling, exception handling mechanism, throwing mechanism, catching mechanism, multiple catch statements, catching all exceptions, rethrowing Example 1. Program with function template with single parameter eg. swap () 2. Program with function template with multiple parameters eg. display () 3. Program with class template with single parameter 4. Program with class template with multiple parameters 5. Program with member function templates 6. Simple program with division by 0 exception 7. Program handling division by 0 exception with try catch block 8. Program with multiple catch statements 9. Program to demonstrate throw keyword	**4**

Pedagogy:	**Suggested strategies for use to accelerate the attainment of the various course outcomes.** 1 The lecture method need not be only a traditional lecture method, but alternative effective teaching methods could be adopted to attain the outcomes. You may use a Video/Animation to explain various concepts. b Collaborative, Peer, Flipped Learning, etc. 2 Ask at least three HOT (Higher-Order Thinking) questions in the class, which promotes critical thinking. 3 Adopt Problem Based Learning (PBL), which fosters students' Analytical skills, and develops design thinking skills such as the ability to design, evaluate, generalize, and analyze information rather than simply recall it. 4 Introduce Topics in manifold representations. 5 Show the different ways to solve the same problem and encourage the students to come up with their own creative ways to solve them. 6 Discuss how every concept can be applied to the real world - and when that's possible, it helps improve the students' understanding 7 To promote self-learning, give at least one assignment (equivalent to 50% assignment weightage) where they can complete one MOOCs (certificate or equivalent) course out of lecture hour. Test their understanding through quizzes or presentations.
References/ Readings:	Main Reading: 1. Object Oriented Programming with C++, E Balagurusamy, 8th Edition, Tata McGraw-Hill Publishing Company Limited Additional Reading: 2. The C++ Programming Language, Bjarne Stroustrup, 4th Edition, Pearson Publication 3. C++ Primer, Stanley Lippman, 4th Edition, Pearson Publication
Course Outcomes:	On completion of the course, students will be able to: 1. Remember the basic concepts & terminologies of Object-Oriented Programming. 2. Understand basic computing concepts in C++ programming language. 3. Apply Object Oriented Programming concepts in designing solutions to simpler problems using algorithm, flowchart and pseudocode. 4. Code, debug and analyze a well-structured programming logic using C++.

Name of the Programme : B.Sc. Computer Science
Course Code : CSC-201
Title of the Course : Mathematical Foundations for Computer Science
Number of Credits : 4 (3T+1P)
Effective from AY : 2024-25 (Revised version of 2023-24)

Pre-requisites for the Course:	Basic Computer Programming	
Course Objectives:	1. Students will be able to construct truth tables for complex propositional expressions, identifying tautologies, contradictions, and contingent statements. They will also gain the ability to write programs that effectively evaluate propositional expressions using logical operators. 2. Students will develop the skills to translate English sentences into predicate logic, determining the validity of predicate logic expressions. They will also be capable of implementing programs to evaluate predicate logic statements. This objective focuses on building a strong foundation in predicate logic and logical inference. 3. Students will acquire the ability to perform set operations, analyze properties of binary relations, and implement closure operations on relations. They will also identify various types of functions from given examples. This objective emphasizes the practical application of mathematical concepts in sets, relations, and functions. 4. Students will demonstrate proficiency in solving problems related to graph representations and implementing basic graph algorithms such as DFS, BFS, and Dijkstra's Algorithm. This objective aims to develop students' skills in algorithmic problem-solving within the context of graph theory.	
Content:		**No. of Hours**
	Unit-1: **Mathematical Logic:** Statements and notations, Connectives, Well-formed formulas, Truth Tables, tautology, converse, inverse and contrapositive, equivalence, implication, Logical identities, Normal forms. **Predicates:** Rules of inference, Consistency, **Predicate calculus:** Free and bounded variable, **Quantifiers:** Universal Quantifiers, Existential Quantifiers. **Strings and their Properties**	**15**
	Unit-II: **Sets:** Sets and Subsets, Sets with One Binary Operation, Sets with Two Binary Operations **Relations:** Relations, Properties of binary Relations, Types of relations: equivalence, compatibility and partial ordering relations, Hasse diagram. Lattices and its properties, Closure of Relations **Functions:** introduction to Functions, types of functions. **Graph Theory:** Representation of Graph, DFS, BFS, Dijkstra's Algorithm Spanning Trees, planar Graphs, Trees,	**15**
	Unit-III: **Automata, Computability, and Complexity**: Complexity Theory, Computability Theory, Automata Theory	**15**

	Definitions, Theorems and Proofs **Types of Proof**: By Construction, Contradiction, Induction	
	Practical	
	Week 1 & 2 a. Constructing truth tables for given propositional expressions. b. Identifying tautologies, contradictions, and contingent statements. c. Writing a simple program to evaluate propositional expressions.	**4**
	Week 3 & 4 a. Translating English sentences into predicate logic. b. Determining the validity of predicate logic expressions. c. Implementing a program to evaluate predicate logic statements.	**4**
	Week 5 & 6 a. Working with sets and performing set operations. b. Analyzing properties of binary relations. c. Implementing closure operations on relations.	**4**
	Week 7, 8 & 9 a. Identifying types of functions from given examples. b. Solving problems related to graph representations. c. Implementing basic graph algorithms (e.g., DFS, BFS, Dijkstra's Algorithm).	**6**
	Week 10 & 11 a. Practicing different types of proofs by construction, contradiction, and induction. b. Proving theorems related to the course content.	**4**
	Week 12 & 13 Applying mathematical foundations to solve a real-world problem.	**4**
	Week 14 & 15 Applying mathematical foundations to solve a real-world problem.	**4**
Pedagogy:	**Suggested strategies for use to accelerate the attainment of the various course outcomes.** 1. The lecture method need not be only a traditional lecture method, but alternative effective teaching methods could be adopted to attain the outcomes. You may use a. Video/Animation to explain various concepts. b. Collaborative, Peer, Flipped Learning, etc. 2. Ask at least three HOT (Higher-Order Thinking) questions in the class, which promotes critical thinking. 3. Adopt Problem Based Learning (PBL), which fosters students' Analytical skills, and develops design thinking skills such as the ability to design, evaluate, generalize, and analyze information rather than simply recall it. 4. Introduce Topics in manifold representations. 5. Show the different ways to solve the same problem and encourage the students to come up with their own creative ways to solve them. 6. Discuss how every concept can be applied to the real world - and	

	when that's possible, it helps improve the students' understanding 7. To promote self-learning, give at least one assignment (equivalent to 50% assignment weightage) where they can complete one MOOCs (certificate or equivalent) course out of lecture hour. Test their understanding through quizzes or presentations.
References/ Readings:	Main Reading: 1. Donald F. Stanat and David F. McAllister, Discrete mathematics in Computer Science. Additional Reading: 2. Mishra, K. L. P., & Chandrasekaran, N. (2006). Theory of computer science: automata, languages and computation. PHI Learning Pvt. Ltd. 3. Sipser, M. (2012). Introduction to the Theory of Computation Third Edition. Cengage Learning.
Course Outcomes:	On completion of the course, students will be able to: 1. Understand truth tables for complex propositional expressions, identify tautologies, contradictions, and contingent statements and write programs to evaluate propositional expressions using logical operators 2. Apply to translate English sentences into predicate logic, determine the validity of predicate logic expressions, and implement programs to evaluate predicate logic statements. 3. Perform set operations, analyze properties of binary relations, and implement closure operations on relations 4. Solve problems related to graph representations and implement basic graph algorithms

Name of the Programme : **B.Sc. Computer Science**
Course Code : **CSC-211**
Title of the Course : **Problem Solving and Programming**
Number of Credits : **4 (3T + 1P)**
Effective from AY : **2024-25 (Revised version of 2023-24)**

Pre-requisites for the Course:	None	
Course Objectives:	1. To understand the concepts and techniques of problem solving. 2. To analyze, understand, and build logic to solve basic problems. 3. To design Algorithms and flowcharts for better understanding and documentation for accurate implementation of the problem. 4. To code and implement a well-structured, robust programming logic using a suitable programming language.	
Content:		**No. of Hours**
	Unit-1: Introduction to Problem Solving Problem Solving Life Cycle — Understanding the Problem Statement, Analyzing the problem, Planning Program design using Hierarchy charts, Expressing Program logic using flowcharts / Pseudocode. Structured Programming concept Modular Programming - Top-Down design, Bottom-up design, Stepwise Refinement **Understanding basic Problem Solving Tools** Algorithms: Definition and its attributes, algorithm constructs, Statements: Input-Output, Decision-making, and Looping, Examples Flowchart: Definition & its attributes, symbols, Statements: Input-Output, Decision-Making & Looping, Module representation, Drawing conventions and standards, Examples. Pseudo-code: Definition and its attributes, constructs, and Examples **Basic Program Structures** Data and its types (Integer, Floating-point, Character, String), Constants and variables, scope, Instructions and their types, how the computer stores data, Operators (Arithmetic, Assignment, Relational, Logical, etc), Expressions and Equations, Evaluation of expressions, Keywords. Local and Global Variables, Parameters, Return Values, naming conventions and standards, Understanding literals, syntax and semantics, functions, and modules.	**15**
	Unit-II: Basic Sequential Instructions Sequential statements using operators, constants, variables, operands, expressions, and equations. ***Activity:*** *Apply the concepts learnt to design the algorithms of at-least 2 basic problems. Represent it using flowchart and pseudo-code.* **Debugging & Documentation** Definition, Types, Need, and how to do it. **Problem Solving with Decisions**	**15**

	The Decision Logic Structure, Multiple If/Then/Else Instructions, Using Straight-Through Logic, Using Positive and Negative Logic, Logic Conversion, Decision Tables, Case Logic Structure. ***Activity:*** *Apply the concepts learnt to design the algorithms of at-least 4 basic problems. Represent it using flowchart and pseudo-code.*	
	Unit-III: Problem Solving with Loops The Loop Logic Structure, Incrementing, Accumulating, While/While End, Repeat/Until, Automatic-Counter Loop, Nested Loops, Indicators (flags). Iterating, accessing, and modifying array elements. ***Activity:*** *Apply the concepts learnt to design the algorithms of at-least 3 basic problems. Represent it using flowchart and pseudo-code.* **Problem Solving with Arrays** Arrays Concepts: One-dimensional Arrays, Creating, Concept of Strings, String as an array of characters. ***Activity:*** *Apply the concepts learnt to design the algorithms of at-least 3 basic problems. Represent it using flowchart and pseudo-code.* **Understanding functions** Functions: Definition and its need and constructs, designing simpler functions, function communication using arguments and return statements. scope of functions, function declaration and prototype, call by Value, and Call by reference. Concept of Recursive functions: why, when, and how. Designing recursive functions and recursive calls. Base case and recursive case. *Apply the concepts learnt to design the algorithms of at-least 3 basic problems. Represent it using flowchart and pseudo-code.*	**15**
	Unit-IV: Practical Work Using any suitable programming language like C, the concepts learned in the units from I to III are required to be implemented practically. The broad area of practical problems is mentioned/ suggested below.	

	Week 1 & 2 [These practical should be done using pen, paper, and using buddy learning strategy]: For each of the following tasks, write a set of numbered, step-by-step instructions (a solution) so complete that another person can perform the task without asking questions. Define the knowledge base of this person by listing what you expect the person to know in order to follow your directions. For example, for task “a” (below), make a cup of cocoa, the knowledge base might include such things as knowledge of milk or water, a refrigerator, pan, spoon, cocoa, cup, range top or microwave, and so forth. a. Make a cup of cocoa. b. Sharpen a pencil. c. Walk from the classroom to the student lounge, your dorm, or the cafeteria. d. Start a car (include directions regarding what to do if the car doesn’t start). e. Get a glass of water from your kitchen. f. Start your computer. Test your solution in problem 1 by giving your instructions to another person to see whether he or she can accomplish the task without your help. If they can’t, modify your solution so that the person can accomplish the task. Check the solution again by giving the instructions to another person.	**04**
	Week 3 & 4: Basic Program Structures a. At-least 10 basic programming problems related to Module II are to be completed during the practical sessions. b. More programs may be given to the learners to complete and practice as part of their Practice Work.	**04**
	Week 5 & 6: Basic Sequential Instructions a. At least 08 programming problems are to be completed during the practical sessions. b. More programs may be given to the learners to complete and practice as part of their Practice Work. Debugging & Documentation a. Debug & Document at-least 02 problems. b. More programs may be given to the learners to complete and practice as part of their Practice Work.	**04**
	Week 7, 8 & 9: Problem Solving with Decisions a. At least 08 programming problems are to be completed during the practical sessions. b. Debug & Document at-least 02 problems. c. More programs may be given to the learners to complete and practice as part of their Practice Work.	**06**

	Week 10 & 11: Problem Solving with Loops a. At least 08 programming problems are to be completed during the practical sessions. b. Debug & Document at-least 02 problems. c. More programs may be given to the learners to complete and practice as part of their Practice Work.	**04**
	Week 12 & 13: Understanding functions a. At least 08 programming problems are to be completed during the practical sessions. b. Debug & Document at-least 02 problems. c. More programs may be given to the learners to complete and practice as part of their Practice Work.	**04**
	Week 14 & 15: Problem Solving with Arrays a. At least 08 programming problems are to be completed during the practical sessions. b. Debug & Document at-least 02 problems. c. More programs may be given to the learners to complete and practice as part of their Practice Work.	**04**
Pedagogy:	**Suggested strategies for use to accelerate the attainment of the various course outcomes.** 1. The lecture method need not be only a traditional lecture method, but alternative effective teaching methods could be adopted to attain the outcomes. You may use a. Video/Animation to explain various concepts. b. Collaborative, Peer, Flipped Learning, etc. 2. Ask at least three HOT (Higher-Order Thinking) questions in the class, which promotes critical thinking. 3. Adopt Problem Based Learning (PBL), which fosters students' Analytical skills, and develops design thinking skills such as the ability to design, evaluate, generalize, and analyze information rather than simply recall it. 4. Introduce Topics in manifold representations. 5. Show the different ways to solve the same problem and encourage the students to come up with their own creative ways to solve them. 6. Discuss how every concept can be applied to the real world - and when that's possible, it helps improve the students' understanding 7. To promote self-learning, give at least one assignment (equivalent to 50% assignment weightage) where they can complete one MOOCs (certificate or equivalent) course out of lecture hour. Test their understanding through quizzes or presentations.	
References/ Readings:	Main Reading: 1. Maureen Sprankle, Jim Hubbard (2013). Problem Solving and Programming Concepts. Pearson Education India. Additional Reading: 2. S. Kuppuswamy, S. Malliga, C. S. KanimozhiSelvi, K. Kousalya (2019). Problem Solving and Programming. Tata McGraw Hill. 3. Behrouz A. Forouzan, Richard F. Gilberg (2007) A Structured Programming Approach Using C. Cengage Learning India.	

Course Outcomes:	On completion of the course, students will be able to: 1. Remember the basic concepts & terminologies of problem solving, algorithms, flowcharts, pseudo-code, language syntax, and debugging. 2. Understand basic computing concepts, algorithm design, flowchart design, pseudo-code, programming constructs, and debugging. 3. Apply problem solving & programming concepts in designing solutions to simpler problems using algorithm, flowchart and pseudocode. 4. Code, debug and analyze a well-structured programming logic using suitable programming language/s.

Name of the Programme : B.Sc. Computer Science
Course Code : CSC-212
Title of the Course : Office Administration
Number of Credits : 4 (3T+1P)
Effective from AY : 2024-25 (Revised version of 2023-24)

Pre-requisites for the Course:	None	
Course Objectives:	1. To provide practical skills in text editing with report formatting, effective presentation tools. 2. To impart basic skills on the spreadsheet package. 3. To acquire practical knowledge on the Internet, efficient search techniques and online collaboration tools.	
Content:		**No. of Hours**
	Unit-I: **Computer: An Introduction-** Computer in office, Elements of Computer System Set-up. Understanding office automation software and its applications. Types of office automation software: Microsoft Office Suite, Google Workspace. **Introduction to Microsoft Word**. Working with Documents - Opening & Saving files, Editing text documents, Inserting, Deleting, Cut, Copy, Paste, Undo, Redo, Find, Search, Replace, Formatting page & setting Margins, Converting files to different formats, Importing & Exporting documents, Sending files to others, Using Tool bars, Ruler, Using Icons, using help, Formatting Documents - Setting Font styles, Font selection- style, size, colouretc, Type face - Bold, Italic, Underline, Case settings, Highlighting, Special symbols, Setting Paragraph style, Alignments, Indents, Line Space, Margins, Bullets & Numbering. Setting Page style - Formatting Page, Page tab, Margins, Layout settings, Paper tray, Border & Shading, Columns, Header & footer, Setting Footnotes & end notes – Shortcut Keys; Inserting manual page break, Column break and line break, Creating sections & frames, Anchoring & Wrapping, Setting Document styles, Table of Contents, Index, Page Numbering, date & Time, Author etc., Creating Master Documents, Web page. Creating Tables- Table settings, Borders, Alignments, Insertion, deletion, Merging, Splitting, Sorting, and Formula, Drawing - Inserting ClipArts, Pictures/Files etc., Tools – Word Completion, Spell Checks, Mail merge, Templates, Creating contents for books, Creating Letter/Faxes, Creating Web pages, Using Wizards, Tracking Changes, Security, Digital Signature. Printing Documents – Shortcut keys.	**15**
	Unit-II: **Introduction to MS Office – MS Excel**: Spread Sheet & its Applications, Opening Spreadsheet, Menus - main menu, Formula Editing, Formatting, Toolbars, Using Icons, Using help, Shortcuts, Spreadsheet types. Working with Spreadsheets- opening, Saving files, setting Margins, Converting files to different formats (importing, exporting, sending files to others), Spread sheet addressing - Rows, Columns & Cells, Referring Cells & Selecting	**15**

	Cells – Shortcut Keys. Entering & Deleting Data- Entering data, Cut, Copy, Paste, Undo, Redo, Filling Continuous rows, columns, Highlighting values, Find, Search & replace, Inserting Data, Insert Cells, Column, rows & sheets, Symbols, Data from external files, Frames, Clipart, Pictures, Files etc, Inserting Functions, Manual breaks, Setting Formula - finding total in a column or row, Mathematical operations (Addition, Subtraction, Multiplication, Division, Exponentiation), Using other Formulae. Formatting Spreadsheets- Labelling columns & rows, Formatting- Cell, row, column & Sheet, Category - Alignment, Font, Border & Shading, Hiding/ Locking Cells, Anchoring objects, Formatting layout for Graphics, Clipart etc., Worksheet Row & Column Headers, Sheet Name, Row height & Column width, Visibility - Row, Column, Sheet, Security, Sheet Formatting & style, Sheet background, Colouretc, Borders & Shading – Shortcut keys. Working with sheets – Sorting, Filtering, Validation, Consolidation, and Subtotal. Creating Charts - Drawing. Printing. Using Tools – Error checking, Spell Checks, Formula Auditing, Creating & Using Templates, Pivot Tables.	
	Unit-III: **Introduction to MS Office-MS Power Point** MS Power point: Introduction to presentation – Opening new presentation, Different presentation templates, setting backgrounds, Selecting presentation layouts. Creating a presentation - Setting Presentation style, Adding text to the Presentation. Formatting a Presentation - Adding style, Colour, gradient fills, arranging objects, Adding Header & Footer, Slide Background, Slide layout. Adding Graphics to the Presentation- Inserting pictures, movies, tables etc into presentation, Drawing Pictures using Draw. Adding Effects to the Presentation- Setting Animation & transition effect. Printing Handouts, Generating Standalone Presentation viewer. **Internet and Web Browsers**: Definition & History of Internet - Uses of Internet - Definition of Web Addressing-URL- browsers and its types, internet browsing, searching - Search Engines - Portals - Social Networking sites- Blogs - viewing a webpage, downloading and uploading the website; Creating an email-ID, e-mail reading, saving, printing, forwarding and deleting the mails, checking the mails, viewing and running file attachments, addressing with cc and bcc. Introduction to various devices & Applications other than the computers, (electronic gadgets), which are widely using by executives in the Offices: Tablet, Smart Phone – concept of mobile phone and Tablet and their uses – Modern forms of communication: Email, video conferencing, zoom, Webex, Google Meet, Teams, Skype, Whatsapp etc.	**15**
	Practical Work	
	Week 1 & 2 **Practical on Word Processing:** Bulleted and numbered list, headers and footers, page numbering.-- Creation of tables , Mail Merge, Draft an official letter for job interview invitation/ job appointment/ invitation to an event, use mail merge to input the recipients list linking with database.	**4**

	Week 3 & 4 Given a project report in PDF format transfer to word processor software and format to include title page, specified Paragraph and Page Formatting (page size, orientation, line spacing, font type and font size, Indent, bullets, paragraph formatting) details, Acknowledgement page, Design a certificate in landscape orientation with a border around the document.	**4**
	Week 5 & 6 **Practical on Spreadsheet** Formatting of cells, rows and columns--Presenting data with Charts. Practical on introduction to spreadsheet using simple tabular data and formatting using paste special, absolute and relative cell references	**4**
	Week 7&8&9 Calculating sum, average, min, max, count and percentage. Practical on different types of charts and pivot table with suitable example	**6**
	Week 10 & 11 **Practical on Presentation software** Usage of text, images and animation for presentation-- Adding slide transition, custom animation, set up show. Creating graphs in presentation.	**4**
	Week 12 & 13 Preparing presentations in areas such as the Impact of Social Media on Youth and emerging trends in Mobile Technology include appropriate slide animation, slide transitions, sound recording, slide timings. Export the presentation as video or save it as slide show. Prepare handouts for the audience.	**4**
	Week 14 & 15 **Practical on Internet browsing**, downloading files, knowing secure browsing. Practical on Email account creation, sending emails, attachments	**4**
Pedagogy:	Suggested strategies to use to accelerate the attainment of the various course outcomes. 1. Lecture method need not be only a traditional lecture method, but alternative effective teaching methods could be adopted to attain the outcomes. You may use a. Video/Animation to explain various concepts. b. Collaborative, Peer, Flipped Learning etc. 2. Ask at least three HOT (Higher-order Thinking) questions in the class, which promotes creative thinking. 3. Adopt Problem Based Learning (PBL), which fosters students' Analytical skills, develop design thinking skills such as the ability to design, evaluate, generalize, and analyse information rather than simply recall it. 4. Introduce Topics in manifold representations. 5. Show the different ways to solve the same problem and encourage the students to come up with their own creative ways to solve them. 6. Discuss how every concept can be applied to the real world- and when that's possible, it helps improve the students' understanding 7. To promote self-learning give at least one assignment (equivalent to 50% assignment weightage) where they can complete at least one MOOC (certificate or equivalent) course out of lecture hour. Test their understanding through quizzes or presentations.	

	8. Activity/ Practical Based Learning (Suggested Activities in Class) a. Real world problem solving using group discussion. E.g., designing poster for road safety etc., b. Demonstration of solution to a problem through design. 9. Demonstration of simple project and motivating the students to develop similar types of projects.
References/ Readings:	Main Reading: 1. Discovering Computers 2022: Digital Technology, Data, and Devices by Misty E. Vermaat. Additional Reading 2. A+ Guide to IT Technical Support (MindTap Course List) by Jean Andrews 3. Shelly, G. B., &Vermaat, M. E. (2017). Microsoft Office 365 & Office 2016: Introductory. Cengage Learning. 4. Russel, C., & Hoque, M. R. (2018). Google Workspace for Dummies. John Wiley & Sons. 5. Meyers, M. (2017). CompTIA A+ Certification All-in-One Exam Guide, Ninth Edition (Exams 220-901 & 220-902). McGraw Hill Professional.
Course Outcomes:	On completion of the course, students will be able to 1. Remember the basic concepts of computer and their application. 2. Understand the use and various functions of spreadsheets. 3. Create and format documents, create and format tables and mail merge. 4. Apply the knowledge of tools to create effective presentations, use the latest Internet technologies in office administration.

Name of the Programme	**: B.Sc. Computer Science**
Course Code	**: CSC-213**
Title of the Course	**: Computer Organization**
No. of credits	**: 4 (3T + 1P)**
Effective from AY	**: 2024-25 (Revised version of 2023-24)**

Pre-requisites for the Course:	None	
Course Objectives:	1. Conceptualize the basics of Computer Organizational and Architectural issues and classify the computers based on performance and machine instructions. 2. Learn various data transfer techniques and the I/O interfaces 3. Estimate and compare performances of various classes of memory 4. Understand the basics of ALU implementation, hardwired and microprogrammed control units, pipelining and parallel architectures	
Content:		**No of hours**
	Unit-I: Introduction to Logic Gates and Boolean Algebra: Logic Gates, Boolean Algebra, Combinational circuits, Karnaugh Map Data representation: Data Type Representation, Number System, Signed number, fixed, floating point, character representation, Addition, Subtraction, Multiplication - Shift and Add, Booth's Algorithm, Division, Pseudo-code: Definition and its attributes, constructs, and Examples Memory Hierarchy: Hierarchical memory organization, Types of Memory-internal and external, Cache memory, Memory interleaving,	**15**
	Unit-II: Introduction to Computer Architecture: Introduction to Computer Architecture, Flynn's Classification of Computers, Performance Metrics (like Latency, throughput), Fundamental Blocks of Computer (like CPU, I/O subsystems, memory, control unit), computer function, interconnection structures, Bus interconnections Peripheral devices: Types of Peripheral Devices, I/O subsystem, programmed I/O, Interrupt-driven I/O, DMA, I/O channels and processors	**15**
	Unit-III: Instruction Set Architecture (ISA): Introduction to Instruction Set, Types of ISA; RISC, CISC; Processor Organization, Registers organization, Instruction Execution Cycle, Instruction formats, Addressing Modes; Register Transfer Language (RTL), Assembly Language Programming, X86-Architecture, ARM Architecture	**15**
	Practical Work Writing assembly language programs in 8086 using MASM or compatible assembler either in Windows or Linux.	
	Week 1 & 2	**04**

	a. Introduction to 8086 architecture and instruction set b. Find the sum of 1 + 2 + 3 + …………..+ n	
	Week 3 & 4 c. Display the multiplication table of a number d. Store and retrieve numbers from memory	**04**
	Week 5 & 6 e. Block Transfer f. Block Transfer in reverse order	**04**
	Week 7, 8 & 9 g. Sort the numbers stored in the memory (Any two methods) h. Searching methods	**06**
	Week 10 & 11 i. Masking of bits j. Counting of number of bits	**04**
	Week 12 & 13 k. Count the number of even or odd numbers from a given set of numbers l. Check if the number is a palindrome	**04**
	Week 14 & 15 m. Count the number of positive and negative numbers from a given set of numbers n. Generate a series like 1, 3, 5, 7, …….. up to n terms	**04**
Pedagogy:	Suggested strategies for use to accelerate the attainment of the various course outcomes.: Lectures, Tutorials, Collaborative/peer learning, Hands-on assignments	
References/ Readings:	Main Reading 1. John L. Hennessy & David Patterson "Computer Architecture: A Quantitative Approach", Morgan Kaufmann, 5th Edition Additional Reading 2. William Stallings, "Computer Organization and Architecture: Designing for performance", Prentice Hall of India, 9th Edition.	
Course Outcomes:	On completion of the course, students will be able to: 1. Explain the theory and architecture of central processing unit, I/O and memory organization 2. Understand the concepts of parallel processing, pipelining and inter-processor communication. 3. Represent different number systems, and perform various binary operations. 4. Analyze some of the design issues in terms of speed, technology, cost, performance, CPU architecture.	

Name of the Programme : B.Sc. Computer Science
Course Code : CSC-231
Title of the Course : Web Designing
Number of Credits : 3T
Effective from AY : 2024-25

Pre-requisites for the Course:	Basic usage of computer	
Course Objectives:	1. Equip students with the fundamental knowledge of web design concepts and principles. 2. Develop an understanding of website development technologies and tools. 3. Explore the latest trends and advancements in web design, including AI-powered tools and accessibility concerns. 4. Critically analyze the user experience (UX) and visual design aspects of websites.	
		No. of Hours
Content:	**Unit-I:** **Introduction to Web Design** The Internet and The World Wide Web: History, evolution, and impact. Building Blocks of Websites: HTML, CSS, and JavaScript (basic overview). Design Principles and Elements: Visual design, layout, typography, colour theory, usability. User Experience (UX): User centered design, information architecture, interaction design. Accessibility for the Web: W3C guidelines, assistive technologies, inclusive design. AI Tools for Web Design: Website generators, design assistants, future trends. Case Studies: Analysing successful websites for design principles and user experience.	**15**
	Unit-II: **Web Design Concepts** Responsive Web Design (RWD): Designing for different devices and screen sizes. Content Management Systems (CMS): WordPress, WIX and their applications. Search Engine Optimization (SEO): Optimizing websites for search engine visibility. Web Performance Optimization: Website loading speed, file optimization, user experience. E-commerce Web Design: Designing online stores for functionality and usability. Web Security: Website security vulnerabilities, protection measures, ethical hacking. Emerging Trends in Web Design: AR/VR, voice interfaces, single-page applications (SPAs). Project Planning: Defining scope, target audience, project timeline, and resources.	**15**

	Unit-III: **The Business of Websites** Basics of Domain and Hosting: Understanding domain names and extensions, Selecting a hosting provider Website Maintenance and Updates: Importance of regular updates, Handling backups and security patches Introduction to SEO: Basics of Search Engine Optimization, Strategies for improving website visibility Legal Considerations in Web Design: Copyright and fair use, Privacy policies and terms of service.	**15**
Pedagogy:	**Suggested strategies for use to accelerate the attainment of the various course outcomes.** i. Interactive lectures: Use multimedia presentations, videos, and case studies to engage students. ii. Guest lectures: Invite industry professionals to share real-world experiences and insights. iii. Group discussions and activities: Encourage critical thinking and collaboration through discussions and exercises. iv. Case studies and website analysis: Analyse existing websites to apply theoretical concepts to real-world scenarios. v. Project-based learning: Assign individual or group projects to implement learned concepts and develop practical skills. vi. Online resources and tools: Utilize online platforms, web design tools, and simulations for virtual learning experiences.	
References/ Readings:	Main Reading 1. Ralph Mason, The Principles of Beautiful Web Design, Fourth Edition (Grayscale Indian Edition) Paperback 2. Laura Lemay, Rafe Colburn, Jennifer Kyrnin, “Mastering HTML, CSS and JavaScript Web Publishing”, BPB Publications Additional Reading 3. Alex Libby, Gaurav Gupta, AsojTalesra, “Responsive Web Design with HTML5 and CSS3 Essentials”, PACKT Publishing 4. Steven Branson, “UX / UI Design: Introduction Guide To Intuitive Design and User-Friendly Experience” Paperback	
Course Outcomes:	On completion of the course, students will be able to : 1. Recall and list key web design terminology, principles, and tools. 2. Explain the functionalities and purposes of different web development technologies like HTML, CSS, and JavaScript 3. Apply different types features and functionalities of static and dynamic sites, content management systems, and e-commerce platforms. 4. Analyse existing websites based on UX principles and accessibility guidelines.	

Name of the Programme : B.Sc. Computer Science
Course Code : CSC-232
Title of the Course : Application Software for Social Science
Number of Credits : 3T
Effective from AY : 2024-25 (Revised version of 2023-24)

Pre-requisites for the Course:	None	
Course Objectives:	1. To provide students with a comprehensive understanding of various application software tools commonly used in social science research. 2. How to use these tools to collect, analyze, and visualize data in the context of social science research. 3. Develop practical skills in data collection, analysis, and visualization using popular software tools.	
Content:		**No. of Hours**
	Unit-I: **Introduction to Social Science Research and Software Tools:** Overview of social science research methodologies Introduction to statistical analysis in social science Importance of software tools in social science research Overview of popular application software tools in social science research (e.g., SPSS, R, Python) **Data Collection and Management:** Data collection methods in social science Survey design and implementation Data management techniques Experience with data collection tools (Qualtrics, SurveyMonkey)	**15**
	Unit-II: **Statistical Analysis with SPSS:** Introduction to SPSS (Statistical Package for the Social Sciences) Descriptive statistics and data exploration Inferential statistics and hypothesis testing Regression analysis in SPSS Exercises using SPSS **Data Analysis with R** Introduction to R programming language Data manipulation and cleaning in R Basic and advanced statistical analysis using R Data visualization with ggplot2	**15**
	Unit-III: **Python for Social Science Research:** Introduction to Python for data analysis Using Python libraries (pandas, numpy) for social science research Data visualization with matplotlib and seaborn	**15**

	Text analysis and sentiment analysis in Python **Qualitative Data Analysis:** Overview of qualitative research methods Introduction to qualitative data analysis software (NVivo, ATLAS.ti) Coding and thematic analysis **Geographic Information Systems (GIS) in Social Science:** Introduction to GIS and its applications in social science Spatial data analysis Exercises with GIS tools (ArcGIS, QGIS) Mapping and visualization of social science data	
Pedagogy:	**Suggested strategies for use to accelerate the attainment of the various course outcomes.** 1. The lecture method need not be only a traditional lecture method, but alternative effective teaching methods could be adopted to attain the outcomes. You may use c. Video/Animation to explain various concepts. d. Collaborative, Peer, Flipped Learning, etc. 2. Adopt Problem Based Learning (PBL), which fosters students' Analytical skills, and develops design thinking skills such as the ability to design and analyze the information. 3. Show the different ways to solve the same problem. 4. To promote self-learning, give at least one assignment. 5. Test their understanding through quizzes or presentations.	
References/ Readings:	Main Reading: 1. "Designing Social Inquiry: Scientific Inference in Qualitative Research" by Gary King, Robert O. Keohane, Sidney Verba 2. "Discovering Statistics Using IBM SPSS Statistics" by Andy Field 3. "R for Data Science" by Hadley Wickham and Garrett Grolemund 4. "Python for Data Analysis" by Wes McKinney Additional Reading: 5. "Qualitative Data Analysis: A Methods Sourcebook" by Matthew B. Miles, A. Michael Huberman, Johnny Saldana. 6. "The GIS Guide to Public Domain Data" by Joseph J. Kerski 7. "Research Design: Qualitative, Quantitative, and Mixed Methods Approaches" by John W. Creswell	
Course Outcomes:	On completion of the course, students will be able to: 1. Remember the concepts, steps and terminologies used in Application Software for Social Science 2. Understand basic data analysis methods used in Application Software for Social Science 3. Apply the techniques of different application softwares in analysing the data. 4. Analyze the collected data using application Software for Social Science	

Name of the Programme : **B.Sc. Computer Science**
Course Code : **CSC-233**
Title of the Course : **Application Software for Science**
Number of Credits : **3T**
Effective from AY : **2024-25 (Revised version of 2023-24)**

Prerequisites for the Course:	None	
Course Objectives:	1. Develop a foundational understanding of computational thinking and its applications in scientific problems. 2. Enable students to proficiently use Python and scientific computing libraries for scientific analysis and visualisation.	
Content:		**No. of Hours**
	Unit 1: **Foundations of Computational Thinking** Introduction to Computational Thinking; Overview of computational thinking concepts, fundamentals of algorithms for scientific problems, Programming languages and its applications for Sciences. Introduction to Python: Variables, data types, and basic operations; Control structures (if statements, loops); Functions and modular programming; Writing code for scientific calculations.	**15**
	Unit 2: **Application Software Development for Scientific Analysis** Scientific Computing Libraries; PSPP: Introduction an overview for mathematical and scientific computing; Invoking pspp; Using pspp - Preparation of Data Files, Data Screening and Transformation; Hypothesis Testing. Database Management for Sciences: MYSQL: Introduction to databases and data management; Designing and querying databases for scientific applications; Building a database for scientific research; Integration of databases with programming languages.	**15**
	Unit 3: **Data Visualization in Science** Introduction to Data Visualization and Matlab: Importance of data visualisation in scientific research; Overview of Matlab programming language and its capabilities; Basics of plotting (line plots, scatter plots, and bar charts) Customising plots (labels, titles, colours, and styles); Multiple plots and subplots; Customising axes and ticks; Plotting scientific data with error bars, Pandas: analysing, cleaning, exploring, and manipulating data.	**15**
Pedagogy:	i. Foster active participation in discussions, problem-solving, and group activities. ii. Assign interdisciplinary group projects to blend scientific expertise with computational skills. iii. Design assignments connecting theoretical concepts with practical applications. iv. Task students with creating databases or visualizing scientific data for their specific domains.	

References/ Readings:	Main Reading: 1. Kelleher, J. D., Mac Namee, B., D’arcy, A., &Proquest (Firm). (2020). *Fundamentals Of Machine Learning For Predictive Data Analytics : Algorithms, Worked Examples, And Case Studies*. 798. 2. Ramakrishnan, R., Gehrke, J., Derstadt, J., Selikoff, S., & Zhu, L. (N.D.). *Database Management Systems Solutions Manual Third Edition*. Additional Reading: 3. Valentine, D. T., & Hahn, B. D. (2022). *Essential MatlabFor Engineers And Scientists*. 4. Taneja S., & Kumar N. (2022) *Python Programming - A Modular Approach*, Pearson.
Course Outcomes:	On completion of the course, students will be able to: 1. Recall the basic concepts of computational thinking and identify the fundamentals of algorithms for scientific problems. 2. Explain the steps involved in data screening, transformation, and hypothesis testing using PSPP. 3. Apply Matlab to create various scientific plots with customization. 4. Analyse and interpret scientific data through effective data visualisation techniques.

Name of the Programme	**: B.Sc. Computer Science**
Course Code	**: CSC-234**
Title of the Course	**: Application Software for Literature**
Number of Credits	**: 3T**
Effective from AY	**: 2024-25**

Pre-requisites for the Course:	The students should have basic computer literacy and fundamental writing skills.	
Course Objectives:	1. To understand the diverse ways in which application software is used in literary studies. 2. To identify variety of software and collaboration tools relevant to literary study. 3. To understand the ethical implications of using software in literary studies. 4. To enhance students' proficiency in leveraging technology for effective literary creation.	
		No of Hours
Content:	**Unit-1:** **Introduction to Application Software** Basic Concepts History of software in literature Need of software in literary studies Features (text analysis, digital editing, electronic literature) Types: Open-Source Software, Proprietary Software Ethical considerations of using technology in literature **Document Processing Software** Basic document processing software (Typing, Formatting) Using templates for different writing projects Use of regional/foreign language keyboards for document typing **Reference Management Software** Organizing and citing sources Creating bibliographies Ethics of citation in the digital age	**15**
	Unit-2: **Writing and Publishing Software** Tools for outlining, structuring, and revising manuscripts Self-publishing and digital distribution platforms Grammar and style checking **Collaborative Writing and Translation Tools** Online platforms for writing, sharing and editing collaboratively Online language translation tools Impact of collaboration on the creative process Case studies of collaborative literary projects **Plagiarism Detection Software** Understanding plagiarism and academic integrity Using plagiarism detection software effectively Avoiding plagiarism in your own work	**15**

	Unit-3: **E-book Readers** Rise of e-books and their impact on reading practices Annotating and highlighting text **Audiobook Players** Listening to literature: audiobooks and the audio experience Choosing and using audiobook players Role of performance and interpretation in audiobooks Future of audiobooks and the changing nature of storytelling **Digital Libraries** Physical Libraries v/s Digital Libraries Online repositories of literary texts Impact on access to literature sources	**15**
Pedagogy:	● Use and demonstration of following Software Tools in class: ❖ **Word Processing Software** (Microsoft Word, Libre Office, etc.) ❖ **Reference Management Software** (Zotero, Mendeley, etc.) ❖ **Text Analysis Software** (MAXQDA, NVivo, etc.) ❖ **Writing and Publishing Software** (Scrivener, ProWritingAid, etc.) ❖ **Collaborative Writing Tools** (Google Docs, Dropbox Paper, etc.) ❖ **Translation Tools (**Google Translate, etc.) ❖ **Plagiarism Detection Software** (Turnitin, iThenticate) ❖ **E-book Readers and Libraries** (Kindle, Kobo, etc.) ❖ **Audiobook Players** (Audible, Libro.fm, etc.)	
References/ Readings:	Main Reading: 1. Bernstein, Mark (2011). Programming the Muse: The Poetics of Code and Culture. Cambridge, MA: The MIT Press. 2. Clark, R. P. (2006). Writing Tools: 50 Essential Strategies for Every Writer. Little, Brown and Company. Additional Reading: 3. Drucker, J. (2014). Graphesis: Visual Forms of Knowledge Production. Harvard University Press. 4. Moretti, F. (2005). Graphs, Maps, Trees: Abstract Models for Literary History. Verso. 5. Hayles, K. (2012). How We Think: Digital Media and Conceptual Change. The MIT Press.	
Course Outcomes:	On completion of the course, students will be able to: 1. Understand the diverse ways in which application software is used in literary studies. 2. Identify variety of software and collaboration tools relevant to literary study, including document processing, reference management and writing/publishing software. 3. Understand the ethical implications of using software in literary studies. 4. Enhance students' proficiency in leveraging technology for effective literary creation and analysis.	

Name of the Programme : B.Sc. Computer Science
Course Code : CSC-235
Title of the Course : Latex
Number of Credits : 3T
Effective from AY : 2024-25

Pre-requisites for the Course:	None	
Course Objectives:	1. Prepare an article or report document using Latex. 2. Include figures and tables in a Latex document. 3. Use the preamble of LaTeX file to define document class and layout options of Latex. 4. Use BibTeX to maintain bibliographic information and to generate a bibliography for a particular document.	
Content:		**No. of Hours**
	Unit-1: **Introduction to Latex, advantage over word processor tools document structure**: preamble, body, package, setting macros, first page, header, footer, font: size, weight, color, face, line spacing, vertical and horizontal spacing (exact and justified), geometry of page minipage, frame boxes, TOC, index page	**15**
	Unit-2: **Environments**: Enumeration, lists, itemize, verbatim, quote label and caption, width, scaling Table, tabular, row and col height and width, row and column spaning, Figure, subfigure, watermark	**15**
	Unit-3: **utils**: creating large documents, mathematical equations, bibliography, longtable, question paper hyperlinking and cross-referencing within documents, beamer - presentation document creation	**15**
Pedagogy:	**Suggested strategies for use to accelerate the attainment of the various course outcomes.** 1. The lecture method need not be only a traditional lecture method, but alternative effective teaching methods could be adopted to attain the outcomes. You may use a. Video/Animation to explain various concepts. b. Collaborative, Peer, Flipped Learning, etc. 2. Ask at least three HOT (Higher-Order Thinking) questions in the class, which promotes critical thinking. 3. Adopt Problem Based Learning (PBL), which fosters students' Analytical skills, and develops design thinking skills such as the ability to design, evaluate, generalize, and analyze information rather than simply recall it. 4. Introduce Topics in manifold representations. 5. Show the different ways to solve the same problem and encourage the students to come up with their own creative ways to solve them.	

	6. Discuss how every concept can be applied to the real world - and when that's possible, it helps improve the students’ understanding 7. To promote self-learning, give at least one assignment (equivalent to 50% assignment weightage) where they can complete one MOOCs (certificate or equivalent) course out of lecture hour. Test their understanding through quizzes or presentations.
References/ Readings:	Main Reading: 1. Frank Mittelbach, Ulrike Fischer (2023) The LaTeX Companion: Parts I & II, 3rd Edition, Addison-Wesley Professional. Additional Reading: 2. Helmut Kopka, Patrick W. Daly (2003), Guide to LaTeX (Adobe Reader), 4th Edition, Addison-Wesley Professional
Course Outcomes:	On completion of the course, students will be able to: 1. Understand the basic principles of LaTeX. 2. Structure documents using chapters, sections, and subsections and Format text with different fonts, sizes, and styles. 3. Apply mathematical equations and symbols. 4. Create tables and figures and generate bibliographies and references.

Name of the Programme	**: B.Sc. Computer Science**
Course Code	**: CSC-236**
Title of the Course	**: 3D Graphics Design**
Number of Credits	**: 3T**
Effective from AY	**: 2024-25**

Pre-requisites for the Course:	**Nil**	
Course Objectives:	1. To understand the fundamental concepts of 3D graphics and its applications in various industries 2. To understand the basics of 3D modeling and application of concepts used in 3D modeling 3. To understand the basics of Lighting and Rendering and its application in 3D graphic design 4. To understand and apply the basic concepts of animation.	
Content		**No. of Hours**
	Unit-1: Introduction to 3D Graphics 1. **Overview of 3D Graphics:** 2. Definition and History of 3D Graphics, Applications in Various Industries 3. **Understanding the 3D Space: Cartesian Coordinates:** a. Coordinate Systems in 3D, Importance of 3D Space in Graphics 4. **Introduction to 3D Modeling Software (Blender, Maya, or Cinema 4D):** Overview of Commonly Used Software, Basic Features and Interface Navigation	**15**
	Unit-2: Basic 3D Modeling 1. Primitive Shapes and Mesh Editing: a. Basics of Creating Primitive Shapes, Introduction to Mesh Editing 2. Extrusion, Beveling, and Boolean Operations: 3. Exploring Extrusion Techniques, Beveling and Boolean Operations in Modeling 4. Introduction to Materials and Textures: 5. Understanding Materials, Basics of Applying Textures, 6. UV Mapping Basics: i. Principles of UV Mapping, Importance in Texturing	**15**
	Unit-3: Lighting and Rendering 1. **Basics of 3D Lighting:** a. Types of Lights in 3D Graphics, Importance of Lighting in Scenes 2. **Understanding Shadows and Reflections**: a. Techniques for Achieving Realistic Shadows, Incorporating Reflections in Rendering 3. **Introduction to Rendering Engines:** a. Overview of Rendering Engines, Selecting Rendering Settings 4. **Introduction to Animation:** 5. **Keyframe Animation Basics:**	**15**

	a. Understanding Keyframes, Creating Basic Animations 6. **Working with the Timeline**: a. Navigating the Animation Timeline, Key Concepts in Animation Timing	
Pedagogy:	Lectures, Discussions, Assignments, group activities, Video/Animation to explain various concepts.	
References/ Readings:	Main Reading: 1. Kerlow, I. V. (2004). The Art of 3D: Computer Animation and Effects. Wiley. 2. Computer Graphics: Principles and Practice by John F. Hughes, Andries van Dam, Morgan McGuire, and David Sklar, 3rd Edition, Addison-Wesley. Additional Reading: 3. Blender Foundations: The Essential Guide to Learning Blender 2.7 by Roland Hess, Focal Press. 4. Blender 3D by Example-A Project-based Guide to Learning the Latest Blender 3D, EEVEE Rendering Engine, and Grease Pencil by Oscar Baechler, Xury Greer 2020, Packt Publishing.	
Course Outcomes:	On completion of the course, students will be able to: 1. Remember the basic concepts & terminologies used in 3D graphic design. 2. Understand the applications of 3D graphic design in various industries 3. Explain the basic concepts used in 3D graphic design 4. Demonstrate the concepts used in 3D graphic design	

Name of the Programme : B.Sc. Computer Science
Course Code : CSC-241
Title of the Course : Mobile App Development
Number of Credits : 3 (1 T+ 2 P)
Effective from AY : 2024-25 (Revised version of 2023-24)

Pre-requisites for the Course:	None	
Course Objectives:	1. Introduce mobile application development for the Android platform using XML and Java/Kotlin. 2. Understand the different components for building the Android App. 3. Develop applications that will run on Android phones and tablets.	
Content:		**No. of Hours**
	Introduction: Need for Mobile Apps, Overview of Android Architecture. **Introduction to Android Basic Building blocks:** Views, Activities, Intents, Services, Fragments, Broadcast Receivers and Content Providers, AndroidManifest.xml **UI Components**: Views, Layouts (Linear, Relative, Constraint) **Activities and UI**: User Interfaces and event handling, Activity LifeCycle. **Advanced UI**: Building Layouts with an Adapter (GridView&ListView), Custom Adapters, Menus (Options menu & Context menu) **Intents**: Intents uses, intent types (Implicit & Explicit) **Broadcast Receivers:** Broadcast receiver registration (Static & Dynamic), Broadcast Receiver Classes. **Services:** Overview of Services in Android, Service lifecycle. **Data Storage**: Introduction to data storage, Shared Preferences, Android File System, SQLite.	**15**
	Practical Work WEEK WISE	
	Week 1 & 2 a. Explore Android Studio IDE b. Set up a device or emulator for running the app c. Demonstrate the use of different layouts: Linear, Relative, Constraint d. Use of basic views (EditText, TextView, Button, RadioButton, Checkboxes) e. Event Handling (using listeners) (Calculator app, TicTacToegame)	**8**
	Week 3 & 4 a. ListView layout design, Load listView with Data, Load different views in ListView, ListView events, Add or Remove items to the listView b. Design GridView Layout, Load gridView with data c. Recycler view (Restaurant App)	**8**
	Week 5 & 6 a. Display a message using Custom Toast b. Intents (Implicit and Explicit)-launch an activity, passing data between activity, display a web page, dial a contact, send message etc.)	**8**

	Week 7, 8 a. Menus, Alert Dialog b. Broadcast Receivers (creating broadcast receiver, registering broadcast receiver, send and receive custom broadcast)	**8**
	Week 10 & 11 Web services - HTTP calls, JSON and XML, HTTP calls and JSON read	**8**
	Week 11, 12 & 13 SQLite database – CRUD operations on a SQLlite Database (Contact list app, notes app)	**12**
	Week 14 & 15 Firebase Realtime Database to build a mobile system	**8**
Pedagogy:	**Suggested strategies for use to accelerate the attainment of the various course outcomes.** 1. Course outcomes, evaluation scheme, prerequisite shall be discussed at the beginning. 2. The lecture method need not be only a traditional lecture method, but alternative effective teaching methods could be adopted to attain the outcomes. You may use 3. Video/Animation to explain various concepts. 4. Collaborative, Peer, Flipped Learning, etc. 5. Ask at least three HOT (Higher-Order Thinking) questions in the class, which promotes critical thinking. 6. Give an assignment based on one of the Course Outcomes. 7. Introduce Topics in manifold representations. 8. Show the different ways to solve the same problem and encourage the students to come up with their own creative ways to solve them. 9. Mini-project can be given to test the application of the concepts learnt.	
References/ Readings:	Main Reading: 1. Dawn Griffiths and David Griffiths, Head First Android Development: A Brain-Friendly Guide, Shroff/O'Reilly; Second edition, 2017. 2. John Horton, Android Programming for Beginners, Packt Publishing Limited, 2015. Additional Reading: 3. Erik Hellman, Android Programming: Pushing the Limits (MISL-WILEY), Wiley, 2013. 4. IyanuAdelekan, Kotlin Programming by Example, Packt Publishing Limited, 2018.	
Course Outcomes:	On completion of the course, students will be able to: 1. Describe the anatomy of a mobile app, and use Android components in designing simple mobile applications. 2. Identify the significance of each of the Android basic building blocks and determine when to use which component. 3. Discuss the data storage options available on android platform and perform basic CRUD operations on persistent data. 4. Design complete Android app by integrating the android building blocks and using firebase as backend tool.	

Name of the Programme : **B.Sc. Computer Science**
Course Code : **CSC-242**
Title of the Course : **Digital Marketing**
Number of Credits : **3 (1T+2P)**
Effective from AY : **2024-25 (Revised version of 2023-24)**

Pre-requisites for the Course:	None	
Course Objectives:	To acquaint the students with basic principles and concepts of digital marketing & advertising To understand and familiarize the students with the concept of Digital Marketing techniques like Adwords, search advertising, display advertising. To understand the concept of Search Engine Optimization (SEO)	
Content:		**No. of Hours**
	Unit-I Fundamentals of Digital Marketing in the digital world; Integrated marketing- The Phygital; Global trends in Digital Marketing; Digital channels- Paid, Owned and Earned; Careers in digital marketing; Skill development in Digital marketing Ad Words Fundamentals Understanding; Pay-per-click Advertisement; Significance and evolution of Ad Words in PPC, Bing Ads, Google Ads. AdWords Certification- Overview, Benefits and Preparation; Different Ad Formats; Keywords - significance and planning; Using Keyword Planner and other tools; Keyword matches and their usage; Campaign Structure and Organization Quality, Rank and Relevance of Ads; Reports and Analysis Metrics; Conversion Tracking; Campaign Optimization Search & Display Advertising with Adwords; Search with Adwords Keywords - planning, matching and combination; Specifications of an Ad and how to put it to good use; Managing Invalid Clicks; Ad extensions and usage; Dynamic search ads; Landing page - your virtual front; Campaign Experiment; Opportunities Tab; AdWords APIs; Ad Words editor-Benefits and usage; Managing multiple accounts Display with Adwords Google Display Network and Partnerships; Double click Ad Exchange and AdSense Campaign Creation and Structuring for display; Keyword and targeting through display network; Campaign Metrics, Analysis and optimization SEO Basics How search engines work; Different Search results and significance; Query types and significance; Components on SEO-onsite and off page; Keyword Planning; Using tools to get effective keywords; Long tail keywords- the hidden gems; Art and science of writing tags-URL, title,	**15**

	meta, H1, alt text, etc.; Write a good meta description; Page speed - its impact and improvement areas; All about links-broken, internal et al; Dealing with duplicate content; Robot.txt and sitemap; Structured data and schema.org. Using Search Console for SEO; KPIs of SEO; Tools for SEO; Moz SEO. Accelerated mobile pages and SEO; Artificial Intelligence, Voice search and SEO–what to look forward	
	Practical Work WEEK WISE	
	Week 1 & 2 **1) Real-world examples of successful digital marketing campaigns** **2) Setting Up Digital Marketing Tools** a. Introduction to key tools (Google Analytics, Google Ads, b. Social Media Management tools). Creating accounts and basic setups **3) Website and Content Marketing** a. Basic website setup using platforms like WordPress or Wix b. UX design principles in action c. Creating and publishing blog content d. Analyzing website traffic using Google Analytics	**08**
	Week 3 & 4 **4) Social Media Marketing** a. Developing a social media content calendar b. Creating engaging posts on different platforms c. Setting up and running paid social media campaigns, Plan, track, and analyze a paid Facebook advertising campaign d. Analyzing ad performance metrics	**08**
	Week 5 & 6 **5) Search Engine Optimization (SEO)** a. Conducting keyword research b. Optimizing website content for search engines c. Analyze page speed and mobile-friendliness of a web page d. Analyzing SEO performance using Google Analytics **6) Email Marketing Campaign** a. Creating and sending an email campaign b. Setting up an automated email sequence c. Analyzing email campaign data	**08**
	Week 7, 8 & 9 **7) Paid Advertising Campaign** a. Planning and executing a Google Ads campaign b. Budget management and optimization **8) Display Advertising and Remarketing** a. Designing and implementing display ads b. Setting up a remarketing campaign	**12**
	Week 10 & 11 **9)Storytelling on LinkedIn** Understand how LinkedIn's algorithm and search functions work a. Create a profile optimized to get you the results you want b. Develop a strategic plan for utilizing LinkedIn as a professional marketing tool	**08**

	c. Construct a compelling career story or narrative	
	Week 12 & 13 **10)Analytics and Reporting** a. Customizing and interpreting advanced Google Analytics reports b. Data-driven decision-making c. Students work on a data analysis project using real-world data sets	**08**
	Week 14 & 15 **11) AI + ChatGPT in Digital Marketing** a. Building chatbots for websites and social media b. AI tools for content generation and curation, Automating content creation processes c. Predictive analytics and customer behavior forecasting d. Designing a ChatGPT-powered Marketing Chatbot	**08**
Pedagogy:	PowerPoint presentations, Case studies	
References/ Readings:	Main Reading: 1. Dave Chaffey & Fiona Ellis-Chadwick, Digital Marketing: Strategy, Implementation and Practice, Pearson Education 2. Ekaterina Walter, The Power of Visual Storytelling, McGraw-Hill Education 3. Ben Hunt, Convert!: Designing Websites For Traffic and Conversions, John Wiley &Sons 4. Lon Safko, The Social Media Bible: Tactics, Tools, & Strategies for Business Success, Brilliance Audio; Unabridged Edition Additional Reading: 5. Pam Didner, Global Content Marketing, McGraw-Hill Education 6. Joe Pulizzi, Content Inc.: How Entrepreneurs Use Content to Build Massive Audiences and Create Radically Successful Businesses, McGraw-Hill Education 7. Mike Monteiro, You're My Favorite Client, A Book A part 8. Seth Godin, All Marketers Are Liars, Portfolio 8. JayBaer, Youtility:Why Smart Marketing Is About Help Not Hype, Portfolio 9. Russell Glass & Sean Callahan, The Big Data-Driven Business, Wiley 10. Damian Ryan and Calvin Jones, Understanding Digital Marketing: Marketing Strategies for Engaging the Digital Generation, Kogan Page 11. Ryan Deiss and Russ Henneberry, Digital Marketing for Dummies, John Wiley and Sons 12. Corey Rabazinski, Google Adwords for Beginners: A Do-It-Yourself 13. Guide to PPC Advertising, Create Space Independent Publishing Platform	
Course Outcomes:	At the end of the course learner will be able to 1. Understanding of digital landscape and building a case to leverage online channels 2. Apply advance concept of Search Engine Optimization to capture the right intent. 3. Strategize, implement and optimize online campaigns successfully 4. Develop and design Online Advertising campaigns, AdWords Campaign management and Campaign Basics across search.	

Name of the Programme : **B.Sc. Computer Science**
Course Code : **CSC-243**
Title of the Course : **Embedded Systems**
Number of Credits : **3 (1T+2P)**
Effective from AY : **2024-25 (Revised version of 2023-24)**

Pre-requisites for the Course:	Nil	
Course Objectives:	1. To introduce the basic concepts of Embedded Systems. 2. To Identify various types of sensors, actuators and their use. 3. To use various embedded software development tools. 4. To develop skills to build and demonstrate the real-life applications of embedded systems.	
Content:		**No of Hours**
	Introduction to Embedded Systems: Definition; History; General-Purpose Computing System v/s Embedded System; Classification of Embedded Systems; Application Areas of Embedded Systems; Characteristics of Embedded Systems; Microprocessors and Microcontrollers; **Components of Embedded system:** Microcontroller Unit (MCU); ASIC; FPGA; DSP processors; Various forms of System Memory; RISC v/s CISC; System-on-chip (SoC); Network-on-chip (NoC); Sensors, A/D-D/A Converters, Actuators; Embedded Systems Design Metrics; **Real-Time Operating System (RTOS)**: Definition and its types; Task and Task States; **Embedded Software Development Tools:** IDE, Code Editor, GUI, Compiler, Assembler, Builder, Disassembly, Linker, Simulator, Emulator.	**15**
	Practical Work Note: Programs to be executed on some of the Embedded boards like Arduino, Raspberry Pi, Bolt, etc.	
	Week 1 & 2 Introduction to any IDE to demonstrate various Embedded Software Development Tools. 1. Blinking a LED (Digital Output). 2. Blinking a LED using switch button (Digital Input-Digital Output).	**08**
	Week 3 & 4 3. Fading a LED (Analog Output). 4. Fading a LED using 10k Preset (Analog Input- Analog Output).	**08**
	Week 5 & 6 5. Traffic Light Simulation. 6. Interfacing LDR Sensor with LED.	**08**
	Week 7, 8 & 9 7. Interfacing Ultrasonic Sensor. 8. Interfacing DHT Sensor. 9. Interfacing PIR Motion Sensor.	**12**
	Week 10 & 11 10. Interfacing Piezo Buzzer with LED. 11. Generating Melodies (tones) using Piezo Buzzer. 12. Interfacing IR LED & Photo Diode.	**08**

	Week 12 & 13 13. Interfacing Seven-Segment Display (Cathode and Anode). 14. Interfacing Liquid Crystal Display (LCD).	**08**
	Week 14 & 15 15. Interfacing Servo Motor. 16. Interfacing DC Motor & Relay.	**08**
Pedagogy:	1. The lecture method need not be only a traditional lecture method, but alternative effective teaching methods could be adopted to attain the outcomes. You may use 2. Video/Animation to explain various concepts. 3. Collaborative, Peer, Flipped Learning, etc. 4. Adopt Problem Based Learning (PBL), which fosters students' Analytical skills, and develops design thinking skills such as the ability to design, evaluate, generalize, and analyze information rather than simply recall it. 5. Test their understanding through quizzes, presentations, case studies, mini-projects.	
References/ Readings:	Main Reading: 1. Raj Kamal (2020). Embedded Systems: SoC, IoT, AI and Real-time Systems. McGraw Hill Education (India) Private Limited. Additional Reading: 2. Lyla B. Das (2013). Embedded Systems: An Integrated Approach. Pearson.	
Course Outcomes:	On completion of the course, students will be able to: 1. Remember the basic concepts & terminologies of embedded systems 2. Identify various types of sensors, actuators and their uses. 3. Apply various embedded software development tools. 4. Build and demonstrate the real-life applications of embedded systems.	

Name of the Programme : B.Sc. Computer Science
Course Code : CSC-244
Title of the Course : 3D Modelling and Animation
Number of Credits : 3
Effective from AY : 2024-25 (Revised version of 2023-24)

Pre-requisites for the Course:	None	
Course Objectives:	1. To develop the skill and knowledge in 3D Modelling and Animation. 2. Students will understand the knowhow and can function either as an entrepreneur or can take up jobs in the multimedia and animation industry, video studios, edit set-up and other special effects sectors.	
Content		No. of hours
	Introduction to 3D Modeling and Animation: Overview of 3D Modeling and Animation, 3D industry and career opportunities, Role of 3D in various industries (film, gaming, virtual reality). Understanding 3D space and Cartesian coordinate system, Introduction to basic geometric shapes in 3D. **3D Modeling Techniques:** Working with vertices, edges, and faces. Basic modeling tools and operations (extrusion, beveling) Working with primitive shapes, Polygonal modeling techniques, Understanding edge loops and topology, mesh optimization, Subdivision surface modeling, Sculpting basics, Introduction to modifiers and their applications **UV Mapping and Texturing:** UV mapping fundamentals, Unwrapping and texturing basics, Understanding materials and shaders, Importing and applying textures, Procedural texturing basics. **Rigging Basics:** Introduction to rigging, Creating armatures and bones, Weight painting and rigging tools, Basic character rigging **Animation Fundamentals:** Keyframes and timeline basics, Animation curves and interpolation, Introduction to the graph editor. **Animation Techniques:** Character animation principles, Lip syncing and facial animation, Camera animation and storytelling through animation **Lighting, Rendering and Output**: Rendering settings and techniques, Principles of lighting in 3D, Output formats and resolutions, Post-production effects and compositing	15
	Practical Work (WEEK WISE)	
	Week 1 & 2 **Software Familiarization and Interface Navigation** - Introduction to the 3D animation and modeling software (open-source alternative - Blender) - Navigation of the user interface and basic tool functions - Understanding viewport controls and camera manipulation - Practice exercises for understanding the software	08

	environment	
	Week 3 & 4 **Basic 3D Modeling** - Creating simple 3D shapes and objects (e.g., a cup, chair) - Basic polygonal modeling techniques - Basic mesh editing tools and modifiers - Applying materials and textures to models - Simple modeling exercises - Model a simple scene or object	08
	Week 5 & 6 **Character Modeling, UV mapping and texturing** - Anatomy basics for character modeling - Character design principles - Guided exercises for character modeling - Model a more complex object or scene (e.g., a character's head) - UV mapping and texturing: Apply textures to a 3D model	08
	Week 7, 8 & 9 **Animation** - Application of animation principles (e.g., squash and stretch, anticipation) - Keyframe animation exercises - Timing and spacing - Walk cycles and basic character animation practice - Create a basic animation (e.g., bouncing ball, character movement) - Rigging basics for character animation - Rig and animate a simple character - Project: - Animate a character performing a simple action - Animate a character performing a complex action	12
	Week 10 & 11 **Advanced Animation Techniques** - Advanced keyframe and procedural animation exercises - Physics-based animation practice - Particle systems and dynamics projects - Group animation projects	08
	Week 12 & 13 **Lighting and Rendering** - Lighting techniques in 3D - Rendering projects with different materials and lighting setups - Post-production and compositing exercises - Render and present a final project	08
	Week 14 & 15 **Independent or Group Projects**	08
Pedagogy:	**Suggested strategies for use to accelerate the attainment of the various course outcomes.** 1. Lecture method 2. Demonstrations 3. Video/Animation to explain various concepts. 4. Assignments to promote self-learning	

	5. Collaborative, Peer, Flipped Learning 6. Group Projects 7. Quizzes 8. Presentations
References/ Readings:	Main Reading: 1. The Animator's Survival Kit Paperback – January 7, 2002 by Richard Williams 2. "Blender Foundations: The Essential Guide to Learning Blender 2.8" by Roland Hess Additional Reading: 3. 3D Art Essentials: The Fundamentals of 3D Modeling, Texturing, and Animation 1st Edition by Ami Chopine (Author) 4. 3D Animation Essentials 1st Edition by Andy Beane (Author) 5. "Digital Lighting and Rendering" by Jeremy Birn
Course Outcomes:	On completion of the course, students will be able to: 1. Remember essential concepts, techniques, and tools to create 3D models and animations for various applications, including games, movies, and simulations. 2. Demonstrate understanding of 3D modeling concepts and methods. 3. Apply modeling, texturing and animation techniques. 4. Compose 3D scenes with animated models.

Name of the Programme : B.Sc. Computer Science
Course Code : CSC-245
Title of the Course : Data Management Essentials
Number of Credits : 03 (1T+2P)
Effective from AY : 2024-25 (Revised version of 2023-24)

Pre-requisites for the Course:	Basics of Python	
Course Objectives:	1. To know the importance of data and its implications in various fields 2. To learn the fundamental data organization techniques and structures 3. To obtain basic skills in data analysis and interpretation 4. To recognize the ethical and legal considerations in data management	
Content:		**No of Hours**
	Unit-1: Introduction to Data Management: a. What is data management and why is it important? b. The impact of data in various fields c. Types of data: structured, unstructured, semi-structured *d. Relationship between data, information, and knowledge* **Data Life Cycle:** a. Stages of the data life cycle: Creation, collection, storage, processing, analysis, deletion b. Data quality and its importance c. Data governance and data ownership d. Importance of data security and privacy **Data Storage and Retrieval:** a. Different types of data storage: Databases, data warehouses, data lakes, cloud storage b. Choosing the right storage solution c. Data access and retrieval methods d. Understanding data formats (CSV, JSON, XML, etc.) **Data Organization and Modeling:** a. Organizing data for efficient storage and retrieval b. Data modelling: Entities, attributes, relationships **Tools and Technologies for Data Management:** a. Introduction to popular data management tools: Spreadsheets, database management systems, data analysis software b. Data visualization tools and techniques, Data visualization tools, cloud data management solutions	**15**
	Practical Work	
	Week 1 & 2 **Identify different data types:** Hands-on exercise classifying data into categories like numerical, categorical, text, etc. **Explore data sources:** Experiment with collecting data from various sources like spreadsheets, websites, APIs, sensors.	**08**
	Week 3 & 4	**08**

	Understand data quality issues: Practice identifying and correcting data errors, missing values, inconsistencies. **Learn basic data visualization:** Create simple charts and graphs to visualize data relationships and patterns.	
	Week 5 & 6 **Organize data in spreadsheets:** Implement different sorting, filtering, and formatting techniques. **Build a relational database model:** Understand and practice entity-relationship diagrams for data relationships.	**08**
	Week 7, 8 & 9 **Import data into a database:** Learn basic SQL commands to load data from external sources. **Query data using SQL:** Practice selecting, filtering, and aggregating data using fundamental SQL operations. Also optimizing and indexing.	**12**
	Week 10 & 11 **Create basic database reports:** Generate reports summarizing data from different tables in your database. **Perform basic data cleaning:** Implement techniques to address missing values, outliers, and inconsistencies.	**08**
	Week 12 & 13 **Implement data access controls:** Understand and practice user roles, permissions, and authentication methods for data security. **Encrypt sensitive data:** Learn basic data encryption techniques to protect confidential information.	**08**
	Week 14 & 15 **Develop a data governance plan:** Create a basic framework for data ownership, usage, and lifecycle management. **Case Study: Analyze movie ratings**: Leverage movie rating data to recommend movies or compare critics' and audience perspectives.	**08**
Pedagogy:	**Suggested strategies for use to accelerate the attainment of the various course outcomes.** 1. Flipped Classroom Approach. 2. Pose data-driven challenges and encourage collaborative problem-solving through group discussions and mini-projects. 3. Host resources, quizzes, and discussions on LMS, fostering a blended learning environment. 4. Peer-Tutoring. 5. Expose student to industry by inviting industry experts to deliver insights about their work culture and industry expectations.	
References/ Readings:	Main Reading: 1. Strengholt, P. (2020). *Data Management at Scale*. O'Reilly Media. Additional Reading: 2. International, D. (2017). *DAMA-DMBOK: Data Management Body of Knowledge (2nd Edition)*. Technics Publications, LLC. 3. Watson, R. (2022). *Data Management: Databases and Analytics*. Prospect Press.	
Course Outcomes:	On completion of the course, students will be able to: 1. Describe the importance of data and its implications in various fields 2. Explain the basic data organization techniques and structures	

	3. Apply the skills in data analysis and interpretation of similar use cases 4. Analyze the ethical and legal considerations in data management

Semester IV
Name of the Programme : Bachelor of Science in Computer Science (Honours) Course
Course Code : CSC-202
Title of the Course : Data Structures and
Algorithms
Number of Credits : 4 (3T + 1P)
Effective from AY : 2024-25 (Revised version of 2023-24)

Pre-requisites for the Course:	Basic programming	
Course Objectives:	1. To familiarize with basic data structures. 2. To develop the ability to choose the appropriate data structure for designing efficient algorithms.	
Content:		**No of Hours**
	Unit-1: Algorithm Basics – Algorithms and Data Structures, Pseudocode, Algorithm Features. Arrays – Basic Concepts, One-dimensional Arrays, Two-dimensional Arrays, Higher-dimensional Arrays. Linked Lists – Basic Concept, Singly Linked List, Doubly Linked List, Sorted Linked Lists, Linked List Algorithms, Linked List Selection-sort, Multithreaded Linked List, Linked List with loops.	**15**
	Unit-2: Stacks – Linked-List Stacks, Array Stacks, Double Stacks, Stack Algorithms-Applications of stacks. Queues – Linked-List Queues, Array Queues, Specialized Queues. Sorting Algorithms, Searching Algorithms, Recursion.	**15**
	Unit-3: Trees – Tree Terminology, Binary Search Tree, Tree Representations, Tree Traversal, Threaded Trees, Specialized Tree Algorithms, Balanced Trees. Graphs – Graph Terminology, Representation, Traversals, Finding Paths. Heaps and Hashing.	**15**
	Practical Work	
	Week 1 & 2 Multi-dimensional arrays. Singly linked list.	**04**
	Week 3 & 4 Doubly linked list, sorted list, multithreaded and circular linked list.	**04**
	Week 5 & 6 Stack using arrays and linked list and applications of stacks.	**04**
	Week 7, 8 & 9 Queues using arrays and linked list. Sorting Algorithms.	**06**
	Week 10 & 11 Searching algorithms, Recursion.	**04**
	Week 12 & 13 Trees with traversals.	**04**
	Week 14 & 15 Heaps and graphs with traversals.	**04**

Pedagogy:	**Suggested strategies for use to accelerate the attainment of the various course outcomes.** 1. The lecture method need not be only a traditional lecture method, but alternative effective teaching methods could be adopted to attain the outcomes. You may use 2. Video/Animation to explain various concepts. 3. Collaborative, Peer, Flipped Learning, etc. 4. Ask at least three HOT (Higher-Order Thinking) questions in the class, which promotes critical thinking. 5. Adopt Problem Based Learning (PBL), which fosters students' Analytical skills, and develops design thinking skills such as the ability to design, evaluate, generalize, and analyze information rather than simply recall it. 6. Introduce Topics in manifold representations. 7. Show the different ways to solve the same problem and encourage the students to come up with their own creative ways to solve them. 8. Discuss how every concept can be applied to the real world - and when that's possible, it helps improve the students' understanding 9. To promote self-learning, give at least one assignment (equivalent to 50% assignment weightage) where they can complete one MOOCs (certificate or equivalent) course out of lecture hour. Test their understanding through quizzes or presentations.
References/ Readings:	Main Reading: 1. Rod Stephens (2013). A Practical Approach to Computer Algorithms. Wiley. 2. Richard F. Gilberg, Behrouz A. Forouzan (2007). Data Structures: A pseudocode Approach with C. Cengage Learning. Additional Reading: 3. Clifford A. Shaffer (2009). A Practical Introduction to Data Structures and Algorithm Analysis. Prentice Hall. 4. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein (2022). Introduction to Algorithms. The MIT Press. 5. Jay Wengrow (2017). A Common-Sense Guide to Data Structures and Algorithms: Level Up Your Core Programming Skills. The Pragmatic Programmers.
Course Outcomes:	On completion of the course, students will be able to: 1. Understand basic data structures, their implementation and some of their standard applications. 2. Analyze space-time complexity of basic algorithms. 3. Design and analyze basic algorithms using appropriate data structures. 4. Code, debug and analyze programs using suitable data structures.

Name of the Programme : B.Sc. Computer Science
Course Code : CSC-203
Title of the Course : Object Oriented Technologies
Number of Credits : 4 (3T + 1P)
Effective from AY : 2024-25 (Revised version of 2023-24)

Pre-requisites for the Course:	Knowledge of Programming using any programming language	
Course Objectives:	1. This course introduces students to the principles and practices of object-oriented technology in software development. 2. Students will learn the fundamentals of object-oriented programming, design principles, and patterns. 3. Practical implementation using a programming language, such as Java or C++, is an integral part of the course.	
Content:		**No of Hours**
	Unit-1: Introduction to Object Oriented Programming and Basic OO concepts	**15**
	• Overview of programming paradigms • Evolution and principles of object-oriented programming • Comparison with procedural programming • Introduction to key OOP concepts: • Classes – attributes and methods; Encapsulation; class as a module and class as a type; uniform type system, static and non-static members • Objects – object creation; constructors and destructors; object references, object copying and cloning, object composition • Inheritance – types of inheritance, deferred features and classes, redeclaration, dynamic method dispatch • Polymorphism – overloading and overriding	
	Unit-2: Object Oriented Features	15
	• Memory management – reclaiming memory, automatic memory management, garbage collection methods • Interfaces • Access specifiers/modifiers • Exception Handling – basic concepts, exception handling mechanism, handling multiple exceptions, rethrowing, throws, user defined exceptions • Collection Framework – use of collection framework • Generics/Templates – horizontal and vertical type generalization, need for type parameterization, generic classes and methods	
	Unit-3: OOAD using UML, Design Patterns and Advanced Features of OOP	15
	• OOAD using UML • OOAD - Understanding the software development life cycle; Object-oriented analysis: identifying objects, classes, and relationships; Object-oriented design principles and patterns • Introduction to UML Diagrams (class diagram, use case diagram, sequence diagrams) • Design Patterns	

	• Introduction to design patterns and their significance; Common design patterns (Singleton, Factory, Observer, etc.); Implementing design patterns in code • Advanced features • Persistence and serialization • Concurrency/threads	
	Practical Work	
	Week 1 & 2 1. Use of command line environment and run-time environment in Java (javac and java) 2. Creating classes	04
	Week 3 & 4 1. Constructors and overloading 2. Object composition using references	04
	Week 5 & 6 1. Inheritance 2. Overriding, polymorphism and dynamic binding	04
	Week 7, 8 & 9 1. Abstract class, interfaces and multiple interface inheritance 2. Use of static keyword	06
	Week 10 & 11 1. Exception handling 2. Arrays	04
	Week 12 & 13 1. Collection framework – ArrayList, Maps 2. Minimum one exercise on each design pattern	04
	Week 14 & 15 1. Generics 2. Concurrency/Threads	04
Pedagogy:	**Suggested strategies for use to accelerate the attainment of the various course outcomes.** Tutorials, Collaborative/peer learning, Hands-on assignments	
References/ Readings:	Main Reading: 1. Timothy Budd, “An Introduction to Object Oriented Programming”, Pearson Edition, 3rd Edition. 2. Bjarne Stroustroup “The C++ programming Language” Addison Wesley. Additional Reading: 3. Khalid Mughal, “A Programmer’s Guide to Java Programming Certification, Pearson Education, 3rd Edition. 4. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides , “Design Patterns: Elements of Reusable Object-Oriented Software” Addison-Wesley Professional, 1st Edition.	
Course Outcomes:	On completion of the course, students will be able to: 1. Define and recall fundamental Object-Oriented (OO) concepts, including classes, objects, encapsulation, and inheritance. 2. Understand object-oriented principles. 3. Analyze given problem, breakdown into logical units and solve using bottom-up approach. 4. Develop simple Object-Oriented programs using a chosen programming language to implement basic concepts like classes, objects, inheritance and polymorphism in practical programming	

	scenarios

Name of the Programme : B.Sc. Computer Science
Course Code : CSC-204
Title of the Course : Operating System
Number of Credits : 4 (4T)
Effective from AY : 2024-25 (Revised version of 2023-24)

Pre-requisites for the Course:	The student should have basic concepts of computer architecture and programming.	
Course Objectives:	1. To understand the fundamental concepts of operating systems. 2. To understand process management and synchronization techniques. 3. To gain knowledge on the memory management techniques. 4. To analyze the various techniques in storage management and file management.	
Content:		**No. of Hours**
	Unit-1: Introduction to Operating Systems: Concept of Operating Systems, Computer System Organization, Computer System Architecture **Operating System Structures:** Services, User and Operating System Interface, System Calls and its types, System Programs, Operating System Structure (Simple, Layered, Microkernel, Modules, Hybrid), System Boot **Process Management:** Process (Concept, State, Process Control Block, Threads), Process Scheduling (Scheduling Queues, Schedulers, Context switching), Operation on Processes (Creation, Termination), Inter-process Communication (Shared Memory Systems, Message Passing Systems) **Threads:** Concept, Multicore Programming, Multithreading Models	**15**
	Unit-2: Process Synchronization: Concept, Critical-Section Problem, Peterson’s Solution, Synchronization Hardware, Mutex Locks, Semaphores, Classic Problems of Synchronization (Bounded-Buffer Problem, Readers-Writers Problem, Dining-Philosophers Problem) **CPU Process Scheduling:** Basic Concepts, Scheduling Criteria, Scheduling Algorithms (First-Come-First-Serve, Shortest-Job First, Priority Scheduling, Round Robin) **Deadlocks:** System model, Characterization, Methods for handling Deadlocks, Deadlock Prevention, Deadlock Avoidance (Safe State, Resource-Allocation-Graph Algorithm, Banker’s Algorithm), Deadlock Detection, Recovery from Deadlock	**15**
	Unit-3: Memory Management: Concept, Swapping, Contiguous Memory Allocation, Segmentation, Paging, Structure of the Page Table.	**15**

	Virtual Memory: Demand Paging, Copy-on-Write, Page Replacement (Basic, FIFO, Optimal, LRU), Allocation of Frames, Thrashing (Concept, Causes)	
	Unit-4: Storage Management: Overview of Mass-Storage Structure, Disk Structure, Disk Attachment, Disk Scheduling (FCFS, SSTF, SCAN, C-SCAN), RAID (Concept, Levels of RAID) **File System:** File Concept, Access Methods, Directory and Disk Structure, File Sharing, Protection. **File-System Implementation:** Structure, File-System Implementation, Directory Implementation, Allocation Methods	**15**
Pedagogy:	Blended Learning, Problem-based Learning, Simulations	
References/ Readings:	Main Reading: 1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne (2018). Operating Systems Concepts. Wiley India. Additional Reading: 2. William Stallings (2018). Operating Systems-Internals and Design Principles. Pearson India. 3. J. Archer Harris, John Cordani (2001). Schaum's Outline Operating Systems. Mc Graw Hill Education India.	
Course Outcomes:	On completion of the course, students will be able to: 1. Remember the concepts of operating systems, its structure and process management. 2. Understand process synchronization techniques to formulate solution for critical section problems and CPU scheduling algorithms. 3. Apply memory management schemes of operating system. 4. Analyze the storage management and file management techniques of operating systems.	

Name of the Programme : B.Sc. Computer Science
Course Code : CSC-205
Title of the Course : Operating System Lab
Number of Credits : 2 (2P)
Effective from AY : 2024-25 (Revised version of 2023-24)

Pre-requisites for the Course:	Basic concepts of programming, computer organization and operating system.	
Course Objectives:	1. To understand how to use commands in an operating system. 2. To code and use shell script programs. 3. To code and use process management using system calls.	
Content:	**Practical Work**	**No. of Hours**
	Week 1-3 **Working with Linux Operating System** ● Installation of Linux Operating System ● Linux commands a. Directory handling utilities: cd, mkdir, rmdir, mv, pwd b. File manipulation utilities: cat, cp, ls, mv, rm, chmod, chown, find, cat, more, head, tail, cmp, wc, touch, pr c. General purpose utilities: date, history, man, who, whoami, uptime, finger, cal, uname, tree, bc, tar, zip d. String manipulation utilites: grep, egrep, cut, paste, tr, sort, rev, awk, sed e. Process utilities: ps, pid, ppid, tty, time, kill, exit f. Network utilities: ping, ifconfig, netstat, hostname, traceroute, telnet, ssh, mount ● Vi Editor ● Advanced Filtering Commands: awk, sed	**12**
	Week 4-9 **Shell Programming** ● To check if a given file exists in the current directory. ● To calculate the sum of the first 10 natural numbers. ● To find and display the largest number among three given numbers. ● Accepts a filename as an argument and displays its content. ● To calculate the factorial of a given number. ● Takes two numbers as input from user and performs basic arithmetic operations (addition, subtraction, multiplication, division). ● To count the number of lines in a given text file. ● Renames all files in a directory with a specified file extension. ● To concatenate two files and save the result in a new file. ● Check if a user is logged in and displays a message accordingly. ● To find and display the smallest and largest elements in an array. ● Searches for a specific pattern in a given text file. ● To find and display all the hidden files in a directory. ● Sorts a list of numbers/text in ascending/descending order. ● Perform basic string manipulation operations (concatenation, substring, length). ● Counts the number of files and directories in the current directory. ● To find and display the process ID of a specific running process. ● To check if an existing file is a read-only or write-only. ● To automate the backup of a specified directory.	**24**

	● Monitors the free disk space and sends an alert if it falls below a certain threshold. ● Checks if a given user exists or not on the system. ● Generates a random password with certain criteria. ● Monitors changes in a specific directory and logs them to a file. ● Checks the availability of a remote server. ● Display the process ID of the process using the most memory. ● Displays information about the operating system. ● To automate the installation of a set of packages. ● To find and delete files older than a specified number of days in a directory. ● Generate a report of user login activity for the last 24 hours. ● Generate a report of the most frequently used commands by users. ● To find and replace a specific pattern in multiple files within a directory. ● Convert a given text file into HTML format. ● Archive and compress log files older than a certain date. ● Automate the cleanup of temporary files and directories on the system. ● Input a file containing two columns representing time in seconds (as X axis) and a corresponding value (as Y axis). Create a line graph for the data file.	
	Week 10-15 **Process Management** ● Create a file, allow the user to write data to the file, display the contents and close the file using system calls ● Copying a file into another file using system calls. ● Display the file details including owner, size, access permissions and file access time using system calls ● Creation of a child process and allow the parent to display “parent” and the child to display “child” on the screen ● Creation of a child process to perform a task and before terminating, the parent waits for the child to finish its task ● Creation of a Zombie Process ● Creation of an Orphan Process ● Simulate the commands using system calls: echo, ls, cp, mkdir, rm, cat, chmod, mv, grep	**24**
Pedagogy:	Lab Exercises, Problem-Based Learning	
References/ Readings:	Main Reading: 1. Sumitabha Das (2017). UNIX Concepts and Applications. Tata McGraw Hill, India. Additional Reading: 2. Kenneth Roson (2007). UNIX: The Complete Reference. McGraw Hill, India. 3. W. Richard Stevens, Steven A. Rago (2013). Advanced Programming in the UNIX Environment. Addison-Wesley.	
Course Outcomes:	**On completion of the course, students will be able to:** 1. Understand how to use commands in operating system. 2. Apply and code programs using shell programming. 3. Apply and code process management using system calls.	

Name of the Programme : B.Sc. Computer Science
Course Code : CSC-221
Title of the Course : Introduction to Python Programming
Number of Credits : 4 (3T+1P)
Effective from AY : 2024-25 (Revised version of 2023-24)

Pre-requisites for the Course:	None	
Course Objectives:	1. To introduce programming concepts using Python. 2. To understand and apply fundamental concepts of Python programming to solve basic computational problems. 3. To demonstrate proficiency in utilizing advanced Python features such as object-oriented programming, file handling, exception handling, and integrating Python libraries for real-world applications.	
Content:		**No. of Hours**
	Unit-1: Python interpreter, python shell, operators, variables and assignment statements, script mode, functions, modules, command line arguments, control structures- if conditional statements, loops, break, continue and pass statements. Data types- Boolean, numbers, coercing integers to floats and vice versa, numerical operations, lists, tuples, sets, dictionaries, strings, formatting strings, unicode, docString, , packages, scope, recursion.	**15**
	Unit-2: Object Oriented concepts- classes, objects, abstract data types, polymorphism, encapsulation, modifier, accessor methods, static method, adding methods dynamically, composition, inheritance, built-in functions for classes.	**15**
	Unit-3: File Handling, Exception Handling, Applications of Python - use of Python libraries such as Matplotlib, Pandas, using databases with python.	**15**
	Practical Work WEEK WISE	
	Week 1 & 2 a) Write a function that returns the sum of digits of a number, passed to it as an argument. b) Write a function that returns True or False depending on whether the given number of a palindrome. c) Take the radius of a circle as input from the user, passes it to another function that computes the area and the circumference of the circle and displays the values.	**04**
	Week 3 & 4 a) Perform following actions on a list : Print the even-valued elements, Print the odd-valued elements, Calculate and print the sum and average of the elements of array, Print the maximum and minimum element of array, Remove the duplicates from the array, Print the array in reverse order. b) Define a function which can generate and print a list where the values are square of numbers between 1 and	**04**

	20 (both included). Then the function needs to print all values except the first 5 elements in the list.	
	Week 5 & 6 a) Write a program that accepts sequence of lines as input and prints the lines after making all characters in the sentence capitalized. b) Write a program that accepts a sentence and calculate the number of letters and digits. c) Given an array of integers, find two numbers such that they add up to a specific target number. d) Write a program that takes a sentence as input from the user and computes the frequency of each letter. Use a variable of dictionary type to maintain the count.	**04**
	Week 7, 8 & 9 a) Write a recursive function that multiplies two positive numbers a and b and return the result. Multiplication is to be achieved as a+a+a (b times). b) Given a list of strings, return the count of the number of strings where the string length is 2 or more and the first and last characters of the string are the same.	**06**
	Week 10 & 11 a) Define a base class Vehicle , having attributes registration number, make, model and color. Also, define classes PassengerVehicle and CommercialVehicle that derive the class Vehicle. The PassengerVehicle class should have additional attribute for maximum passenger capacity. The CommercialVehicle class should have an additional attribute for maximum load capacity. Define __init__ method for all these classes. Also, get and set methods to retrieve and set the value of the data attributes. b) Define classes Car, Autorickshaw and Bus which derive from the PassengerVehicle class mentioned in the previous question. The Car and Bus should have attributes for storing information about the number of doors, not shared by Autorickshaw. The Bus should have Boolean attribute doubleDecker not shared by Car and Autorickshaw. Define __init__ method for all these classes. Also define get and set methods to determine and set the value of the day attributes.	**04**
	Week 12 & 13 a) Write a function that takes two file names, file1 and file2 as input. The function should read the contents of the file file1 line by line and should write them to another file file2 after adding a newline at the end of each line. b) Write a function that reads a file file1 and displays the number of words and the number of vowels in the file.	**04**
	Week 14 & 15	**04**

	a) Write a function that reads the contents of the file Peom.txt and counts the number of alphabets, blank spaces, lowercase letters and uppercase letters, the number of words starting with a vowel and the number of occurrences of word “beautiful” in the file. b) Write a function that takes two files of equal size as an input from the user. The first file contains weights of items and the second file contains corresponding prices. Create another file that should contain price per unit weight for each item.	
Pedagogy:	● Incorporate hands-on coding exercises and projects to reinforce practical application. ● Encourage students to work in teams to solve complex problems, fostering analytical skills. ● Assign projects involving Python libraries and database interactions to bridge theory and practical application. ● Conduct regular code reviews and debugging sessions to enhance analytical and problem-solving skills.	
References/ Readings:	Main Reading: 1. Taneja Sheetal, Kumar Naveen, “Python Programming - A modular approach”, Pearson 2. Guttag John V., “Introduction to Computation and Programming using Python”, MIT Press, 2nd Edition. Additional Reading: 3. Downey, A. (2015). Think Python: How to Think Like a Computer Scientist. O'Reilly Media. 4. VanderPlas, J. (2016). Python Data Science Handbook: Essential Tools for Working with Data. O'Reilly Media. 5. https://www.w3schools.com/python/	
Course Outcomes:	On completion of the course, students will be able to: 1. Recall and apply fundamental Python constructs in programming tasks. 2. Understand the foundational concepts of scientific computing, including the use of libraries for mathematical operations and data analysis. 3. Apply Python programming skills to solve scientific problems, utilizing libraries for specific scientific computations and analysis. 4. Critically analyze scientific problems, applying Higher-Order Thinking (HOT) questions and real-world applications to develop effective problem-solving skills.	

Name of the Programme : B.Sc. Computer Science
Course Code : CSC-222
Title of the Course : Visual Computing
Number of Credits : 4 (3T + 1P)
Effective from AY : 2024-25 (Revised version of 2023-24)

Pre-requisites for the Course:	Programming Skills	
Course Objectives:	1. Gain insight into the connections between Visual Computing and key disciplines such as Computer Graphics, Computer Vision, Image Processing, and Geometry. 2. Explore the differences between captured and synthesized images and learn 3D data capture methods using scanning technology along with reverse engineering approaches. 3. Master diverse data structures for object representation, including volume and sweep representation, CSG, curves, surfaces, and delve into mesh operations such as fairing, remeshing. 4. Develop a comprehensive understanding of image-related processes, covering sampling, filtering, edge detection, textures, warping, transformations and advanced pipeline concepts like motion capture algorithms.	
Content:		**No. of hours**
	Unit-1: Fundamentals and Foundations ● Overview of Visual Computing: Visual computing and its relationship to traditional discipline like Computer Graphics, Computer Vision, Image Processing and Computational Geometry ● Image synthesis: Understand the concepts - Captured image in 2D v/s synthesized image via modeling & rendering. Capturing of 3D data using scanning technology and reverse engineering the shape of the object	**15**
	Unit-2: Geometry and Representation ● Geometric modeling: Data structures for object representation, Volume representation, Sweep representation, Cell decomposition, CSG, Boundary representation, Euler Operators, Beizier& B-Spline curves and surfaces ● Meshes: Polygonal mesh representation, Basic mesh descriptions, Mesh topology, Triangulations and Tessellations, Winged edge data structures for meshes, Operations on meshes like Fairing, Smoothing, Remeshing etc. ● Coordinate pipeline: Transformation principles, Geometry pipeline, Graphics pipeline, Vision pipeline, Advanced Coordinate pipeline	**15**
	Unit-3: Image Analysis and Manipulation ● Image geometry: Sampling and Quantization, Region and edges, Image filtering, Edge detection, Contours, Textures, Warping and morphing of images, Interpolations, Colors, Half-toning and Dithering, Highdynamic range imaging, Image pyramids	**15**

	● Motion capture: Motion tracking, randomized incremental algorithms, computational geometry algorithms Delaunay triangulations, Voronoi diagrams, kD-trees, Clustering by kMeans	
	Practical Work WEEK WISE	
	Week 1 & 2 Explore 3D programming IDE (eg. Alice 3D) to understand basic graphic concepts. ● Compose scenes and make objects interact ● Animate objects using built-in features (e.g., movement, rotation)	**04**
	Week 3 & 4 Using 3D programming IDE, ● Modify object properties (size, color, texture) and apply materials. ● Experiment with camera movements and angles and set up multiple viewpoints for a scene ● Write simple scripts to control object behavior and create dynamic and interactive scenes.	**04**
	Week 5 & 6 Using OpenGL/WebGL, ● Write a program to create basic 2D/3D geometric shapes. ● Create a program that allows users to apply 2D transformations (translation, rotation, scaling) to shapes like polygons or images.	**04**
	Week 7 & 8 Using OpenGL/WebGL, ● Create 3D primitives like cubes, spheres, cylinders. ● Add light source and customize the appearance of objects with colors, textures, and shaders to create realistic-looking surfaces	**04**
	Week 9 & 10 Using OpenGL/WebGL, ● Simulate a bouncing ball that falls under gravity and collides with the edges of the screen. ● Animate a 3D cube rotating along each axis with different speeds and directions. ● Implement keyframe animation for a simple object like a walking stick figure. Define key poses and use interpolation to generate smooth transitions between them.	**04**
	Week 11 & 12 Using a 3D modeling tool (eg. Blender), create complex objects like cars, aircrafts, etc. ● Set up a 3D scene with various objects, lighting, and camera angles. ● Animate the objects and show object collision.	**04**
	Week 13, 14 & 15 ● Mini-Project	**06**

Pedagogy:	1. Interactive Lectures to keep students engaged. This could include asking questions or having short discussions. Encourage students to share their thoughts and experiences related to the lecture topics. 2. Interactive demonstrations to illustrate complex concepts. 3. Using visual aids to enhance understanding, especially for topics related to geometric modeling, image synthesis and motion capture. 4. Organizing group activities or projects that require students to collaborate on solving problems related to Visual Computing. 5. Encouraging peer-to-peer learning by having students discuss their approaches to problem-solving. 6. Assigning projects that allow students to apply the concepts learned in lectures to real-world scenarios. Provide constructive feedback on projects to help students improve their practical skills. 7. Implement continuous assessment methods, such as quizzes, short tests or online discussions, to gauge students' understanding.
References/ Readings:	Main Reading: 1. Frank Nielsen (2005). Visual Computing: Geometry, Graphics, and Vision. Charles River Media. 2. Richard Szeliski (2022). Computer Vision: Algorithms and Applications. Springer Nature. Additional Reading: 3. Rafael C. Gonzalez, Richard E. Woods (2019). Digital Image Processing. Pearson Education. 4. Kessenich, J., Sellers, G., &Shreiner, D. (2016). OpenGL Programming Guide, Addison-Wesley Professional, 9th Edition.
Course Outcomes:	On completion of the course, students will be able to: 1. Remember the basic concepts and definitions related to visual computing. 2. Understand the relationships between visual computing and traditional disciplines, differentiate between 2D captured and synthesized images, elucidate various data structures for object representation, describe principles of transformation and the coordinate pipeline and explain image geometry concepts in the context of visual computing. 3. Develop the skills to capture 3D data, implement winged edge data structures for meshes with operations, apply transformation principles in various pipelines and implement motion tracking techniques using randomized incremental algorithms. 4. Assess and evaluate the impact of geometric modeling techniques, mesh operations and coordinate pipeline principles on the quality of visual computing applications.

Name of the Programme : B.Sc. Computer Science
Course Code : CSC-223
Title of the Course : Multimedia Technologies
Number of Credits : 4 (3T + 1P)
Effective from AY : 2024-25 (Revised version of 2023-24)

Pre-requisites for the Course:	Basic Knowledge of Computers and Internet.	
Course Objectives:	1. To understand the concepts of Color Models and Color harmony. 2. To understand Raster and Vector Graphics formats & basic graphic editing. 3. To identify and understand Font types and the selection of fonts. 4. To understand the types of Audio formats, codecs, basic audio editing, filters. 5. To understand the types of Video formats, codecs, basic video editing, filters and transitions, Data compression.	
Content:		**No. of Hours**
	Unit-1: Multimedia: Introduction, Uses of Multimedia, Social & Ethical considerations, Digital Representation, Standards. Color Theory: Color Basics, Color Systems, Color Gamut, ICC profiles, Gamma Correction, RGB Model, CMYK model, HSL model, Color Wheel , additive and subtractive colors, Complementary Colors, After Images, Color Combinations, Color & Contrast, Color Psychology, Itten's Contrasts, Proportion & Intensity, Contrast & Dominance, Shades, Tones & Tints, Color Studies	**15**
	Unit-2: Difference between Raster and Vector Graphics, Raster graphics: resolution, image compression, file formats, manipulation, Geometrical transformations Vector graphics: fundamentals, file formats, shapes, transforms and filters. Text and Layout: character set, fonts, layout & Text in graphics	**15**
	Unit-3: Sound: Sampling, quantization, Audio Codec & file formats, processing sound, sound editing and effects, compression, MIDI Audio. Animation: Principles of Animation, Perception of vision, Human Color Perception, Types of Animation, Keyframe, Sprite, file formats, animated gifs, interpolating motion. Video: How Video Works and is Displayed, Aspect Ratio, Frame size, Frame Rate, Regions, Video Codec & File formats, Processing & Delivery of Video.	**15**
	Practical Work Practical can be done using GIMP, Inkscape, Scribus, Photoshop, Illustrator, Flash, Blender, Audacity, Lightworks.	
	Week 1 & 2 1. Image compositing: Remove background and combine images to create a work of art 2. Learn to create images for Print, Web and Video	**04**
	Week 3 & 4 3. Design a Logo for a company	**04**

	4. Design a Brochure for given Product and details. Learn about different file formats	
	Week 5 & 6 5. Design a poster with given information and learn about image compression	**04**
	Week 7, 8 & 9 6. Edit the sound file and Learn about Effects and Filters of sound. 7. Record your voice and learn about Audio Compression 8. Learn Audio mixing and streaming of audio content	**06**
	Week 10 & 11 9. Learn about Video editing. Prepare video with rough cut.	**04**
	Week 12 & 13 10. Prepare video content with title and special effects. 11. Record video content and learn about video compressions.	**04**
	Week 14 & 15 12. Prepare Video content for a social media platform such as vimeo / YouTube	**04**
Pedagogy:	**Suggested strategies for use to accelerate the attainment of the various course outcomes.** 1. Conventional Lecture method a) Video/Animation to explain various concepts. b) Collaborative, Peer, Flipped Learning, etc. 2. Case based learning 3. Experiential Design Thinking 4. Formative and summative assessments 5. Live experimental projects	
References/ Readings:	Main Reading: 1. Nigel Chapman, Jenny Chapman; Digital Multimedia; Wiley India Edition, 2nd Edition 2. Vaughan Tay, Multimedia: Making it Work, 8th edition, Tata McGraw-Hill 3. Ranjan Parekh, Principles of Multimedia McGraw Hill Education; 2nd edition Additional Reading: 4. Roger Parker; “One-Minute Designer”; Hungry Minds Inc,U.S.; 2nd edition 5. Adobe Creative Team, Adobe Photoshop Classroom in a Book, Adobe Press 6. Adobe Creative Team, Adobe Illustrator Classroom in a Book, Adobe Press 7. Adobe Creative Team, Adobe Flash Professional CS6 Classroom in a Book, Adobe Press, 1st Edition 8. Ze-Nian Li & Mark S Drew; Fundamentals of Multimedia; Pearson Education International Edition 9. Jeffcoate Judith, Multimedia in Practice, Technology and Applications, PHI. 10. Multimedia Technologies: Concepts, Methodologies, Tools, and Applications - Syed Mahbubur Rahman Minnesota State University, Mankato, US.	

Course Outcomes:	On completion of the course, students will be able to: 1. Remember the fundamentals and underlying theories of multimedia. 2. Understand the concepts of Computer Graphics, Text, Audio, Animation & Video. 3. Apply concepts to create images, films, visual & sound effects for the creative media. 4. Design and develop different types of 2D/3D animations.

Exit Course

Name of the Programme : B.Sc. Computer Science

Course Code : CSC-261

Title of the Course : Digital Media Marketing &Analytics

Number of Credits : 4 (2T + 2P)

Effective from AY : 2024-25

Pre-requisites for the Course:	Website Designing and Programming knowledge	
Course Objectives:	1. To understand the concepts and techniques of Search Engine Optimization and Social Media Marketing. 2. To learn Web & Social Media Analytics, Inbound Marketing and emerging trends. 3. To apply the understanding of Search Engine Optimization, Social Media Marketing, web analytics and inbound marketing. 4. To analyze case studies of successful digital marketing campaigns and apply it in real-world scenario.	
Content:		**No. of Hours**
	Unit-1: Search Engine Optimization ● Introduction to SEO - How do Search Engines work?, Organic Search vs. Paid Search Results, Keyword Research ● On-page optimization - On-page SEO Elements, Technical SEO, Mobile SEO, Schema Markup ● Off-page optimization - Link Building, Social SEO, Local SEO, Backlink Audits using SEMrush ● SEO Audit, Tools, Measurement - SEO Audit, Algorithm Updates, Measurement with Google Analytics, SEO Resources, Careers in SEO **Social Media Marketing** ● Introduction to Social Media Marketing ● Creating Content for Facebook & Social Media, Tools for Content Creation ● Facebook Marketing - Facebook for Business, Facebook Insight, Facebook Pages and Post Best Practices, Facebook Ads – Campaign Objectives, Facebook Ads – Targeting Audiences, Facebook Ads – Impactful Creatives, Facebook Avatar, Apps, Live, Hashtags, Optimization and Reporting, Facebook Ad Policies, Facebook Messenger, Facebook Shop, Building Brand Awareness, Driving In-store Footfall, Facebook Pixel, Driving Online Sales, Generating Leads ● LinkedIn Marketing - Importance of LinkedIn presence, LinkedIn Strategy, Content Strategy, LinkedIn analysis, Targeting, Ad Campaign ● Instagram Marketing, X (Twitter) & Snapchat Marketing ● Social Media Marketing Tools, Crafting a Successful Social Media Strategy	**15**
	Unit-2: Web and Social Media Analytics ● Introduction to web analytic - What's analysis?, Is analysis worth the effort?, Small businesses, Medium and Large scale businesses, Analysis vs intuition ● Google Analytics -Getting Started With Google Analytics, How Google Analytics works?, Accounts, profiles, and users	**15**

	navigating Google Analytics, Basic metrics, Main sections of Google Analytics reports, Traffic Sources Direct, referring, and search traffic Campaigns AdWords, Adsense. ● Content Performance Analysis- Pages and Landing Pages, Event Tracking and AdSense, Site Search. ● Visitor Analysis- Unique visitors, Geographic and language information, Technical reports, Benchmarking. ● Social Media Analytics- Facebook insights, Twitter analytics, YouTube analytics, Social Ad analytics /ROI measurement. ● Actionable Insights **Inbound Marketing** ● Attracting your potential customers into the conversion funnel ● Converting your prospects into leads using emails ● Landing Page ● Conversion Optimization, Conversion Optimization Patterns for Engaging website Visitors ● Lifecycle Emails **Emerging Trends - An Introduction** ● AI and machine learning in digital marketing, Voice search optimization, Chatbots and conversational marketing, Augmented Reality (AR) and Virtual Reality (VR) marketing	
	Unit-3: Practical Activities - To be carried out along in sync with the concepts mentioned in Unit I & II respectively. 1. To learn to optimize web content for better search engine visibility, Perform keyword research using tools like Google Keyword Planner or SEMrush and optimize a webpage accordingly. 2. To understand the importance of content planning and creation, develop a content calendar for a hypothetical business, create blog posts or articles, and schedule their publication. 3. To gain hands-on experience in managing social media accounts and creating engaging content, create social media profiles for a business on platforms like Facebook, Instagram, and LinkedIn, and develop a social media content calendar. 4. To gain practical experience in launching and optimizing PPC advertising campaigns, set up a Google Ads campaign targeting specific keywords relevant to a business, create ad copies, and monitor the campaign's performance. 5. To collect and interpret data to measure the effectiveness, set up Google Analytics for a website, track key metrics such as traffic sources, user behavior, and conversions, and generate a report analyzing the data. 6. To gain practical experience in strategic planning and decision-making, develop a comprehensive digital marketing strategy for a fictional business, including setting objectives, identifying target audiences, allocating budgets, and selecting appropriate digital marketing channels. 7. To explore innovative ways to incorporate emerging trends, experiment with emerging technologies like AI-powered	**35**

	chatbots or virtual reality experiences and evaluate their potential applications in digital marketing.	
	Unit-4: Case Studies Analyze case studies of successful digital marketing campaign, like 1. ICICI Bank: Building India's Most Social Bank on facebook 2. Barclays Business Banking SEO Campaign **Mini - Project** Develop a mini-project applying the insights gained from the case studies to a real-world scenario. **Optional** -*Prepare for industry-recognized certifications by taking practice exams, completing online courses, and participating in certification programs offered by platforms like Google, Facebook, or HubSpot. It will enhance the credentials and increase the employability in the digital marketing field.*	**25**
Pedagogy:	Suggested strategies for use to accelerate the attainment of the various course outcomes. 1. A plan is to be developed by the student/s in consultation with the teacher incharge and to be approved. 2. One or methods mentioned below may be used for learning purposes. a. Intensive training / teaching b. Online or offline training (approved by the college or instructor) c. Approved MOOCS Courses d. Workshops - on-campus or off-campus e. Self-learning means & methods f. Enquiry-based learning 3. A work diary to be maintained where all the learning & work carried out to maintained and certified by the teacher incharges. 4. All deliverable & artifacts to be submitted in the college for evaluation and assessments.	
References/ Readings:	**Main Reading:** 1. Alhlou, F., Asif, S., & Fettman, E. (2016). Google Analytics Breakthrough: From Zero to Business Impact.(1st ed.). [Kindle Edition]. Wiley. 2. Deiss, R., & Henneberry, R. (2020). *Digital Marketing for Dummies.* [Paperback]. Wiley. 3. Enge, E., Spencer, S., &Stricchiola, J. (2023). *The Art of SEO.*(4th ed.). O'Reilly Media. 4. Gupta, Seema. (2022). *Digital Marketing*(3rd ed.). [Paperback]. McGraw Hill. 5. Rai, A. K. (2014). Social Media Marketing: Theories and Applications. Pearson Education India. **Additional Reading:** 1. Chaffey, D., Ellis-Chadwick, F., Johnston, K., & Smith, P. R. (2019). Digital Marketing: Strategy, Implementation, and Practice. Pearson. 2. Dover, D., & Agrawal, A. (2016). Search Engine Optimization (SEO) Secrets. Wiley. 3. Kumar, V. (2018). Analytics in Digital Marketing. Wiley. 4. Ratan, A. (2019). Digital Marketing: Concepts and Strategies. Oxford University Press.	

Course Outcomes:	On completion of the course, student will be able to 1. Understand the concepts and techniques of Search Engine Optimization, Social Media Marketing, Web & Social Media Analytics, Inbound Marketing. 2. Apply Search Engine Optimization, Social Media Marketing, web analytic and inbound marketing strategies. 3. Analyze the performances of digital marketing campaigns. 4. Create and run a small digital marketing campaign successfully.

Semester V
Name of the Programme : B.Sc. Computer Science
Course Code : CSC-300
Title of the Course : Database Management System
Number of Credits : 4T
Effective from AY : 2024-25 (Revised version of 2023-24)

Pre-requisites for the Course:	Knowledge of programming language and Operating System	
Course Objectives:	1. Provide a strong foundation in database concepts, technology, and practice. 2. Practice SQL programming through a variety of database problems. 3. Understand the use of concurrency and transactions in database	
Content:		**No. of Hours**
	Unit-1: Basic Concepts and Data Modelling • Introduction to Data Base Systems: File Systems versus a DBMS, The Relational Model, Levels of abstraction in a DBMS, Data independence, Schemas and Instances, Queries in DBMS, Concurrent Access and Crash Recovery, Structure of DBMS, Advantage of DBMS, Database Users. • Conceptual design and Entity Relationship model: Overview of Data Base Design, The ER model-features, Key Constraints, Participation Constraints, weak Entities, Class Hierarchies, Aggregation, Entity versus attribute, Entity versus relationship, Binary versus ternary relationship, aggregation versus ternary relationships.	**15**
	Unit-2: Relational Database Design • The Relational Model: Attributes and domains, Relations, Integrity Constraints, Key Constraints, Foreign Key Constraints, General Constraints, Enforcing Integrity constraints. • Logical Database design ER to relational : Entity sets to tables, Relationship sets (without constraints) to tables, translating relationship sets with key constraints, translating relationship sets with participation constraints, translating weak entity sets, translating class hierarchies, translating ER diagrams with aggregation. • Schema Refinement and Normal forms: Introduction, Why Schema Refinement? Functional Dependencies, Normal Forms: BCNF, Third Normal Form, Normalisation-Decomposition up to BCNF	**15**
	Unit-3: Relational Algebra and SQL • Relational Algebra: Relational algebra operations- select, project, division, cartesian product, rename, join, natural join, equijoin and their implementation. • SQL: DDL, DML, The Form of Basic SQL query, Condition specification, SQL Joins, Outer joins, Union, Intersect, Except, Nested queries, Grouping, Aggregate Operators, Views.	**15**

	• SQL: Embedded SQL, Cursors, Dynamic SQL, Triggers and Stored Procedures	
	Unit-4: Transaction Management and Crash Recovery • Transaction management : The concept and properties of transaction, transaction and schedule, Notion of consistency, Serializability, Isolation levels, Lock based concurrency control, concurrency control without locking, deadlocks • Crash Recovery: Introduction to crash recovery, Recovery and atomicity, Log based recovery, Shadow paging.	**15**
Pedagogy:	**Suggested strategies for use to accelerate the attainment of the various course outcomes.** Lectures, Tutorials, Collaborative/peer learning, Hands-on assignments	
References/ Readings:	Main Reading: 1. Henry F. Korth, Abraham Silberchartz, S. Sudarshan(2021), "*Databse System Concepts*", 7th Edition, McGraw Hill Publication. Additional Reading: 1. R. Ramkrishnan, Johannes Gehrke(2002), "*Database Management Systems*", 3rd Edition, McGraw Hill Publication. 2. RamezElmasri, Shamkant B. Navathe(2017), "*Fundamentals of Database Systems*", 7th Edition, Pearson Education. 3. Bipin Desai(2010), "*An Introduction to Database systems*", Revised Edition, Galgotia Publications.	
Course Outcomes:	On completion of the course, students will be able to: 1. Remember the basic concepts of relational database management systems, entity-relationship model, relational database design, relational algebra, SQL, Concurrency Control, Embedded SQL and Crash Recovery. 2. Understand the concept of keys (primary, foreign, candidate keys) and their role in maintaining data integrity. 3. Apply normalization techniques to design normalized relational database schemas, Write SQL queries for data retrieval, insertion, updating, and deletion. 4. Analyze given specifications, design ER Diagrams and convert it tables.	

Name of the Programme : **B.Sc. Computer Science**
Course Code : **CSC-301**
Title of the Course : **Computer Networks**
Number of Credits : **4(3T +1P)**
Effective from AY : **2024-25 (Revised version of 2023-24)**

Pre-requisites for the Course:	Introduction to Programming using C	
Course Objectives:	1. To provide a strong background of Network Concepts. 2. To be familiar with the components required to build and design different types of networks. 3. To explain the various protocols associated with the network layers.	
Content:		**No. of Hours**
	Unit-1: Data Communication and Physical Layer **Introduction:** ● Networking and data communication, ● Applications of Networks **Networks:** ● Components and Categories, Types of Connections, Topologies, Transmission Modes **Protocols and Standards:** ● Layered Architecture, OSI model, Functions of each layer, TCP/IP model; **Data Encoding:** ● Manchester, Differential Manchester **Transmission Media:** ● Twisted pair, Coaxial Cable, Fiber Optics, Wireless Media	**15**
	Unit-2: DataLink Layer **Data Framing techniques:** ● Character Count, Character Stuffing, Bit Stuffing **Error detection and correction:** ● Parity, Hamming code **Elementary Data Link Protocols:** ● Stop and wait, Sliding window protocols - Go back-N: ARQ, Selective repeat ARQ **MAC Sublayer,** ● Random Access Protocols: ALOHA, CSMA, CSMA/CD, CSMA/CA **Network Standards:** ● IEEE 802.3 (Ethernet) frame format ● IEEE 802.11 Architecture	**15**
	Unit-3: Network Layer, Transport Layer and Application Layer **Network Service types:** ● Virtual Circuits, Datagrams **Routing Algorithms:-** ● Shortest path routing, Flooding, Distance Vector routing, **Congestion:** ● causes and prevention	**15**

	Internet Protocols: ● IP Frame Format, IP Addressing, Subnets, **Transport Services:** ● Connectionless, Connection-oriented, Transport service primitives Berkley sockets **Transport layer Protocols:** ● User Datagram Protocol, Transmission Control Protocol; **Connecting Devices of different layers:** ● Hub, Repeater, Bridges, Switches, Routers. Gateways **Application Layer** ● Electronic Mail; Domain Name System	
	Practical Work Note : Languages/Tools: C/C++, NS-2 should be used for practical	
	Week 1 & 2 Create scenario and study the performance of network with star topology through NS-2 simulation.	**4**
	Week 3 & 4 Implementation of framing using Bit stuffing/destuffing	**4**
	Week 5 & 6 Implementation of framing using char stuffing/destuffing	**4**
	Week 7 Implementation of parity generator and checker	**2**
	Week 8 Study of Stop and Wait Protocol using NS-2.	**2**
	Week 9 & 10 Implementation of Dijkstra's algorithm for Shortest Path Routing.	**4**
	Week 12 &13 Implementation of IP fragmentation and reassembly.	**4**
	Week 14 Using diagnostic Network Commands: ping, traceroute, netstat, nslookup.	**2**
	Week 15 Study of Congestion Control Algorithms using NS2 simulation	**2**
Pedagogy:	**Suggested strategies for use to accelerate the attainment of the various course outcomes.** 1. The lecture method need not be only a traditional lecture method, but alternative effective teaching methods could be adopted to attain the outcomes. You may use 2. Video/Animation to explain various concepts. 3. Collaborative, Peer, Flipped Learning, etc. 4. Adopt Problem Based Learning (PBL), which fosters students' Analytical skills, and develops design thinking skills such as the ability to design and analyze the information. 5. To promote self-learning, give at least one assignment. 6. Test their understanding through quizzes or presentations.	

References/ Readings:	Main Reading: 1. Behrouz A. Forouzan(2017), "*Data Communications and Networking*" , Fifth Edition, McGraw Hill Education; 2. Andrew S. Tanenbaum(2010), "*Computer Networks*", 5th Edition, Pearson Education India;
Course Outcomes:	Upon completion of the course, students should be able to: 1. Remember the concepts, terminologies, network models and networks based on type and topology. 2. Understand different protocols for data transmission at the DLL. 3. Apply the techniques of setup networks and also implement subnetting. 4. Analyze the protocols used, transport and application layer protocols

Name of the Programme : B.Sc. Computer Science
Course Code : CSC-302
Title of the Course : Software Engineering
Number of Credits : 4 (3T + 1P)
Effective from AY : 2024-25(Revised version of 2023-24)

Pre-requisites for the Course:	Nil	
Course Objectives:	This course will enable the student to learn 1. Recall Agile values, principles, scrum techniques 2. Describe software development, Agile methods 3. Explain Evolution, fundamentals of software engineering, Agile development 4. Compare the different methods used in software development 5. Discuss source control tools for collaborative work, testing methods 6. Justify the need of software development, testing of software, continuous integration	
Content:		**No. of Hours**
	Unit-1: Introduction to Software Engineering: Software Development phases (Requirement, Analysis, Design and Implementation, testing and Maintenance), SDLC, waterfall methodology, Prototyping and Iterative, Reverse engineering, Reengineering Introduction to Source Control tool: versioning, Check-in/checkout. commit, branching, merging, synchronization	**15**
	Unit-2: Agile Approach: Agile Framework, Agile Manifesto, Agile principles, Extreme Programming, Scrum Software Project Management using Scrum: User stories, Estimation using story points, sprint, blacklog (product and sprint), Scrum team, scrum artifacts, scrum ceremonies	**15**
	Unit-3: Design and implementation: using XP, TDD, refactoring (Code smells and refactoring techniques), unit testing and Pair programming Quality assurance: verification and validation, Testing approaches, Types of testing, testing tools, Junit, selenium, build tools, Iteration and release planning, Introduction to continuous integration	**15**
	Practical 1. Git 2. User stories and estimation 3) Burndown Charts, Scrum board and Trello 4) Junit, Selenium 5) Refactoring Exercise (pair programming) 6)Debugging and defect tracking using Bugzilla 7) Maven Build 8) Javadoc	**30**

	The above practicals should be done using a mini project. Using SCRUM, refracturing using pair programming and tools like Eclipse, Git, Selenium, Bugzilla, Trello
Pedagogy:	Lecture method using ICT tools
References/ Readings:	Main Reading: 1. Pankaj Jalote(2005), “*Integrated approach to Software Engineering*”, 3rd Edition,Narosa Publishing House 2. Chris Sims and Hillary Louise Johnson(2011), “*Elements of Scrum*”, Dymaxicon, LLC 3. Martin Fowler(2018), “*Refactoring: Improving the Design of Existing Code*”2nd Edition, Addison, Wessley Additional Reading: 1. Ken Schwaber and Mike Beedle(2002), “*Agile Software Development with Scrum”*, 1st Edition, Pearson Education 2. S Kenneth Rubin(2015), *“Essential SCRUM: A practical guide to the most popular Agile Process*”, Pearson Education 3. Kent Beck, Cynthis Andres(2004), “*Extreme Programming Explained: Embrace Change*”, 2nd Edition,Addison,Wessley Second Edition
Course Outcomes:	At the end of the course, learner will be able to: 1. Remember the concepts of software Engineering 2. Understand Agile values, principles, scrum techniques 3. Apply software engineering, Agile methods in software development 4. Analyze software Evolution, Agile Approach, SCRUM, Design and implementation and Quality testing

Name of the Programme : B.Sc. Computer Science
Course Code : CSC-303
Title of the Course : DBMS Lab
Number of Credits : 2(2P)
Effective from AY : 2024-25 (Revised version of 2023-24)

Pre-requisites for the Course:	The student should have basic knowledge of Information Technology and Programming concepts.	
Course Objectives:	1. Designing and conceptualizing a relational data model. 2. Implementing the relational database concepts through some DBMS package 3. Managing users and access control to data. 4. Designing and creating mini -Client-Server Application	
Content:	**Practical Work WEEK WISE**	**Practical**
	Week 1 & 2 ● Installation of DBMS Software's ● Gathering information, analyzing data, Creating ER Diagram, Reduction to Tables. ● Creation/modification of database tables using DDL statements and GUI tools of the DBMS software. ● **Writing SQL Queries using** - Simple Select Query - Select with where clause	**08**
	Week 3 & 4 ● Writing SQL Queries using SQL Joins ● **Writing SQL Queries using** - Aggregate Functions - Group function and having clause - Operators - Functions	**08**
	Week 5 & 6 ● **Writing SQL Queries using** - Set operations - Sorting data ● **Writing SQL Sub query** - Returning single row - Returning multiple rows	**08**
	Week 7, 8 & 9 ● **Writing SQL Sub query** - Returning more than one column - Correlated sub query ● Writing SQL Queries using Stored Procedures ● Writing SQL Queries using Triggers	**12**
	Week 10 & 11 ● Writing SQL Queries for Views and User management, granting/revoking privileges, roles. ● Writing SQL Queries for Database User Level Security for Database tables and Views.	**08**
	Week 12 & 13 ● Develop a desktop or web application using any front-end tools. Establish a connection to the database and incorporate functionality to insert records into the tables.	**08**

	Week 14 & 15 ● Develop a desktop or web application using any front-end tools. Establish a connection to the database and integrate functionality for updating, deleting, and querying records from the tables.	**08**
Pedagogy:	**Suggested strategies for use to accelerate the attainment of the various course outcomes.** 1. The lecture method need not be only a traditional lecture method, but alternative effective teaching methods could be adopted to attain the outcomes. You may useVideo/Animation to explain various concepts.Collaborative, Peer, Flipped Learning, etc. 2. Ask at least three HOT (Higher-Order Thinking) questions in the class, which promotes critical thinking. 3. Adopt Problem Based Learning (PBL), which fosters students' Analytical skills, and develops design thinking skills such as the ability to design, evaluate, generalize, and analyze information rather than simply recall it. 4. Introduce Topics in manifold representations. 5. Discuss how every concept can be applied to the real world - and when that's possible, it helps improve the students' understanding 6. To promote self-learning, give at least one assignment (equivalent to 50% assignment weightage) where they can complete one MOOCs (certificate or equivalent) course out of lecture hour. Test their understanding through quizzes or presentations.	
References/ Readings:	Main Reading: 1. Henry F. Korth, Abraham Silberchartz, S. Sudarshan(2021), "*Databse System Concepts*", 7th Edition, McGraw Hill Publication. Additional Reading: 1. R. Ramkrishnan, Johannes Gehrke(2002), "*Database Management Systems*", 3rd Edition, McGraw Hill Publication. 2. Ramez Elmasri, Shamkant B. Navathe(2017), "*Fundamentals of Database Systems*", 7th Edition, Pearson Education. 3. Bipin Desai(2010), "*An Introduction to Database systems*", Revised Edition, Galgotia Publications.	
Course Outcomes:	On completion of the course, students will be able to: 1. Implement and use a relational databDesign and implement relational database concepts using data 2. definition language for a given problem-domain. 3. Design, implement and manipulate the database schema using SQL queries for a given problem-domain. 4. Develop application using PL/SQL & Front end	

Name of the Programme : B.Sc. Computer Science
Course Code : CSC-321
Title of the Course : Python for Data Science
Number of Credits : 4(3T+1P)
Effective from AY : 2024-25 (Revised version of 2023-24)

Pre-requisites for the Course:	Basic programming	
Course Objectives:	1. To familiarize with data science concepts. 2. To develop the ability to apply data science concepts to problems.	
Content:		**No. of Hours**
	Unit-1: Introduction to Data Science Toolboxes for Data Scientist-Fundamental Python Libraries for Data Scientist, Data Science Ecosystem Installation, Integrated Development Environments, Python for Data Science Descriptive statistics – Introduction, Data Preparation, Exploratory Data Analysis, Estimation. Statistical Inference- Introduction, Statistical Inference: The Frequentist Approach, Measuring the Variability in Estimates, Hypothesis Testing	**15**
	Unit-2: Data Visualization - Controlling the line properties of a chart, creating multiple plots, playing with text, styling your plots, Box plots, Heatmaps, Scatter plots with histograms and such. Machine Learning - Different types of machine learning, Decision trees, Linear regression, Logistic regression, the naïve Bayes classifier, k-means clustering, Hierarchical clustering.	**15**
	Unit-3: Performing Predictions with Linear Regression – Simple linear regression, multiple regression, training and testing a model. Estimating the Likelihood of Events – Logistic regression. Case studies of regression, supervised and unsupervised learning.	**15**
	Practical Work	
	Week 1 & 2 Python Basics-Lists, Tuples, Dictionary, Sets, Numpy, Matrix, Descriptive Statistics	**04**
	Week 3 & 4 Pandas Dataframes, Data Visualization using appropriate data sets and python libraries	**04**
	Week 5 & 6 Linear regression case-study	**04**
	Week 7, 8 & 9 Logistic regression case-study	**06**
	Week 10 & 11: Tree Based Models	**04**
	Week 12 & 13: Supervised Machine Learning Case-study	**04**
	Week 14 & 15: Unsupervised Machine Learning Case-study	**04**

Pedagogy:	**Suggested strategies for use to accelerate the attainment of the various course outcomes.** 1. The lecture method need not be only a traditional lecture method, but alternative effective teaching methods could be adopted to attain the outcomes. You may use 2. Video/Animation to explain various concepts. 3. Collaborative, Peer, Flipped Learning, etc. 4. Ask at least three HOT (Higher-Order Thinking) questions in the class, which promotes critical thinking. 5. Adopt Problem Based Learning (PBL), which fosters students' Analytical skills, and develops design thinking skills such as the ability to design, evaluate, generalize, and analyze information rather than simply recall it. 6. Show the different ways to solve the same problem and encourage the students to come up with their own creative ways to solve them. 7. Discuss how every concept can be applied to the real world - and when that's possible, it helps improve the students' understanding 8. To promote self-learning, give at least one assignment (equivalent to 50% assignment weightage) where they can complete one MOOCs (certificate or equivalent) course out of lecture hour. Test their understanding through quizzes or presentations.
References/ Readings:	Main Reading: 1. Laura Igual, Santi Seguí (2017),"*Introduction to Data Science - A Python Approach to Concepts, Techniques and Applications*", 1st Edition,Springer. 2. Samir Madhavan (2015),"*Mastering Python for Data Science*", Packt Publishing. Additional Reading: 1. Sinan Ozdemir (2016), "*Principles of Data Science: Mathematical techniques and theory to succeed in data-driven industries*",Packt Publishing. 2. Frank Kane (2017),"*Hands-On Data Science and Python Machine Learning*",Packt Publishing. 3. Alberto Boschetti, Luca Massaron (2017). "*Python Data Science Essentials*",3rd Edition, Packt Publishing.
Course Outcomes:	On completion of the course, students will be able to: 1. CO1. Remember fundamental Python libraries for data science. 2. CO2. Understand quantitative modeling and data analysis techniques 3. CO3. Use data visualization techniques. 4. CO4. Apply supervised and unsupervised machine learning techniques to solve problem.

Name of the Programme	**: B.Sc. Computer Science**
Course Code	**: CSC-322**
Title of the Course	**: Image Processing**
Number of Credits	**: 4 (3T + 1P)**
Effective from AY	**: 2024-25 (Revised version of 2023-24)**

Pre-requisites for the Course:	Basic Knowledge of Mathematics, Linear Algebra and Programming (preferably in a language like Python)	
Course Objectives:	1. To introduce students to the Basic concepts and analytical methods of analysis of digital images. 2. To Study fundamental concepts of Digital Image Processing and basic relations among pixels. 3. To Study Different Spatial and Frequency domain concepts 4. To understand Restoration process of degraded image and Multi resolution processing. 5. To understand image compression and Segmentation Techniques.	
Content:		**No. of Hours**
	Unit-1: Digital Image fundamentals: Digital Image Representation, Fundamental steps in image processing, Concept of grey levels. Grey level to binary image conversion, Sampling and quantization, Resolution, Relationship between pixels. Image Enhancement in Spatial Domain: Point processing, Histogram processing, Image smoothing & Image sharpening. Image Enhancement in frequency Domain: Steps involved in frequency domain filtering, Fourier Transform, Image smoothing & Image sharpening.	**15**
	Unit-2: Image compression: Redundancies and their removal methods, Fedility criteria, Image compression models, lossy and lossless compression.	15
	Unit-3: Image segmentation: Detection of discontinuities, edge linking and boundary detection, thresholding, region – oriented segmentation Colour image processing: Colour fundamentals, Colour models, Pseudo colour image processing, full colour image processing	15
	Practical Work WEEK WISE	**Practicals**
	Week 1 & 2 1. Simulation and Display of an Image, Negative of an Image(Binary & Gray Scale) 2. Implementation of Relationships between Pixels	04
	Week 3 & 4 3. Implementation of Transformations of an Image 4. Contrast stretching of a low contrast image, Histogram, and Histogram Equalization	04
	5. Display of bit planes of an Image 6. Display of FFT(1-D & 2-D) of an image	04
	Week 7, 8 & 9 7. Computation of Mean, Standard Deviation, Correlation coefficient of the given Image	06

	8. Implementation of Image Smoothening Filters(Mean and Median filtering of an Image)	
	Week 10 & 11 9. Implementation of image sharpening filters and Edge Detection using Gradient Filters 10. Image Compression by DCT,DPCM, HUFFMAN coding	04
	Week 12 & 13 11. Implementation of image restoring techniques	04
	Week 14 & 15 12. Implementation of Image Intensity slicing technique for image enhancement 13. Canny edge detection Algorithm	04
Pedagogy:	Use of octave and scilab for Image Processing Concepts Use of gimp and inkscape for relationships among pixel	
References/ Readings:	Main Reading**:** 1. – R.C. Gonzalez & R.E. Woods (2002), “*Digital Image processing*”, 3rd Edition, Addison Wesley/ Pearson education 2. - S Jayaraman, S Esakkirajan and T. Veerakumar (2010), “*Digital Image processing*”, 3rd Edition, Tata McGraw Hill Additional Reading**:** 1. William K. Pratt(2004), “*Digital Image Processing*”, 3rd Edition, Wiley 2. Alan c. Bovik(2009), “*The Essential Guide to Image Processing*”, Academic Press 3. Anil K.Jain(1995), “*Fundamentals of Digital Image processing*”, PHI	
Course Outcomes:	On completion of the course, students will be able to 1. Remember the fundamentals and advances in Machine vision. 2. Understand the concepts of image analysis 3. Apply the mathematical knowledge for image analysis& apply image processing algorithms to real time applications 4. Analyze various image processing algorithms	

Name of the Programme	**: B.Sc. Computer Science**
Course Code	**: CSC-323**
Title of the Course	**: Statistical Package**
Number of Credits	**: 4(3T+1P)**
Effective from AY	**: 2024-25 (Revised version of 2023-24)**

Pre-requisites for the Course:	Nil	
Course Objectives:	1. Understand functions of statistical software package for managing variables and generate descriptive statistics to describe and analyze data through graphs and charts. 2. Test differences in sample means 3. Understand data structures and identify clusters in data 4. Identify principal components that are relevant from a host of variables.	
Content:		**No. of Hours**
	Unit-1: Getting started with the software: Introduction: Data Entry,Storing and Receiving Files, Generating New Variables, Managing Data-Listing cases,replacing missing values, computing new variables, recording variables, selecting cases,sortingcases,merging files, Graphs-creating and editing graphs and charts:Descriptive Statistics Procedures: Frequencies,Descriptive,Explore,Cross Tabulation.	**15**
	Unit-2: Hypothesis Testing for Means: T-tests : One sample test,Independent samples and paired samples t-test:Anova – One way analysis of variance with post hoc analysis, Two way analysis of variance.	**15**
	Unit-3: Testing for relationship between variables: Parametric and Non –Parametric Chi-square analysis:Bivariate Correlation and simple scatter Plot:LinearRegression,Multiple regression analysis with matrix scatter plot;Binary Logistics, Discriminant Analysis **Analysis of Structure:** Cluster analysis: Hierarchical Cluster analysis, K-Means Cluster Analysis; Factor analysis / Principal Components Analysis.	**15**
	Week 1 & 2: Calculate and interpret basic descriptive statistics. Use a dataset containing numerical variables (e.g., height, weight, age). ● Compute measures of central tendency (mean, median, mode). ● Calculate measures of dispersion (range, variance, standard deviation). ● Generate summary statistics for the dataset. ● Create visualizations such as histograms and box plots.	**04**

	Week 3 & 4: Perform hypothesis testing on a given dataset. Choose a dataset with two groups (e.g., experimental and control groups). ● Formulate a hypothesis and select an appropriate test (t-test, chi-square, etc.). ● Conduct the hypothesis test. ● Interpret the results and draw conclusions. ● Visualize the data to support your findings.	**04**
	Week 5 & 6: Conduct linear regression analysis to explore relationships between variables. Select a dataset with at least two numerical variables. ● Perform simple linear regression. ● Interpret the regression coefficients and assess model fit. ● Extend to multiple regression if the dataset has more than two variables. ● Visualize the regression line and residuals.	**04**
	Week 7, 8 & 9: Practice data cleaning and transformation techniques. Use a dataset with missing values, outliers, or categorical variables. ● Handle missing values (impute or remove). ● Identify and handle outliers. ● Convert categorical variables using one-hot encoding or label encoding. ● Normalize or standardize numerical variables.	**06**
	Week 10 & 11: Perform exploratory data analysis to understand the dataset. Choose a dataset with multiple variables. ● Generate summary statistics and visualizations. ● Explore relationships between variables using scatter plots or correlation matrices. ● Identify patterns and trends in the data. ● Conduct subgroup analyses if applicable.	**04**
	Week 12 & 13: Analyze time-series data. Use a time-series dataset (e.g., stock prices, temperature records). ● Plot the time series data. ● Conduct trend analysis and decomposition. ● Perform time-series forecasting using methods like ARIMA or exponential smoothing. ● Evaluate the accuracy of the forecasting model	**04**
	Week 14 & 15: Implement a classification model using machine learning. Select a dataset suitable for classification tasks. ● Split the dataset into training and testing sets. ● Choose a classification algorithm (e.g., logistic regression, decision trees). ● Train the model and evaluate its performance. ● Fine-tune hyperparameters if necessary.	**04**

Pedagogy:	**Suggested strategies for use to accelerate the attainment of the various course outcomes.** Lecture, demonstration of software application,videos,PowerPointpresentations,data analysis using software and lab assignments.
References/ Readings:	Main Reading: 1. Lawrence S. Meyers, Glenn C. Gamst, A. J. Guarino(2013), "*Performing Data Analysis using IBM SPSS*", 1st Edition, Wiley Publication. Additional Reading: 1. Darren George and Paul Malley(1999), "*SPSS for Windows Step by Step A simple Guide*", Prentice Hall 2. Kiran Pandya, Smruti Bulsari, Sanjay Sinha (2011), "*SPSS in Simple Steps*", Dreamtech Press 3. Dr Radha Mohan (2016), "*Using SPSS in research*", Neelkamal
Course Outcomes:	1. Remember the functions of statistical software package. 2. Understand the data structures. 3. Identify principal components that are relevant from a host of variables. and identify clusters in data. 4. Analyse data using various statistical techniques

Semester VI
Name of the Programme : B.Sc. Computer Science (Honours) Course
Course Code : CSC-304
Title of the Course : Cloud Computing
Number of Credits : 4 (3T + 1P)
Effective from AY : 2024-25 (Revised version of 2023-24)

Pre-requisites for the Course:	Data Structures, Object Oriented Programming, Operating Systems, Computer Networks	
Course Objectives:	1. Gain a comprehensive understanding of cloud computing concepts, models, and services. 2. Develop hands-on skills in deploying and managing applications on major cloud platforms. 3. Introduce containerization and orchestration technologies like Docker and Kubernetes. 4. Enable critical analysis of cloud solutions and their impact on businesses.	
Content:		**No. of Hours**
	Unit-1: Cloud Computing Fundamentals ● Introduction to Cloud Computing: Definition, history, evolution, benefits, and challenges. ● Cloud Service Models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS) and their comparison. ● Deployment Models: Public, Private, Hybrid cloud, and multi-cloud strategies. ● Cloud Architecture: Understanding distributed computing, virtualization, and resource management in the cloud. ● Cloud Security and Compliance: Secure cloud adoption, data protection, privacy, and regulatory considerations.	**15**
	Unit-2: Cloud Management and Technologies ● Identity and Access Management (IAM): Users, roles, permissions, and access control in the cloud. ● Networking in the Cloud: Virtual private clouds (VPCs), network management, and firewalls. ● Storage and Database Services: Object storage, block storage, file systems, and managed databases in the cloud. ● Compute Services: Virtual machines, containers, serverless computing, and auto-scaling. ● Management Tools and Automation: Cloud consoles, CLI tools, automation scripts, and configuration management.	**15**

	Unit-3: DevOps and Cloud Orchestration ● Introduction to DevOps: Principles, practices, and tools for continuous integration and continuous delivery (CI/CD) in the cloud. ● Infrastructure as Code (IaC): Tools like Terraform, Ansible, Chef for automated infrastructure provisioning. ● DevOps on Cloud Platforms: Integrating DevOps tools and practices with specific cloud platforms (e.g., AWS CodePipeline, Azure DevOps). ● Containerization: Docker for application packaging and isolation. ● Kubernetes: Container orchestration for scalable deployments and management. ● Monitoring and Optimization: Cloud monitoring tools, logs, and metrics for performance optimization and cost management.	**15**
	Practical Work	
	Week 1, 2 Introduction to AWS/Open source platform console and basic navigation. Create and manage virtual machines (EC2 instances or equivalent). Configure basic networking elements like security groups and virtual private clouds. Implement basic storage concepts: upload/download objects, create buckets.	**04**
	Week 3, 4, 5 Deploy a simple web application on a cloud platform. Connect the application to a cloud database service (e.g., MySQL, PostgreSQL). Implement CI/CD pipeline for automated application deployment. Explore containerization with Docker: create and run basic Docker images.	**06**
	Week 6, 7, 8 Implement load balancing for increased application availability. Configure auto-scaling to dynamically adjust resources based on demand. Explore serverless computing functionalities (AWS Lambda, Azure Functions etc.). Implement monitoring and logging tools for application performance insights.	**06**
	Week 9 &10 Design and deploy a multi-tier cloud application architecture. Explore advanced DevOps practices: GitOps, Infrastructure as Code. Secure cloud resources: IAM roles, security groups, access control. Backup and recovery: create and test backups of cloud resources	**04**
	Week 11 & 12 Introduction to Kubernetes: core concepts, deployments, services. Deploy a containerized application using Kubernetes clusters (e.g., Minikube). Manage pod and service configurations within Kubernetes. Explore advanced Kubernetes features: ReplicaSets, ingresses, secrets.	**04**

	Week 13 & 14 Cost optimization in the cloud: identify and control cloud spending. Migrating on-premises applications to the cloud: strategies and considerations. Disaster recovery planning and execution in cloud environments. Exploring different cloud providers: AWS, Azure, GCP - comparing features and pricing. Case studies: Analyze real-world applications of cloud computing across different industries. Industry trends: Serverless computing.	**04**
	Week 15 Cloud Project: Integrate learned concepts through a project, building and deploying a cloud-based application.	**02**
Pedagogy:	**Suggested strategies for use to accelerate the attainment of the various course outcomes.** 1. Interactive lectures: Engaging presentations supplemented with case studies, real-world examples, and guest lectures from industry professionals. 2. Hands-on labs: Weekly practical sessions using AWS or OpenStack platforms to reinforce theoretical learning and build practical skills. 3. Group projects: Collaborative assignments applying cloud technologies to solve real-world problems and develop teamwork skills. 4. Debates and discussions: Critical analysis of the impact of cloud computing on organizations, society, and ethics. 5. Virtual labs and cloud access: Provide students with hands-on experience through virtual labs or real cloud accounts where they can experiment and build cloud applications. 6. Online resources and forums: Supplement classroom teaching with curated online resources, discussion forums, and knowledge-sharing platforms for continuous learning and peer support. 7. Invite guest speakers: Bringing in professionals from the cloud industry can provide students with valuable insights and practical perspectives. 8. Organize hackathons or innovation contests: Challenge students to develop creative cloud-based solutions for real-world problems, fostering innovation and collaboration.	
References/ Readings:	Main Reading: 1. ChellammalSurianarayanan, Pethuru Raj Chelliah(2023), “*Essentials of Cloud Computing – A Holistic, Cloud Native Perspective*”, 2nd Edition, Springer. Additional Reading: 1. Kamal Kant Hiran, Ruchi Doshi, Dr. TemitayoFagbola, Mehul Mahrishi(2019), “*Cloud Computing – Master the concepts, architecture and applications with real world examples and case studies*”, 1st Edition, BPB 2. Toby Velte, Anthony Velte, Robert, Elsenpeter(2017), “*Cloud Computing A Practical Approach*”, Fourth Edition, Paperback, McGraw Hill 3. Yogesh Raheja(2018), “*Effective DevOps with AWS: Implement continuous delivery and integration in the AWS environment*”, 2nd Edition, PACKT	

	4. Soumiyajit Das Chowdhury(2023), "*Mastering DevOps in Kubernetes*", BPB
Course Outcomes:	On completion of the course, students will be able to: 1. Define key cloud computing terms and concepts. Identify different cloud deployment models and service offerings. Describe the benefits and challenges of adopting cloud solutions. 2. Understand the technical architecture of cloud platforms. Analyse the cost considerations and resource optimization techniques in cloud environments. Compare and contrast major cloud providers (AWS, Azure, GCP) based on specific criteria. 3. Deploy and manage applications on a selected cloud platform (AWS or Open Source Platform). Implement DevOps principles for continuous integration and deployment of cloud applications. Configure and utilize containerization technologies like Docker for application packaging. 4. Evaluate the suitability of cloud solutions for specific business needs and scenarios. Analyse the security considerations and risks associated with cloud deployments. Assess the scalability and performance of cloud-based applications.

Name of the Programme : B.Sc. Computer Science
Course Code : CSC-305
Title of the Course : Foundations of Data Science
Number of Credits : 4 (3T + 1P)
Effective from AY : 2024-25 (Revised version of 2023-24)

Pre-requisites for the Course:	Statistics, Probability Theory and Python Programming	
Course Objectives:	1. To get started with basics of data science and learn all aspects of data science in its entirety. 2. Gain a strong understanding of how data is generated, collected, stored, and analyzed. To conduct cleaning, organizing, and prepare data for analysis to gain insightful patterns. 3. Use powerful statistical tools to summarize data, identify trends, and draw meaningful conclusions. To transform data into visuals. Learn to create informative charts, graphs to communicate insights effectively. 4. Understand how machine learning algorithms can learn from data, make predictions, and solve complex problems.	
Content:		**No. of Hours**
	Unit-1: 1. Fundamentals of Data Science: Introduction to data science, Why Learn Data Science, What Is Big Data?, Examples Of Big Data, Types Of Big Data, Characteristics Of Big Data, Advantages Of Big Data Processing, Data Analytics Life Cycle, Types Of Data Analysis, Types Of Jobs In Data Analytics, Data Science Tools, Fundamental Arears Of Study In Data Science. 2. Data Preprocessing: Introduction To Data Preprocessing, Data Types And Forms, Possible Data Error Types, Various Data Preprocessing Operations. 3. Data Plotting And Visualization: Introduction To Data Visualization, Visual Encoding, Data Visualization Libraries, Basic Data Visualization Tools.	**15**
	Unit-2: 1. Statistical Data Analysis: Role Of Statistics In Data Science, Kinds Of Statistics, Probability Theory. 2. Machine Learning For Data Science: Overview Of Machine Learning, Supervised Machine Learning (Linear Regression, Logistic Regression, KNN Classification, Decision Tree Classification, Random Forest Classification, Naïve Bayes Classification), Unsupervised Machine Learning (Clustering Methods, Association Analysis), Reinforcement Learning.	**15**
	Unit-3: 1. Social Media Analytics: Overview of social media analytics, seven layers of social media analytics, social media analytics cycle, key social media analytics methods, accessing social media data, challenges to social media analytics. 2. Business Analytics: An overview of business analytics, the	**15**

	business analytics life cycle, basic tools used in business analytics, challenges faced in business analytics. 3. Ethics in Data Science: What is ethics, why ethics in data science is important, who regulates and owns our data, data science ethics, some ethical practices	
	Practical Work	
	Week 1 & 2 ● NumPy Foundation: Importing NumPy package, NumPy array attributes, Creating NumPy arrays, Accessing an element of a NumPy array, slicing in NumPy array, Array concatenation. ● Pandas and DataFrame: Importing Pandas, Pandas data structure, DataFrame, Some useful DataFrame functions, Handling missing values in DataFrame.	**04**
	Week 3 & 4 ● How to import data in Python: Importing text data, Importing csv data, Importing excel data, Importing JSON data, Importing pickled data, Importing compressed data. ● Cleaning of imported data: Analyzing missing values, dropping missing values, automatically filling missing values, how to scale and normalize data?, how to parse dates?, cleaning inconsistent data.	**04**
	Week 5 & 6 ● Data Preprocessing: Loading data and exploration, handling missing values, handling duplicates, data type correction, categorical feature encoding, feature scaling and normalization, data splitting. ● Combining data from multiple sources: Combining DataFrames vertically (stacking rows), Combining DataFrames horizontally (joining cloumns), without a common column, combining data from different file formats, combining data from databases.	**04**
	Week 7, 8 & 9 ● Create your own dataset (DataFrame, CSV file) ● Data Visualization: bar chart, line chart, histogram, scatter plot, box plot, heat map. (Use visualization tools such as Tableau, Gapminder, PowerBI) ● Web Scraping. ● Implementation of Linear Regression Machine Learning Algorithm	**06**
	Week 10 & 11 ● Implementation of Logistic Regression Machine Learning Algorithm ● Implementation of Decision Tree Machine Learning Algorithm	**04**
	Week 12 & 13 ● Implementation of K-Means Machine Learning Algorithm ● Implementation of K-Nearest Neighbors Machine Learning Algorithm	**04**

	Week 14 & 15 ● Handling Time Series Data: How to handle date and time?, transforming time series data, manipulating time series data, comparing time series growth rates. ● Data Cross Validation (train test and validation split)	**04**
Pedagogy:	**Suggested strategies for use to accelerate the attainment of the various course outcomes.** 1. The lecture method need not be only a traditional lecture method, but alternative effective teaching methods could be adopted to attain the outcomes. You may use 2. Video/Animation to explain various concepts. 3. Collaborative, Peer, Flipped Learning, etc. 4. Ask at least three HOT (Higher-Order Thinking) questions in the class, which promotes critical thinking. 5. Adopt Problem Based Learning (PBL), which fosters students' Analytical skills, and develops design thinking skills such as the ability to design, evaluate, generalize, and analyze information rather than simply recall it. 6. Introduce Topics in manifold representations. 7. Show the different ways to solve the same problem and encourage the students to come up with their own creative ways to solve them. 8. Discuss how every concept can be applied to the real world - and when that's possible, it helps improve the students' understanding 9. To promote self-learning, give at least one assignment (equivalent to 50% assignment weightage) where they can complete one MOOCs (certificate or equivalent) course out of lecture hour. Test their understanding through quizzes or presentations.	
References/ Readings:	Main Reading: 1. Dr. Gypsy Nandi, Dr. Rupam Kumar Sharma (2020), "*Data Science Fundamentals and Practical Approaches*", BPB Publications, India. Additional Reading: 2. Rachel Schutt & Cathy O'Neil (2019), "*Doing Data Science, Straight talk from the frontline", O'Reilly. 3. Joel Grus (2019), "*Data Science from Scratch, First Principles with Python*", O'Reilly. 4. Murtaza Haider(2016), "*Getting Started with Data Science, Making Sense of Data with Analytics",* Pearson India Education Services Pvt. Ltd.	
Course Outcomes:	On completion of the course, students will be able to: 1. Remember the basic concepts & terminologies of Data Science, Machine Learning Algorithms and Ethics in Data Science 2. Understanding of fundamental concepts and techniques in data science. Proficiency in data manipulation, analysis, and visualization using tools like Python 3. Apply Python programming concepts in performing Data Analytics and writing programs for Machine Learning Algorithms. 4. Perform analysis on Data Sets for machine learning algorithms.	

Name of the Programme : B.Sc. Computer Science
Course Code : CSC-306
Title of the Course : Software Quality Assurance
Number of Credits : 4 (3T + 1P)
Effective from AY : 2024-25 (Revised version of 2023-24)

Pre-requisites for the Course:	None	
Course Objectives:	1. Understanding of the principles, practices, and tools used in Software Quality Assurance (SQA). 2. To plan, implement, and manage quality assurance processes to ensure the delivery of high-quality software.	
Content:		**No. of hours**
	Unit-1: **Introduction to Software Quality Assurance** ● Overview of Software Quality Assurance ● Importance of SQA in the Software Development Life Cycle (SDLC) ● Role of SQA in achieving business goals **Fundamentals of Software Testing** ● Types of Testing (Unit, Integration, System, Acceptance, Regression) ● Black Box vs. White Box Testing ● Manual vs. Automated Testing **Test Planning and Documentation** ● Test Planning and Strategy ● Test Case Design and Execution ● Traceability Matrix	**15**
	Unit-2: **Automated Testing Tools** ● Introduction to Automated Testing ● Popular Automated Testing Tools (e.g., Selenium, JUnit, TestNG) **Performance Testing** ● Introduction to Performance Testing ● Load, Stress, and Scalability Testing ● Tools for Performance Testing (e.g., JMeter) **Defect Life Cycle and Tracking** ● Defect Life Cycle ● Defect Identification and Reporting ● Defect Tracking Tools (e.g., Jira, Bugzilla)	**15**
	Unit-3: **Continuous Integration and Continuous Deployment (CI/CD)** ● Introduction to CI/CD ● Integration of Testing in CI/CD ● Jenkins and other CI/CD tools **Software Quality Metrics and Measurement** ● Key Quality Metrics (e.g., defect density, test coverage) ● Importance of Measurement in SQA ● Continuous Improvement in SQA **Emerging Trends in SQA** ● Shift-Left Testing	**15**

	• DevOps and SQA • Artificial Intelligence in Software Testing	
	Practical Work	
	Week 1 & 2: Test Planning: • Task: Develop a test plan for a given software project. • Deliverables: Test plan document outlining test objectives, scope, resources, schedule, and test cases.	**04**
	Week 3 & 4: Test Case Design: • Task: Create test cases for a specific feature of a given application. • Deliverables: Test case document with detailed steps, expected results, and preconditions.	**04**
	Week 5 & 6: Manual Testing: • Task: Execute a set of test cases manually on a provided application. • Deliverables: Test execution report with details of pass/fail status, issues encountered, and suggestions for improvement.	**04**
	Week 7, 8 & 9: Automated Testing: • Task: Automate a set of test cases using a selected testing tool (e.g., Selenium, JUnit). • Deliverables: Automated test scripts, test execution report, and a comparison of manual vs. automated testing results.	**06**
	Week 10 & 11: Defect Reporting and Tracking: • Task: Identify defects during testing and report them using a bug tracking tool (e.g., Jira, Bugzilla). • Deliverables: Bug reports with detailed steps to reproduce, screenshots, and severity/priority assessments.	**04**
	Week 12 & 13: Performance Testing: • Task: Conduct performance testing on a web application (e.g., load testing, stress testing). • Deliverables: Performance test plan, execution results, and recommendations for improvement.	**04**
	Week 14 & 15: Continuous Integration/Continuous Deployment (CI/CD) Testing: • Task: Implement automated testing within a CI/CD pipeline. • Deliverables: CI/CD pipeline configuration, automated tests integration, and a report on the benefits of CI/CD in testing.	**04**
Pedagogy:	Hands-On Assignment, Peer Teaching, Mini Projects, Case Studies	
References/ Readings:	Main Reading: 1. Dorothy Graham, Rex Black, Erik van Veenendaal(2020),"*Foundations of Software Testing*", 4th Edition, Cengage Learning India Pvt. Ltd. Additional Reading: 1. Lisa Crispin and Janet Gregory(2009),"*Agile Testing: A Practical Guide for Testers and Agile Teams*", 1st Edition, Addison-Wesley 2. Jez Humble and David Farley(2010), "*Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*", 1st Edition, Addison Wesley	
Course Outcomes:	On completion of the course, students will be able to: 1. Remember the basic concepts & terminologies of software quality assurance.	

	2. Understand software testing and quality assurance as a fundamental component of software life cycle. 3. Apply Quality assurance activities using modern software tools. 4. Analyze prepared test plans and schedules for quality assurance project.

Name of the Programme : B.Sc. Computer Science
Course Code : CSC-324
Title of the Course : R Programming for Data Science
Number of Credits : 4 (3T + 1P)
Effective from AY : 2024-25 (Revised version of 2023-24)

Pre-requisites for the Course:	Basic Knowledge of Programming.	
Course Objectives:	1. Manipulate primitive data types in the R programming language using RStudio or Jupyter Notebook. 2. Control program flow with conditions and loops, write functions, perform character string operations, write regular expressions, handle errors. 3. Construct and manipulate R data structures, including vectors, factors, lists, and data frames. 4. Read, write, and save data files and scrape web pages using R. Understand and program machine learning algorithms.	
Content:		**No. of Hours**
	Unit-1: 1. Basics of R: Basic Math, Variables, Data Types, Vectors, Calling Functions, Function Documentation, Missing Data, Pipes. 2. Advanced Data Structures: DataFrames, Lists, Matrices, Arrays. 3. Reading Data into R: Reading CSV's, Excel Data, Reading from Databases, Data from other Statistical Tools, R Binary Files, Data Included with R, Extract Data from Web Sites. Reading JSON Data. 4. Statistical Graphics: Base Graphs, ggplot2. 5. Writing R Functions: Hello World, Function Arguments, Return Values, do.call.	**15**
	Unit-2: 1. Control Statements: if and else, switch, ifelse, Compound Tests. 2. Loops, the Un-R Way to Iterate: for loops, while loops, Controlling Loops. 3. Group Manipulation: Apply Family, aggregate, plyr, data.table. 4. Faster Group Manipulation with dplyr: pipes, tbl, select, filter, slice, mutate, summarize, group_by, arrange, do, dplyr with Databases. 5. Iterating with purr: map, map with specified types, Iterating over a DataFrame, map with multiple inputs. 6. Manipulating Strings: paste, sprint, Extracting text, Regular Expressions.	**15**
	Unit-3: 1. Linear Models: Simple Linear Regression, Multiple Regression 2. Generalized Linear Models: Logistics Regression, Poisson Regression, Other Generalized Linear Models, Survival Analysis.	**15**

	3. Nonlinear Models: Nonlinear Least Squares, Decision Trees, Random Forests. 4. Clustering: K-Means, PAM, Hierarchical Clustering.	
	Practical Work	
	Week 1 & 2 ● Introduction to RStudio ● Writing and Running R in Jupyter Notebooks ● Hello World with R using RStudio ● Basic Math with R using Jupyter Notebook	**04**
	Week 3 & 4 ● Hands-on Lab: Operators ● Hands-on Lab: Data Structures (Vectors, Matrices)	**04**
	Week 5 & 6 ● Hands-on Lab: Data Structures (DataFrames, Lists) ● Hands-on Lab: Loops and Conditions	**04**
	Week 7, 8 & 9 ● Hands-on Lab: Functions ● Hands-on Lab: Strings and Regular Expressions ● Hands-on Lab: Use statistical functions to compute function mean, standard deviation, statistical test on data(t-test)	**06**
	Week 10 & 11 ● Hands-on Lab: Loading data and saving files ● Hands-on Lab: Probability distribution ● Hands-on Lab: HTTP Request in R	**04**
	Week 12 & 13 ● Hands-on Lab: Webscraping in R ● Visualizing data using R: Bar Chart, Histogram, Boxplot, Scatter plot	**04**
	Week 14 & 15 ● Predictive modelling in R	**04**
Pedagogy:	**Suggested strategies for use to accelerate the attainment of the various course outcomes.** 1. The lecture method need not be only a traditional lecture method, but alternative effective teaching methods could be adopted to attain the outcomes. You may use 2. Video/Animation to explain various concepts. 3. Collaborative, Peer, Flipped Learning, etc. 4. Ask at least three HOT (Higher-Order Thinking) questions in the class, which promotes critical thinking. 5. Adopt Problem Based Learning (PBL), which fosters students' Analytical skills, and develops design thinking skills such as the ability to design, evaluate, generalize, and analyze information rather than simply recall it. 6. Introduce Topics in manifold representations. 7. Show the different ways to solve the same problem and encourage the students to come up with their own creative ways to solve them.	

	8. Discuss how every concept can be applied to the real world - and when that's possible, it helps improve the students' understanding 9. To promote self-learning, give at least one assignment (equivalent to 50% assignment weightage) where they can complete one MOOCs (certificate or equivalent) course out of lecture hour. Test their understanding through quizzes or presentations.
References/ Readings:	Main Reading: 1. Jared P. Lander(2018), "*R for Everyone, Advanced Analytics and Graphics*",2nd Edition, Pearson Education Inc.,. Additional Reading: 1. Tilman M. Davies(2016), "*The Book of R. A First Course in Programming and Statistics*", 1st Edition, No Starch Press, US. 2. Brett Lantz(2015), "*Machine Learning with R*", 2nd Edition, Packt Publishing Limited.
Course Outcomes:	On completion of the course, students will be able to: 1. Remember the basic concepts, terminologies and syntax of R programming. 2. Understanding of fundamental concepts and techniques in R programming. 3. Apply R programming concepts in data manipulation, visualization and machine learning algorithms. 4. Perform analysis on Data Sets for machine learning algorithms.

Name of the Programme : B.Sc. Computer Science
Course Code : CSC-325
Title of the Course : Computer Graphics
Number of Credits : 4(3T+1P)
Effective from AY : 2024-25 (Revised version of 2023-24)

Pre-requisites for the Course:	None	
Course Objectives:	1. Understand the core concepts of computer graphics. 2. Develop proficiency in implementing essential graphics algorithms. 3. Comprehend the underlying mathematical and physical principles of computer graphics. 4. Design and implement simple animations using keyframe animation, tweening, and particle systems. 5. Critically analyze the impact and ethical considerations of computer graphics in various domains.	
Content:		**No. of Hours**
	Unit-1: Introduction and 2D Graphics ● **Concepts:** Overview of the computer graphics, pipeline and component breakdown ● **Applications of computer graphics** in various industries such as video game, movie scene, virtual reality ● **Hardware and software fundamentals**: Understanding core technologies and tools ● **Rasterization and vector graphics**: Comparing fundamental rendering approaches. ● **Line drawing algorithms**: Bresenham's line algorithm, midpoint line algorithm ● **Polygon filling algorithms**: scanline filling, boundary filling ● **Clipping algorithms**: Cohen-Sutherland line clipping, Sutherland-Hodgman polygon clipping, point clipping, line clipping, text clipping	**15**
	Unit-2: 3D Graphics: Modeling and Transformation ● **3D modeling techniques**: Polygonal modeling, subdivision surfaces, procedural modeling, implicit surfaces ● **Geometric transformations**: Translation, rotation, scaling, shearing, Reflection, World Coordinates and Viewing Coordinates ● **Viewing transformations**: Orthographic projections, perspective projections ● **Camera models and frustums**	**15**
	Unit-3: Rendering and Animation: ● **Lighting models**: ambient occlusion, diffuse, specular ● **Shading techniques**: Gouraud shading, Phong shading ● **Texture mapping**: Enhancing surface details and realism with textures. ● **Antialiasing**: Smoothing edges and reducing visual artifacts for improved image quality. ● **Keyframe animation**: Defining key positions and interpolating movement for smooth animation.	**15**

	● **Tweening**: Automating intermediate frames and transitions between keyframes. ● **Particle systems**: Simulating dynamic effects like fire, smoke, and explosions.	
	Practical Work	
	Week 1 & 2 ● Write a program to draw the basic properties: point, segment, colored pixel, colored line. ● Write a program to draw triangle, square, circle, cube using basic shapes.	**04**
	Week 3 & 4 ● Write a program to implement Bresenham's line algorithm. ● Write a program to implement midpoint line algorithm.	**04**
	Week 5 & 6 ● Write a program to implement boundary filling. ● Write a program to implement point clipping. ● Write a program to implement line clipping.	**04**
	Week 7, 8 & 9 ● Write a program to implement text clipping. ● Write a program to Rotate the given input figure 2D or 3D through given angle.	**06**
	Week 10 & 11 ● Write a program to perform Reflection transformation on the given input figure 2D or 3D.	**04**
	Week 12 & 13 ● Write a program to perform Scaling transformation on the given input figure 2D or 3D. ● Write program to perform Translation transformation on the given input figure 2D or 3D.	**04**
	Week 14 & 15 ● Write a program to demonstrate shear transformation in different directions on a unit square situated at the origin.	**04**
Pedagogy:	**Suggested strategies for use to accelerate the attainment of the various course outcomes.** 1. The lecture method need not be only a traditional lecture method, but alternative effective teaching methods could be adopted to attain the outcomes. You may use a. Video/Animation to explain various concepts. b. Collaborative, Peer, Flipped Learning, etc. 2. Ask at least three HOT (Higher-Order Thinking) questions in the class, which promotes critical thinking. 3. Adopt Problem Based Learning (PBL), which fosters students' Analytical skills, and develops design thinking skills such as the ability to design, evaluate, generalize, and analyze information rather than simply recall it. 4. Introduce Topics in manifold representations. 5. Show the different ways to solve the same problem and encourage the students to come up with their own creative ways to solve them. 6. Discuss how every concept can be applied to the real world - and when that's possible, it helps improve the students' understanding	

	7. To promote self-learning, give at least one assignment (equivalent to 50% assignment weightage) where they can complete one MOOCs (certificate or equivalent) course out of lecture hour. Test their understanding through quizzes or presentations.
References/ Readings:	Main Reading: 1. A. P. Godse, D. A. Godse (2014), "*Computer Graphics*", Technical Publication Additional Reading: 1. Bhattacharya, Samit. (2018), "*Computer Graphics*", Oxford University Press. 2. Foley, J.D. and Van Dam, A. (1982), "*Fundamentals of Interactive Computer Graphics*", Addison-Wesley Publishing Company
Course Outcomes:	On completion of the course, students will be able to: 1. Remember the core concepts of computer graphics. 2. Implementing essential graphics algorithms. 3. Understand the underlying mathematical and physical principles of computer graphics. 4. Design and implement simple animations. 5. Understand the use of color, lighting, composition, and animation.

Name of the Programme : B.Sc. Computer Science
Course Code : CSC-326
Title of the Course : Business Intelligence
Number of Credits : 4 (3T+1P)
Effective from AY : 2024-25 (Revised version of 2023-24)

Pre-requisites for the Course:	None	
Course Objectives:	1. Gather relevant data from various sources within and outside the organization and integrate it into a central repository. 2. Analyze structured and unstructured data to derive meaningful insights and generate reports. 3. Enhance the decision-making process by providing decision-makers with actionable insights.	
Content:		**No. of Hours**
	Unit-1: **Introduction to Business Intelligence** Operational and Decision Support System, Data-Information-Knowledge-Decision Making-Action cycle. Basic definitions-Business Intelligence; Data warehousing, Business Intelligence architecture, Use and benefits of Business Intelligence. Knowledge Discovery in Databases: KDD process model, Data Pre-processing: Cleaning: Missing Values; Noisy Values; Inconsistent values; redundant values. Outliers, Integration, transformation, reduction, Discretization: Equal Width Binning; Equal Depth Binning, Normalization, Smoothing.	**15**
	Unit-2: **Introduction to Business Data Warehouse** Definition of Data warehouse, Logical architecture of Data Warehouse, Data Warehouse model- Enterprise warehouse; Data Marts; Virtual warehouse. Populating business Data Warehousing: data integration and extract, transform, load (ETL).	**15**
	Unit-3: **Designing Business Data Warehouse** OLTP and OLAP systems, Designing business information warehouse: Principles of dimensional modeling, Data cubes, Data cube operations, data cube schemas.	**15**
	Practical Work	
	Week 1 & 2 ● Import the legacy data from different sources such as (Excel, SQL Server, Oracle etc.) and load in the target system.	**04**
	Week 3 & 4 ● Perform the Extraction Transformation and Loading (ETL) process to construct the database in the SQL Server/Power BI.	**04**
	Week 5 & 6 ● Data Visualization from ETL Process	**04**
	Week 7, 8 & 9	**06**

	● Creating a Cube in SQL Server ● Apply the what – if Analysis for data visualization. Design and generate necessary reports based on the data warehouse data.	
	Week 10 & 11 ● Practical Implementation of Decision Tree using R ● k-means clustering using R	**04**
	Week 12 & 13 ● Prediction Using Linear Regression ● Data Analysis using Time Series Analysis	**04**
	Week 14 & 15 ● Data Modelling and Analytics with Pivot Table in Excel ● Data Analysis and Visualization using Advanced Excel	**04**
Pedagogy:	**Suggested strategies for use to accelerate the attainment of the various course outcomes.** 1. The lecture method need not be only a traditional lecture method, but alternative effective teaching methods could be adopted to attain the outcomes. You may use 2. Video/Animation to explain various concepts. 3. Collaborative, Peer, Flipped Learning, etc. 4. Ask at least three HOT (Higher-Order Thinking) questions in the class, which promotes critical thinking. 5. Adopt Problem Based Learning (PBL), which fosters students' Analytical skills, and develops design thinking skills such as the ability to design, evaluate, generalize, and analyze information rather than simply recall it. 6. Introduce Topics in manifold representations. 7. Show the different ways to solve the same problem and encourage the students to come up with their own creative ways to solve them. 8. Discuss how every concept can be applied to the real world - and when that's possible, it helps improve the students' understanding 9. To promote self-learning, give at least one assignment (equivalent to 50% assignment weightage) where they can complete one MOOCs (certificate or equivalent) course out of lecture hour. Test their understanding through quizzes or presentations.	
References/ Readings:	Main Reading**:** 1. Efraim Turban, Ramesh Sharda, Dursun Delen, David King(2013), "*Business Intelligence*", 2nd Edition, Pearson 2. Swain Scheps (2008), "*Business Intelligence for Dummies*", Wiley Publications (2008). 3. W. H. Inmon(2002), "*Building the Data Warehouse*", 3rd Edition, Wiley (1993). 4. Dunham, Margaret H (2006), "*Data Mining: Introductory and Advanced Topics*", Prentice Hall 5. Witten, Ian and Eibe Frank, Morgan Kaufmann (2011), "*Data Mining: Practical Machine Learning Tools and Techniques*", Second Edition, Morgan Kauffman Publishers Additional Reading**:** 1. Larissa T. Moss, Shaku Atre (2003), "*Business Intelligence Road Map"*, Addison-Wesley	

	2. Chuck Ballard, Dirk Herreman, Don Schau, Rhonda Bell, Eunsaeng Kim, Ann Valencic(1998), “*Data Modeling Techniques for Data Warehousing by IBM*”, International Technical Support Organization 3. , Han J. and Kamber M. Morgan Kaufmann Publishers, (2000), “*Data Mining: Concepts and Techniques*”, The Morgan Kaufmann Series in Data Management Systems 4. MacLennan Jamie, Tang ZhaoHui and Crivat Bogdan (2009), “*Data Mining with Microsoft SQL Server 2008*”, Wiley India Edition.
Course Outcomes:	On completion of the course, students will be able to: 1. Understand the role of business analytics in decision-making 2. Apply data warehouse concepts for data analysis and report generation 3. Design and create visually appealing and informative dashboards

Semester VII

Name of the Programme : B.Sc. Computer Science

Course Code : CSC 400

Title of the Course : Design and Analysis of Algorithms

Number of Credits : 4 (3T + 1P)

Effective from AY : 2024-25 (Revised version of 2023-24)

Pre-requisites for the Course:	1. Proficiency in a programming language such as Python, Java, C++, or a similar language. 2. Familiarity with fundamental data structures like arrays, linked lists, stacks, queues, and trees.	
Course Objectives:	1. To implement and analyze using Big O notation the basic sorting algorithms like bubble sort, selection sort, and insertion sort. 2. To encourage students to think critically about different algorithmic approaches for problem-solving. 3. To promote creativity in algorithm design.	
Content:		**No. of Hours**
	Unit-1: **Basics of Algorithm Analysis** ● Definition of Algorithms ● Importance of Algorithms in Computing ● Examples of Everyday Algorithms ● Time Complexity and Space Complexity ● Asymptotic Notation (Big O, Omega, Theta) ● Best, Average, and Worst-Case Analysis **Sorting and Searching Algorithms** ● Overview of Sorting Algorithms (Bubble Sort, Selection Sort, Insertion Sort) ● Overview of Efficient Sorting Algorithms (Merge Sort, Quicksort) ● Binary Search and Linear Search ● Concept of Divide and Conquer ● Recursion in Algorithms ● Master Theorem ● Example: Merge Sort	**15**
	Unit-2: **Greedy Algorithms** ● Knapsack Problem ● Minimum Spanning Tree: Kruskal's Algorithm ● Disjoint Sets and Kruskal's Algorithm ● Job Sequencing with Deadlines ● Heap ● Heap Sort ● Priority Queue ● Minimum Spanning Tree: Prim's Algorithm ● Huffman's Codes - Building Huffman Tree ● Huffman's Codes - Printing Huffman Codes **Dynamic Programming** ● Dynamic Programming: memoization ● Dynamic Programming: edit distance	**15**

	● Dynamic Programming: longest ascending subsequence ● Dynamic Programming: matrix multiplication ● Dynamic Programming: shortest paths: Bellman Ford ● Dynamic Programming: shortest paths: Floyd Warshal	
	Unit-3: **Backtracking** ● Rat in Maze ● n-Queens Algorithm ● Graph Coloring **Branch and Bound** ● Introduction to Branch and Bound ● 0/1 Knapsack Problem ● The 15 Puzzle Problem ● Solvability of 15 Puzzles **NP Completeness** ● Introduction to NP Completeness ● Reductions ● The Circuit Satisfiability Problem ● Approximation Algorithms ● The Vertex Cover Problem	**15**
	Practical Work	
	Week 1 & 2 Implement and compare the performance of bubble sort, insertion sort, and selection sort on various input sizes.	**04**
	Week 3 & 4 Implement and analyze the efficiency of merge sort and quicksort for large datasets.	**04**
	Week 5 & 6 Implement linear search and binary search and analyze their time complexity. Implement and compare the performance of different hash functions for hash table lookups.	**04**
	Week 7, 8 & 9 Solve the classic problem of calculating Fibonacci numbers using both recursive and dynamic programming approaches. Implement the Knapsack problem using dynamic programming and analyze its time complexity.	**06**
	Week 10 & 11 Implement the spanning trees algorithms: Prim’s and Kruskal’s	**04**
	Week 12 & 13 Implement Bellman Ford and Floyd Warshal’s algorithms dynamic programming algorithms	**04**
	Week 14 & 15 Implement n-queen’s and graph coloring algorithm	**04**

Pedagogy:	**Suggested strategies for use to accelerate the attainment of the various course outcomes.** 1. The lecture method need not be only a traditional lecture method, but alternative effective teaching methods could be adopted to attain the outcomes. You may use 2. Video/Animation to explain various concepts. 3. Collaborative, Peer, Flipped Learning, etc. 4. Ask at least three HOT (Higher-Order Thinking) questions in the class, which promotes critical thinking. 5. Adopt Problem Based Learning (PBL), which fosters students' Analytical skills, and develops design thinking skills such as the ability to design, evaluate, generalize, and analyze information rather than simply recall it. 6. Introduce Topics in manifold representations. 7. Show the different ways to solve the same problem and encourage the students to come up with their own creative ways to solve them. 8. Discuss how every concept can be applied to the real world - and when that's possible, it helps improve the students' understanding 9. To promote self-learning, give at least one assignment (equivalent to 50% assignment weightage) where they can complete one MOOCs (certificate or equivalent) course out of lecture hour. Test their understanding through quizzes or presentations.
References/ Readings:	Main Reading: 1. Ellis Horowitz, SatrajSahni and Rajasekharam(1998), "*Fundamentals of Computer Algorithms*", Galgotia publications Pvt. Ltd. Additional Reading: 1. M.T.Goodrich and R.Tomassia(2006), "*Algorithm Design: Foundations, Analysis and Internet examples*", 1st Edition, John Wiley and sons. 2. Aho, Ullman and Hopcroft (2002), "*Design and Analysis of algorithms*", 1st Edition, Pearson education.
Course Outcomes:	**On completion of the course, students will be able to:** 1. Recognize and remember the steps involved in various algorithmic paradigms. 2. Understand the difference between the lower and upper bounds of various problems and their Importance in deciding the optimality of an algorithm. 3. Analyze various techniques for efficient algorithm design (divide and conquer, greedy, and Dynamic programming algorithms) and able to apply them while designing algorithms. 4. Apply backtracking and branch and bound techniques to deal with some hard problems.

Name of the Programme	**: B.Sc. Computer Science**
Course Code	**: CSC-401**
Title of the Course	**: Artificial Intelligence**
Number of Credits	**: 4 (3T + 1P)**
Effective from AY	**: 2024-25 (Revised version of 2023-24)**

Pre-requisites for the Course:	Nil	
Course Objectives:	1. To understand the concept of Artificial Intelligence (AI). 2. To learn various important search strategies, Planning &knowledge representation in AI. 3. To acquaint with the fundamentals of Learning, Computer Vision & Expert Systems. 4. To develop a mind to solve real world problems in AI.	
Content:		**N0. of Hours**
	Unit-1: Introduction to AI ● Definition of AI, Historical development of AI **State Space Search** ● Breadth First Search, Depth First Search, Depth Bounded DFS (DBDFS), Depth First Iterative Deepening (DFID). **Heuristic Search** ● Heuristic Functions, Best First Search, Hill Climbing, Variable Neighbourhood Descent **Optimal Search** ● A* algorithm, Iterative Deepening A*, Recursive Best First Search. **Problem Decomposition** ● Goal Trees, Rule Based Systems, Rule Based Expert Systems. **Planning** ● STRIPS, Forward and Backward State Space Planning, Goal Stack Planning, Plan Space Planning.	**15**
	Unit-2: Constraint Satisfaction ● N-Queens, Constraint Propagation. Game Playing: Alpha-Beta Pruning. **Knowledge Based Reasoning** ● Agents, Facets of Knowledge **Logic and Inferences** ● Formal Logic, Propositional Logic, Resolution method in Propositional Logic, and First Order Logic ● Resolution Refutation in FOL, Forward & Backward Chaining. **Knowledge Representation** ● Frames, Semantic nets.	**15**
	Unit-3: Applications of AI ● **Learning**: Introduction, Types of Learning: Rote Learning, Learning by taking advice, Learning by Induction ● **Computer Vision**: Human Vision Processing, Edge detection, The Waltz algorithm. ● **Expert System**: Architecture of Expert System, Role of Expert system in Knowledge acquisition.	**15**

	● **AI and Robotics :**Introduction to Robotics, AI applications in Robotics	
	Practical Work (Experiments to be performed from the following list in Java/Python.)	
	Week 1 & 2 1. Program to implement depth first search algorithm. 2 . Program to implement breadth first search algorithm.	**04**
	Week 3 & 4 3 . Program to implement Best First Search algorithm. 4 . Program to simulate 4-Queen / N-Queen problem.	**04**
	Week 5 & 6 5 . Program to implement alpha beta search. 6 . Program for implementation Hill climbing problem.	**04**
	Week 7, 8 & 9 7 . Program to implement A* search algorithm. 8 . Program to solve water jug problem. 9 . Program to simulate tic – tac – toe game using min-max algorithm.	**06**
	Week 10 & 11 10. Program to implement Constraint satisfaction problem 11. Program to solve Missionaries and Cannibals problem.	**04**
	Week 12 & 13 12. Program to implement Traveling salesman problem. 13. Program to implement Expert System using prolog.	**04**
	Week 14 & 15 14. Program for simulation of Logical functions using Neural networks	**04**
Pedagogy:	**Suggested strategies for use to accelerate the attainment of the various course outcomes.** 1. Interactive lectures with real-world case studies and practical examples. 2. Problem-solving sessions and challenges to reinforce critical thinking and application skills. 3. Utilization of online resources and visualization tools to enhance learning. 4. Collaborative group work and discussions to encourage deeper understanding 5. Adopt Problem Based Learning (PBL), which fosters students' Analytical skills, and develops design thinking skills such as the ability to design, evaluate, generalize, and analyze information rather than simply recall it. 6. Show the different ways to solve the same problem and encourage the students to come up with their own creative ways to solve them. 7. Discuss how every concept can be applied to the real world - and when that's possible, it helps improve the students' understanding	
References/ Readings:	Main Reading: 1. Deepak Khemani (2013), "*A First Course in Artificial Intelligence*", ISBN: 978-1-25-902998-1, McGraw Hill Education (India) 2. Ela Kumar(2008), "*Artificial Intelligence*", I.K. International Publishing House Pvt. Ltd.	

	Additional Reading: 1. Stuart Russell and Peter Norvig (2003), “*Artificial Intelligence: A Modern Approach*”, 3rd Edition, ISBN :10: 0136042597, Pearson 2. Elaine Rich, Kevin Knight and Nair(2017), “*Artificial Intelligence*”, , ISBN-978-0-07-008770-5,Tata McGraw Hill 3. Nilsson Nils J(1998) , “*Artificial Intelligence: A new Synthesis*”, Morgan Kaufmann Publishers Inc. 4. Patrick Henry Winston(2002), “Artificial Intelligence”, Pearson Education
Course Outcomes:	**On completion of the course, students will be able to:** 1. Remember the basic concepts of AI 2. Understand the structure of an A.I. Problem and requirement, representation and application of the knowledge to solve an AI problem, planning of heuristic based search algorithms 3. Apply heuristic based state space search techniques, knowledge and planning models for AI applications 4. Analyse the suitability of knowledge models, search algorithms and the machine learning algorithms to solve any AI application and design a solution strategy in any domain to transfer human expertise into machine.

Name of the Programme : B.Sc. Computer Science
Course Code : CSC-402
Title of the Course : Formal Language and Automata Theory
Number of Credits : 4(3T+1P)
Effective from AY : 2024-25 (Revised version of 2023-24)

Pre-requisites for the Course:	Knowledge of Mathematical Foundations for Computer Science and Computer Programming	
Course Objectives:	1. To construct Deterministic Finite Automata (DFA), Non-deterministic Finite Automata (NDFA), Pushdown Automata (PDA), and Turing Machines to represent and analyze various languages, sets, and patterns. 2. To develop the skill to generate languages and regular expressions corresponding to DFA, PDA, Turing Machines, and grammars. This involves understanding the relationships between different formal language models. 3. To distinguish between decidability and undecidability, gaining insights into the limits of computation. They will comprehend the concepts of recursively enumerable and undecidable languages and recognize the significance of the Church-Turing thesis. 4. To be proficient in simulating the behavior of DFA, PDA, and Turing Machines through computer programming. This involves practical implementation, debugging, and testing, enhancing their understanding of the theoretical concepts in a hands-on manner.	
Content:		**No. of Hours**
	Unit=1: Automata Theory, Formal Languages, Regular Sets and Regular Grammars **Automata Theory** - DFA, NDFA, Equivalence of NDFA and DFA, Mealy and Moore Models, Transforming a Moore Machine into Mealy Machine, Minimization of Finite Automata **Formal Languages-** Grammar, Derivations and Language Generated by Grammar, Chomsky Classification of Languages, Languages and their Relation, Recursive and Recursively Enumerable Sets, Operations on Languages, Languages and Automata **Regular Sets and Regular Grammars-**Regular Expressions and Identities, Transition System containing Λ-moves, NDFAs and Regular Expressions, Arden's Theorem, Kleene's Theorem, Conversion of Nondeterministic System into Deterministic System, Algebraic Method using Arden's Theorem, Construction of Finite Automata Equivalent to Regular Expression, Equivalence of Two Finite Automata, Equivalence of Two Regular Expressions, Pumping Lemma for Regular Sets and Application for Regular Languages, Closure Properties of Regular Sets, Construction of Regular Grammars for DFA, Construction of Transition system for Regular Grammar,	**15**

	Unit-2: Context-Free Languages, Turing Machines and Linear Bounded Automata **Context- Free Languages-** CFL and Derivation Trees, Ambiguity, Reduced Grammars, Elimination of Null and Unit Productions, Chomsky Normal Form, Greibach Normal Form, Pumping Lemma for CFLs Pushdown Automata- Acceptance by PDA(by final state and empty store), PDA and CFLs **Turing Machines and Linear Bounded Automata-** Turing Machine Representatoins (Instantaneous Description, Transition Table, Transition Diagram), Languages Accepted by Turing Machines, Design and Construction, Turing Machine Variants(Multitape, Nondeterministic, Enumerators), Linear Bounded Automaton Model, LBA and Context Sensitive Languages, Turing Machines and Type 0 Grammars.	**15**
	Unit-3: Decidability and Reducability **Decidability-** Algorithm Definition,Hilbert's Problem, Halting Problem of Turing Machine, Church-Turing Thesis, Recurively Enumerable Language, Decidable and Undecidable Language (Regular,Context-Free, Turing Recognizable), Diagonalization Method, Countable and Uncountable Sets, Turing-Unrecognizable Language. **Reducability-** Concept, Undecidability of (Halting Problem, Turing Machine recognizing Regular Language, Languages of two TMs are the same)	**15**
	Practical Work	
	Week 1 & 2 ● Construction of DFA and NDFA for Languages/Sets ● Conversion of NDFA to DFA	**04**
	Week 3 & 4 ● Implementation of DFA for a set using Computer Programming	**04**
	Week 5 & 6 ● Minimization of DFA ● Equivalence of two DFA	**04**
	Week 7, 8 & 9 ● Construction of Grammars for Languages/Sets ● Generation of Regular Expressions for DFA ● Construction of DFA from Regular Expression	**06**
	Week 10 & 11 ● Construction of Grammars for Regular Expressions ● Equivalence of Two Regular Expressions	**04**
	Week 12 & 13 ● Construction of PDA for Languages/ Sets ● Implementation of PDA for a set using Computer Programming	**04**
	Week 14 & 15 ● Construction of Turing Machine for Languages/Sets ● Implementation of Turing Machine for a set using	**04**

	Computer Programming	
Pedagogy:	**Suggested strategies for use to accelerate the attainment of the various course outcomes.** 1. The lecture method need not be only a traditional lecture method, but alternative effective teaching methods could be adopted to attain the outcomes. You may use 2. Video/Animation to explain various concepts. 3. Collaborative, Peer, Flipped Learning, etc. 4. Ask at least three HOT (Higher-Order Thinking) questions in the class, which promotes critical thinking. 5. Adopt Problem Based Learning (PBL), which fosters students' Analytical skills, and develops design thinking skills such as the ability to design, evaluate, generalize, and analyze information rather than simply recall it. 6. Introduce Topics in manifold representations. 7. Show the different ways to solve the same problem and encourage the students to come up with their own creative ways to solve them. 8. Discuss how every concept can be applied to the real world - and when that's possible, it helps improve the students' understanding 9. To promote self-learning, give at least one assignment (equivalent to 50% assignment weightage) where they can complete one MOOCs (certificate or equivalent) course out of lecture hour. Test their understanding through quizzes or presentations.	
References/ Readings:	Main Reading: 1. Mishra, K. L. P., & Chandrasekaran, N. (2006), "*Theory of computer science: automata, languages and computation",* PHI Learning Pvt. Ltd.. Additional Reading: 1. Sipser, M. (2012),"*Introduction to the Theory of Computation*", Third Edition, Cengage Learning. 2. Hopcroft et al. (2001)*, "Introduction* to automata *theory*, languages, and *computation",* 2ndEdition, Addison-Wesley	
Course Outcomes:	**On completion of the course, students will be able to:** 1. Construct DFA, PDA, Turing Machine and Grammars for Languages/Sets/Patterns 2. Generate Language/ Regular Expressions for DFA, PDA, Turing Machine and Grammars 3. Distinguish between Decidable and Undecidable Languages 4. Simulate DFA, PDA and Turing Machines using Computer Programming.	

Name of the Programme : B.Sc. Computer Science
Course Code : CSC-403
Title of the Course : Network Security
Number of Credits : 4(3T + 1P)
Effective from AY : 2024-25 (Revised version of 2023-24)

Pre-requisites for the Course:	Knowledge of Computer Networks	
Course Objectives:	1. To understand the fundamental concepts and challenges of network security and gain practical knowledge of classical encryption techniques and steganography. 2. To analyze and compare symmetric and asymmetric key cryptography models. 3. To explore message authentication and public key infrastructure applications. 4. To be familiar with essential network security tools and techniques.	
Content:		**No. of hours**
	Unit-1: Introduction to Network Security and Classical Encryption Techniques **A) Introduction:** ● The need for security ● Security approaches ● Security attacks ● Security Services ● Security Mechanisms **B) A Model for Network Security:** ● Symmetric and asymmetric models **C) Encryption techniques:** ● Substitution techniques – Caesar, Mono alphabetic, Homophonic, Polygram, Polyalphabetic, Playfair, Hill, Vernam cipher ● Transposition techniques – Rail fence, Simple Columnar **D) Steganography:** ● Difference between steganography and cryptography ● Techniques **–** Text steganography, Image steganography ● Applications ● Limitations	**15**
	Unit-2: Algorithm Types, Symmetric and Asymmetric Key Cryptography ● **Algorithm types and Modes:** ○ Block Cipher Operation ○ Electronic Code Book ○ Cipher Block Chaining ○ Block Cipher Principles ● **Symmetric Key Cryptography** ○ Overview of Symmetric Key Cryptography ○ Diffie Hellman Key Exchange Algorithm ○ The Data Encryption Standard	**15**

	● **Asymmetric Key Cryptography:** ○ Overview Asymmetric Key Cryptography ○ RSA algorithm ○ Comparison between symmetric and asymmetric key cryptography ○ Digital Signatures	
	Unit-3: Cryptographic Hash Functions, Public Key Infrastructure and Network Security Techniques **A) Cryptographic Hash Functions:** ● Message Digest, MD5, SHA-1 and SHA-512 ● Message Authentication Requirements ● Message Authentication Functions, MAC, HMAC ● Applications of Cryptographic Hash Functions **B) Public Key Infrastructure (PKI) and Digital Certificates** ● Introduction to PKI ● Digital Certificates - Technical Details, Certification Authority, Digital Certificates Creation and verification. ● Certificate Hierarchies and Self-signed Digital Certificates **C) Introduction to Network Security Techniques** ● IP Security ● Firewalls ● Intrusion Detection ● Virtual Private Networks	**15**
	Practical Work	
	Week 1 & 2 ● Implementation of encryption techniques – Caesar, Polyalphabetic, Vernam	**04**
	Week 3 & 4 ● Implementation of encryption techniques: Playfair, Rail Fence	**04**
	Week 5 & 6 ● Perform Steganography using simple DoS commands and tools such as OpenStego	**04**
	Week 7 & 8 Password Cracking and Policy Enforcement: ● Perform password cracking exercises using tools like John the Ripper or ophcrack. ● Implement and enforce strong password policies.	**04**
	Week 9 & 10 Network Mapping and Discovery: ● Use tools like Nmap to scan and discover devices on a network. ● Identify open ports, services running and potential vulnerabilities. Packet Sniffing and Analysis: ● Use Wireshark to capture and analyze network traffic. ● Identify protocols, inspect packets, and recognize potential security threats.	**04**

	Week 11 & 12 ● Use openSSl/JCrypt tool (or any other equivalent) and demonstrate asymmetric, symmetric cryptography, hashing and digital/PKI signatures / certificates.	**04**
	Week 13, 14 & 15 Installing and configuring ● Firewall. ● VPN. ● Intrusion detection system.	**06**
Pedagogy:	1. Interactive Lectures to keep students engaged. This could include asking questions or having short discussions. Encourage students to share their thoughts and experiences related to the lecture topics. 2. Interactive demonstrations to illustrate complex concepts. 3. Using visual aids to enhance understanding, especially for topics related to symmetric and asymmetric cryptography and hash functions. 4. Encouraging peer-to-peer learning. 5. Implement continuous assessment methods, such as quizzes, short tests or online discussions, to gauge students' understanding.	
References/ Readings:	Main Reading: 1. Stallings William(2005), *“Cryptography and Network Security: Principles and Practices*”, 5th edition, Prentice Hall. 2. KahateAtul(2017), *“Cryptography and Network Security*”,3rd Edition, Tata McGraw-Hill. Additional Reading: 1. Behrouz A Forouzan(2007), “*Cryptography and Network Security*”, 1st Edition, McGraw Hill 2. Charlie Kaufman, Radia Periman, Mike Speciner, “*Network Security*”,2nd Edition, Pearson Education.	
Course Outcomes:	**On completion of the course, students will be able to:** 1. Remember the basic concepts and definitions related to network security. 2. Understand the motivations, mechanisms, strengths, and weaknesses of various security attacks, approaches, and techniques employed in protecting networked systems. 3. Apply and implement encryption techniques, message authentication functions, network security tools like firewalls and VPNs and digital certificates to secure communication and protect network resources. 4. Analyze the strengths, weaknesses, trade-offs and suitability of different security approaches, algorithms and techniques based on specific contexts and security goals.	

Name of the Programme : B.Sc. Computer Science
Course Code : CSC-411
Title of the Course : Ethical Hacking
Number of Credits : 4(3T+1P)
Effective from AY : 2024-25 (Revised version of 2023-24)

Pre-requisites for the Course:	Knowledge of operating systems, programming, web technology and database management systems.	
Course Objectives:	1. To understand the fundamental concepts of ethical hacking. 2. To understand the tools and practices used in ethical hacking. 3. To gain knowledge on the techniques to protect systems from data breaches.	
Content:		**No. of Hours**
	Unit-1: Foundations of Ethical Hacking **The Ethical Hacker Mindset** ● Importance of cybersecurity and ethical hacking ● Principles of ethical hacking ● Legal and professional responsibilities ● Hacking phases and attack categories **Reconnaissance & Information Gathering** ● Foot printing techniques: Open-source intelligence, OSINT tools ● Network scanning: Active and passive scanning, port discovery ● Enumeration: User accounts, resources, applications, services ● Countermeasures and threat mitigation strategies. **System Hacking** ● Common system vulnerabilities: Password attacks, buffer overflows, privilege escalation ● Frameworks and penetration testing tools ● Password cracking: Hashing algorithms, rainbow tables, cracking tools	15
	Unit-2: Network Security & Defenses **Sniffing & Network Attack** ● Active and passive sniffing techniques: Network traffic analysis, ARP spoofing ● Denial-of-Service attacks: Types, tools, countermeasures ● Session hijacking: Spoofing, sequence prediction, hijacking tools ● Network security monitoring and intrusion detection systems **Wireless Network Security** ● Introduction to wireless technologies: 802.11 standards, WEP, WPA ● Wireless network vulnerabilities: WEP cracking, WPA vulnerabilities ● Wireless attack tools and penetration testing methodologies ● Securing wireless networks: Encryption,	15

	authentication, best practices **Social Engineering Hacking** ● Common social engineering techniques: Pretexting, phishing, baiting ● Online social engineering: Social media manipulation, identity theft ● Reverse social engineering: Building trust, gaining information	
	Unit-3: Web & Application Security **Web Server and Application Vulnerabilities:** ● Common web server vulnerabilities: Apache, IIS, vulnerabilities; ● Web application security flaws: SQL injection, XSS, CSRF ● Penetration testing methodologies for web applications ● Web application security tools and scanners **Secure Password Management & Cryptography:** ● Password authentication methods: HTTP Basic, Digest, NTLM, Forms-based ● Password cracking tools and techniques: Hashing algorithms, rainbow tables ● Secure password best practices: Password complexity, multi-factor authentication ● Introduction to cryptography: Symmetric and asymmetric encryption algorithms **Advanced Threats & Defense Strategies** ● Emerging threats: Malware, ransomware, zero-day vulnerabilities ● Cloud security: Vulnerabilities and security best practices ● Incident response and breach management ● Security automation and threat intelligence	**15**
	Practical Work	
	Week 1-2: Introduction to Kali Linux ● Install Kali Linux on a virtual machine. ● Explore the Kali Linux menu, network commands and understand the purpose of various tools. **Foot printing & Information Gathering** ● Use OSINT tools to gather information about a target website. ● Practice network scanning techniques with tools. ● Enumerate user accounts and resources on a test network.	**04**
	Week 3-6: System Hacking and Password Cracking ● Identify common system vulnerabilities on a target system. ● Use penetration testing tools to exploit vulnerabilities and gain unauthorized access. ● Perform password cracking using various techniques and tools.	**08**

	Sniffing and Network Attacks ● Conduct active and passive sniffing using tools. ● Perform ARP spoofing to intercept and analyze network traffic. ● Simulate a Denial-of-Service (DoS) attack.	
	Week 7-9: Wireless Network Security ● Explore wireless technologies and standards (e.g., 802.11). ● Crack WEP encryption on a simulated wireless network. ● Implement security measures to secure a wireless network. **Web Server and Application Vulnerabilities** ● Identify and exploit common vulnerabilities in web servers ● Perform SQL injection attacks on a web application. ● Use penetration testing techniques to assess web application security.	**06**
	Week 10-13: Password Cracking & Cryptography ● Understand hashing algorithms and the limitations of password complexity. ● Use rainbow tables to crack offline password hashes. ● Implement secure password hashing and storage techniques in a web application. **Cryptographic Algorithms & Tools** ● Experiment with symmetric and asymmetric encryption algorithms. ● Use digital signatures and certificates for secure communication. ● Conduct a security audit of a web application for cryptographic implementation.	**08**
	Week 14-15: Social Engineering Hacking ● Execute common social engineering techniques such as phishing or pretexting. ● Explore online social engineering through manipulation of social media platforms. ● Practice reverse social engineering by building trust and extracting information.	**04**
Pedagogy:	Lab Exercises, Problem-based Learning, Simulations, Case Studies	
References/ Readings:	Main Reading: 1. Osborne, S., Scambray, J., & McClure, S.(2009),“*Hacking Exposed: Network Security Secrets and Solutions*”, 6th Edition, McGraw-Hill. 2. Scambray, J., Hatch, M., & Kurtz, G. (2008),“*Hacking Exposed: Linux Security Secrets and Solutions*”, 3rd Edition, McGraw-Hill. 3. McClure, S., Scambray, J., & Kurtz, G. (2007),“*Hacking Exposed: Windows Security Secrets and Solutions*”, 3rd Edition, McGraw-Hill. 4. McClure, S., Scambray, J., & Kurtz, G.,“*Hacking Exposed: Web Application Security Secrets and Solutions*”, McGraw-Hill.	
Course Outcomes:	**On completion of the course, students will be able to:** 1. Understand the concepts of ethical hacking for applications.	

	2. Analyze real-world security scenarios and apply ethical hacking principles to identify and assess potential vulnerabilities. 3. Use tools for penetration testing and security assessment.

Semester VIII
Name of the Programme : B.Sc. Computer Science
Course Code : CSC-404
Title of the Course : Machine Learning
Number of Credits : 4(4T)
Effective from AY : 2024-25 (Revised version of 2023-24)

Pre-requisites for the Course:	Nil	
Course Objectives:	1. To provide a solid understanding of fundamental ML concepts and algorithms. 2. To develop skills in building and evaluating ML models for real-world problems. 3. To cultivate critical thinking and problem-solving abilities in the context of ML applications. 4. To introduce students to popular ML tools and libraries for practical implementation. 5. Explore the ethical implications and challenges associated with machine learning applications.	
Content		**No. of Hours**
	Unit 1: Foundations of Machine Learning **What is Machine Learning?** Definition, types (supervised, unsupervised, reinforcement), key paradigms (learning from data, generalization) **Data and Preprocessing** Data preparation and exploration (cleaning, scaling, visualization); Feature engineering: feature scaling, selection, transformation; Model evaluation metrics: error analysis, cross validation, confusion matrix, ROC curve **Foundations of Learning** Hypothesis spaces and model selection; Overfitting and underfitting; Bias-variance trade-off	**15**
	Unit-2: Supervised Learning **Regression** Linear regression: assumptions, parameter learning, regularization (L1, L2); Logistic regression: binary classification, multi-class classification **Decision Trees and Ensemble Methods** Decision tree induction, pruning, ensembles (bagging, boosting); Random forests: advantages and limitations **Other Classifiers** Support vector machines: maximum margin hyperplane, kernels; k-Nearest Neighbours: distance-based classification; Naive Bayes classifier: Bayesian inference for classification	**15**

	Unit-3: Unsupervised Learning **Clustering** k-Means clustering: algorithm, initialization, metrics; Hierarchical clustering: agglomerative and divisive techniques; Density-based clustering: DBSCAN, OPTICS **Dimensionality Reduction** Principal component analysis (PCA): eigenvalues, variance maximization; Factor analysis: statistical modeling approach; Manifold learning for non-linear data representation **Other Unsupervised Methods** Anomaly detection: outlier identification techniques; Association rule learning: market basket analysis, frequent pattern mining	**15**
	UNIT-4: Advanced Topics and Applications **Deep Learning** Artificial neural networks: architectures, activation functions, backpropagation; Convolutional neural networks: image recognition and applications; Recurrent neural networks for sequence data processing **Ensemble Learning** Introduction, Ensemble methods: bagging, boosting, random forests, Stacking, blending, theoretical advantages, and applications. **Ethical Considerations and Social Impact** Bias and fairness in machine learning models; Explainability and interpretability of models; Privacy and security concerns	**15**
Pedagogy:	**Suggested strategies for use to accelerate the attainment of the various course outcomes.** ● Interactive lectures with real-world case studies and practical examples. ● Problem-solving sessions and challenges to reinforce critical thinking and application skills. ● Utilization of online resources and visualization tools to enhance learning. ● Collaborative group work and discussions to encourage deeper understanding.	
References/ Readings:	Main Reading: 1. Alpaydin (2013),“*Introduction to Machine Learning*”, Pearson Education India. 2. Murphy, K. P. (2012),“*Machine learning: A probabilistic perspective*”, MIT press. 3. Müller, A. C., & Guido, S. (2017), “*Introduction to machine learning with Python: A guide for data scientists*”, O'Reilly Media. Additional Reading: 4. Géron, A. (2019), “*Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*”, O'Reilly Media.	

	5. Raschka, S., &Mirjalili, V. (2019),“*Python machine learning*”, Third Edition,Packt Publishing Ltd.
Course Outcomes:	**On completion of the course, students will be able to:** 1. Identify and formulate different types of machine learning problems. 2. Gain proficiency in implementing supervised learning algorithms like linear regression, logistic regression, and decision trees. 3. Understand and apply unsupervised learning techniques like clustering and dimensionality reduction. 4. Evaluate the performance of ML models, analyze errors, and tune hyperparameters for optimal results.

Name of the Programme : B.Sc. Computer Science
Course Code : CSC 405
Title of the Course : Internet of Things
Number of Credits : 4 (4T)
Effective from AY : 2024-25 (Revised version of 2023-24)

Pre-requisites for the Course:	Basic programming knowledge, Computer Architecture, and also knowledge of Networking and Embedded Systems.	
Course Objectives:	1. To Introduce concepts for internet of things and the different devices involved in IOT. 2. To introduce cloud concepts and its use in IOT. 3. Gain hands on experience of working with different sensors/actuators and their use in IOT projects. 4. To gain knowledge of Arduino, NodeMcu, Raspberry pi Boards and to develop IOT projects by integrating these boards with a cloud platform.	
Content:		**No. of Hours**
	Unit-1: IOT Concepts Introduction Definition, modern day IoT applications, Baseline technologies- M2M,WoT, IOT categories- industrial and consumer, IOT components **Sensors and Actuators** Sensors, transducers, sensor features, resolution, analog sensors, digital sensors, scalar sensors, vector sensors, sensor types. Actuators-types-hydraulic, pneumatic, electrical, thermal/mechanical,motors-DC, Servo, Stepper, relays, motor drivers for interfacing **IOT Networks** IoTWF Standardized Architecture, Connectivity Protocols-MQTT,SMQTT; communication protocols-IEEE 802.15.4,802.11,LORA wireless protocol, ZigBee	**15**
	Unit-2: IOT Boards: Arduino Introduction to Arduino Programming-features of arduino, Arduino IDE, sketch, sketch structure, supported data types, Arduino function libraries, operators, control statements, arrays, String functions, Interrupts, sensor interface with Arduino, DHT sensor library, types of motor actuators, Arduino servo library **Raspberry Pi and comparative study** Introduction to Raspberry Pi – specifications, GPIOs, Features of EsP8266 ,comparative studies of Arduino uno, raspberry pi, nodemcu boards and their applications	**15**
	Unit-3: Cloud Technology-Part I **Introduction to cloud computing** definition, characteristics, components, service models-IaaS, Pass, SaaS, Deployment models-public, private, hybrid, open source and commercial clouds-examples, facilities offered **Cloud computing case studies** Microsoft Azure-features, Azure as PaaS, Azure as Iaas; OPenStack-components and features, Firebase cloud service features	**15**

	Unit-4: Cloud Technology- Part II **Visual tool for wiring IOT** NodeRed, its features, installing on Raspberry pi **Wireless sensornetworks** definition, limitations; Sensor cloud-definition, difference with WSN, Actors in sensor cloud, architecture **Fog computing** Introduction, why use fog computing, when to use fog computing, architecture of fog, fog nodes, working of fog, applications of fog	**15**
Pedagogy:	**Suggested strategies for use to accelerate the attainment of the various course outcomes.** ● Interactive lectures with real-world case studies and practical examplesActivities/Demonstration can be conducted On ● Blinking an LED . Traffic lights using Arduino board. ● Night security light using PIR motion sensor and photo resistor ● Arduino weather station with temperature, humidity, pressure date and time. ● Controlling a server motor using arduino. ● Problem-solving sessions and challenges to reinforce critical thinking and application skills. ● Utilization of online resources and visualization tools to enhance learning. ● Collaborative group work and discussions to encourage deeper understanding.	
References/ Readings:	Main Reading: 1. ArshdeepBagha, Vijay Madisetti(2015), "*Internet of Things: A Hands-On Approach*", 1st Edition, Orient Blackswan Private Limited 2. Adrian Mcewen, Hakim Cassimally(2013) , "*Designing The Internet of Things*", 1st Edition,Wiley Additional Reading: 3. Hanes David,SalguieroGonzalo, Patrick Grossetete, Rob Barton, Jerome Henry(2017), "*IoT Fundamentals: Networking Technologies, Protocols and Use Cases for the Internet of Things*", 1st Edition , Cisco Press 4. Sudip Misra, AnandarupMukherjee ,Arijit Roy(2022), "*Introduction to IoT*", 1st Edition, Cambridge University Press.	
Course Outcomes:	**On completion of the course, students will be able to:** 1. Explain the requirements and components of an IOT system. 2. Understand the role of IoT in various domains of Industry and identify sensor technologies 3. Develop different IOT Projects using the cloud technology, Arduino, NodeMcu, Raspberry pi Boards such as Nodered or similar. 4. Deploy an IoT application and connect to the cloud.	

Name of the Programme : B.Sc. Computer Science
Course Code : CSC-406
Title of the Course : Introduction to Parallel Computing
Number of Credits : 4T
Effective from AY : 2024-25 (Revised version of 2023-24)

Pre-requisites for the Course:	Basic understanding of computer architecture and programming languages.	
Course Objectives:	1. To understand the fundamental concepts of parallel computing and its architectures. 2. To use parallel programming models such as OpenMP, MPI, or CUDA to develop parallel algorithms. 3. To design and implement parallel algorithms for common computational problems. 4. To apply parallel computing concepts to solve real-world problems.	
Content:		**No. of Hours**
	Unit-1: Introduction to Parallel Computing and Parallel Architectures: • Overview of parallel computing • Motivation and significance • A brief history of parallel computing • Flynn's taxonomy of computer architectures • Shared memory vs. distributed memory systems • SIMD and MIMD architectures • Multiprocessors and multicomputers • Interconnection networks	**15**
	Unit-2: Parallel Programming Models and Algorithms: • Message Passing Interface (MPI) • OpenMP • CUDA and GPGPU programming • Parallel programming languages (e.g., Cilk, Chapel) • Parallel algorithm design principles • Sorting algorithms • Matrix multiplication and other basic parallel algorithms • Load balancing techniques	**15**
	Unit-3: Parallel Applications • Parallel scientific computing • Parallel database systems • Parallel machine learning • Parallel rendering and graphics **Performance Optimization and Evaluation** • Performance metrics in parallel computing • Parallel performance models • Strategies for performance optimization • Benchmarking and profiling tools	**15**
	Unit-4: Emerging Trends in Parallel Computing:	**15**

	• Cloud computing and parallelism • Quantum computing • Parallel computing in edge devices • Future directions and challenges in parallel computing	
Pedagogy:	**Suggested strategies for use to accelerate the attainment of the various course outcomes.** 1. The lecture method need not be only a traditional lecture method, but alternative effective teaching methods could be adopted to attain the outcomes. You may use 2. Video/Animation to explain various concepts. 3. Collaborative, Peer, Flipped Learning, etc. 4. Ask at least three HOT (Higher-Order Thinking) questions in the class, which promotes critical thinking. 5. Adopt Problem Based Learning (PBL), which fosters students' Analytical skills, and develops design thinking skills such as the ability to design, evaluate, generalize, and analyze information rather than simply recall it. 6. Introduce Topics in manifold representations. 7. Show the different ways to solve the same problem and encourage the students to come up with their own creative ways to solve them. 8. Discuss how every concept can be applied to the real world - and when that's possible, it helps improve the students' understanding 9. To promote self-learning, give at least one assignment (equivalent to 50% assignment weightage) where they can complete one MOOCs (certificate or equivalent) course out of lecture hour. Test their understanding through quizzes or presentations.	
References/ Readings:	**Main Reading:** 1. Ananth Grama, Anshul Gupta, George Karypis, and Vipin Kumar(2003), "*Introduction to Parallel Computing*", 2nd Edition, Addison Wesley **Additional Reading:** 1. Michael J. Quinn(2003), "*Parallel Programming in C with MPI and OpenMP*", McGraw Hill Education	
Course Outcomes:	**On completion of the course, students will be able to:** 1. Understand the basics of parallel computing and its architectures. 2. Design scalable and efficient parallel algorithms for various types of problems. 3. Use parallel computing to solve large-scale problems in various applications. 4. Analyze the performance of parallel programs.	

Name of the Programme : B.Sc. Computer Science
Course Code : CSC-407
Title of the Course : Research Methodology
Number of Credits : 4 (3T + 1P)
Effective from AY : 2024-25 (Revised version of 2023-24)

Pre-requisites for the Course:	Basic knowledge of Statistics	
Course Objectives:	1. To identify characteristics of scientific method, foundations of research, problem identification and problem formulation. 2. To understand the design concepts for qualitative, quantitative research, and concepts of measurements. 3. To apply concepts of research reporting/publishing. 4. To use statistical techniques/tools for data analysis.	
Content:		**No. of Hours**
	Unit-1: Foundations of Research: Meaning, Objectives, Motivation, Utility. Concept of theory, empiricism, deductive and inductive theory; Characteristics of scientific method, Understanding the language of research – Concept, Construct, Definition, Variable. Problem Identification & Formulation, Research Question, Investigation Question – Measurement Issues. **Research Design:** Concept and Importance in Research – Features of a good research design; Exploratory Research Design – concept, types and uses; Descriptive Research Designs – concept, types and uses; Experimental Design: Concept of Independent & Dependent variables. **Qualitative and Quantitative Research:** Concept, Approach and Application: Qualitative research & Quantitative research examples and problems.	**15**
	Unit-2: Data Collection Methods: Collection of Primary Data, Observation Method, Interview Method, Questionnaires, Schedules, Other Methods of Data Collection, Collection of Secondary Data, Case study method. **Measurement:** Concept of measurement– What is measured? Problems in measurement in research – Validity and Reliability. Levels of measurement – Nominal, Ordinal, Interval, Ratio**.** **Processing and Analysis of Data:** Processing operations, Elements/ types of analysis, Statistics in research- measures of central tendency or statistical averages, measures of dispersion, measures of asymmetry (skewness), measures of relationship, Simple regression analysis, Multiple correlation and regression, Partial correlation, Association in case of attributes.	**15**

	Unit-3: Hypothesis: Qualities of a good Hypothesis –Null Hypothesis & Alternative Hypothesis, procedure for hypothesis testing, flow diagram, Test of hypothesis, procedure for hypothesis testing, Hypothesis for means, difference between means, comparing two related samples, proportions, difference between proportions, comparing a variance to some hypothesized population variance, power of test. **Chi-square test:** χ2 test and their applications in research studies. **Analysis of variance:** Basic principles of ANOVA, ANOVA technique, setting up of analysis of variance table, one way, ANOVA, two-way ANOVA. **Research Reporting:** Scientific Writing Structure and components of Scientific Reports – types of Report – Technical Reports and Thesis – Significance – Different steps in the preparation – Layout, structure and Language of typical reports - Illustrations and tables – Bibliography, Referencing and footnotes	**15**
	Practical Work The broad area of practical problems is to be taken from the following two heads:	
	Week 1 to 8 ● Data Analysis using statistical tools: Data Preparation – Univariate analysis (frequency tables, bar charts, pie charts, percentages), ● Bivariate analysis – Cross tabulations and Chi-square test including testing hypothesis of association. Interpretation of Data and results	**16**
	Week 9 to 15 ● Paper Writing – Layout of a Research Paper, Software for paper formatting like LaTeX/MS Office. ● Explore Journals in Computer Science, Impact factor of Journals, When and where to publish? UGC Care List, Scopus Indexed, Web of Science. ● Explore ethical issues related to publishing, Plagiarism and Self-Plagiarism. ● Explore softwares for detection of Plagiarism. ● Use of Encyclopedias, Research Guides, Handbook etc., Academic ● Databases for Computer Science Discipline. ● Use of tools / techniques for Research: methods to search required information effectively, Reference Management Software like Zotero/Mendeley	**14**

Pedagogy:	**Suggested strategies for use to accelerate the attainment of the various course outcomes.** 1. The lecture method need not be only a traditional lecture method, but alternative effective teaching methods could be adopted to attain the outcomes. You may use a. Video/Animation to explain various concepts. b. Collaborative, Peer, Flipped Learning, etc. 2. Ask at least three HOT (Higher-Order Thinking) questions in the class, which promotes critical thinking. 3. Adopt Problem Based Learning (PBL), which fosters students' Analytical skills, and develops design thinking skills such as the ability to design, evaluate, generalize, and analyze information rather than simply recall it. 4. Introduce Topics in manifold representations. 5. Show the different ways to solve the same problem and encourage the students to come up with their own creative ways to solve them. 6. Discuss how every concept can be applied to the real world - and when that's possible, it helps improve the students' understanding To promote self-learning give at least one assignment (equivalent to 50% assignment weightage) where they can complete one MOOCs (certificate or equivalent) course out of lecture hour. Test their understanding through quizzes or presentations.
References/ Readings:	Main Reading: ● Kothari, C. R. (2004), "*Research Methodology*", 2ndEdition, New Age International Publishers. Additional Reading ● Jain, R. K. (2021),"*Research Methodology: Methods and Techniques",* Special Edition, Vayu Education of India. ● Panneer Selvam (2007), "*Research Methodology*", PHI Learning Pvt. Ltd.
Course Outcomes:	**On completion of the course, students will be able to:** 1. Recall the characteristics of scientific method, foundations of research, research process and design. 2. Understand the design concepts for qualitative, quantitative research, and concepts of measurements. 3. Apply concepts of research reporting/publishing. 4. Use statistical techniques to analyze data.

Name of the Programme : **B.Sc. Computer Science**
Course Code : **CSC-412**
Title of the Course : **Operations Research**
Number of Credits : **4 (3T+1P)**
Effective from AY : **2024-25 (Revised version of 2023-24)**

Pre-requisites for the Course:	Nil	
Course Objectives:	1. To equip students with the fundamental knowledge of OR concepts and modeling techniques. 2. To develop problem-solving skills through applying OR methodologies to real-world scenarios. 3. To enhance critical thinking and decision-making abilities in complex systems. 4. To familiarize students with software tools for OR analysis.	
Content:		**No. of Hours**
	Unit-1: Introduction to Operations Research • Definition, scope, and applications of OR in various disciplines. • Modeling approaches - Deterministic and stochastic models. • Introduction to linear programming - Formulation, graphical solution, and economic interpretation	**15**
	Unit-2: Linear Programming: Theory and Algorithms • Standard and canonical forms of linear programming problems. • Simplex method - Phases I and II, duality theory and its applications. • Sensitivity analysis and post-optimality analysis for efficient decision-making.	**15**
	Unit-3: Optimization Techniques and Applications • Transportation and assignment problems - Applications and solution strategies. • Network flow problems - Shortest path and minimum cost flow algorithms. • Introduction to integer programming and branch and bound algorithm. • Project management techniques - CPM and PERT for scheduling and resource allocation. • Decision making under uncertainty - Expected value, decision trees, and simulation methods.	**15**
	Practical Work ● Introduction to popular OR software packages like LINGO, Solver, and Excel Solver. ● Building and solving linear programming models with software. ● Network analysis and optimization using software tools.	

	● Simulation and sensitivity analysis for probabilistic decision-making. ● Case studies and project work applying OR techniques to real-world problems.	
	Week 1 & 2 1. Case study analysis of an OR application in real-world business or industry.	**04**
	Week 3 & 4 2. Formulate and solve simple linear programming problems graphically. 3. Apply the simplex method to solve small-scale linear programming models.	**04**
	Week 5 & 6 4. Conduct sensitivity analysis and interpret its implications for decision-making.	**04**
	Week 7, 8 & 9 5. Solve a transportation problem and analyze its cost efficiency. 6. Apply a network flow algorithm to find the shortest path or minimum cost flow in a network.	**06**
	Week 10 & 11 7. Formulate and solve an integer programming problem using the branch and bound method	**04**
	Week 12 & 13 8. Develop a CPM or PERT schedule for a project using software.	**04**
	Week 14 & 15 9. Make a decision under uncertainty using expected value and decision trees.	**04**
Pedagogy:	**Suggested strategies for use to accelerate the attainment of the various course outcomes.** 1. Interactive lectures with real-world examples and case studies. 2. Problem-solving and group discussions. 3. Assignments and quizzes to assess comprehension. 4. Individual and group projects applying OR methods to practical problems. 5. Utilization of OR software packages for modeling and analysis.	
References/ Readings:	Main Reading: 1. Hillier, Frederick S., and Gerald J. Lieberman(2018.),“*Introduction to operations research*”, McGraw-Hill Education Additional Reading: 1. Taha, HamdyA.(2017),“*Operations research: An introduction*”, Pearson. 2. Wayne L. Winston.(2019),“*Operations research: Applications and algorithms*”, Cengage Learning.	
Course Outcomes:	**On completion of the course, students will be able to:** 1. Identify and formulate optimization problems in various contexts.	

	2. Understand the solving linear programming problems using graphical and simplex methods. 3. Understand essential network optimization concepts and apply them to different situations. 4. Gain practical experience in using Operations Research software to analyze and solve decision-making problems.

Name of the Programme : B.Sc. Computer Science
Course Code : CSC-413
Title of the Course : Natural Language Processing
Number of Credits : 4 (3T + 1P)
Effective from AY : 2023-24 (Revised version of 2023-24)

Pre-requisites for the Course:	Knowledge of standard concepts in artificial intelligence, Basic familiarity with logic and probability, Adequate experience with programming, Knowledge of using Python libraries.	
Course Objectives:	1. To understand the fundamental concepts and ideas in Natural Language Processing (NLP). 2. To be familiar with natural language processing methods and tools. 3. To have an understanding both the algorithms available for processing linguistic information and the underlying computational properties of natural languages. 4. To Apply NLP techniques to real-world problems and datasets, and gain hands-on experience in implementing and evaluating NLP models.	
Content:		**No. of Hours**
	Unit I **Introduction to NLP** ● What is NLP?, NLP vs. Computational Linguistics. ● Levels of Linguistic Representation, Morphology, Lexical Analysis, Syntax, Semantics, Pragmatics and Discourse. **Introduction to Machine Learning and Deep Learning** ● The evaluation of NLP applications **NLP Applications** ● Machine Translation, Question Answering and Information Retrieval, Chatbots, and Dialogue Systems, Automatic Speech Recognition and Text-to-Speech	**15**
	Unit II **NLP Algorithms** ● Regular Expressions, Text Normalization, Edit Distance, N-gram Language Models, ● Naive Bayes and Sentiment Classification, Logistic Regression, Vector Semantics and Embeddings, Neural Networks and Neural Language Models, Sequence Labelling for Parts of Speech and Named Entities, RNNs and Transformers and Pretrained Language Models, Fine-tuning and Masked Language Models, Prompting and Instruct Tuning.	**15**
	Unit III **Annotating Linguistic Structure** ● Context-Free Grammar and Constituency Parsing, Dependency Parsing, Logical Representations of Sentence Meaning, ● Computational Semantics and Semantic Parsing, Relation and Event Extraction, Time and Temporal Reasoning,	**15**

	Word Senses and WordNet, Semantic Role Labelling and Argument Structure, Lexicons for Sentiment, Affect, and Connotation, Coreference Resolution, Discourse Coherence, Phonetics	
	PRACTICAL WORK The concepts learned in the units from I to IV are required to be implemented practically. The broad area of practical problems is mentioned/ suggested below.	
	Week 1-4 ● NLTK, Python 3, and the Jupyter Notebook similar IDE, Introduction to Keras, or the Natural Language Toolkit in Python for basic text processing tasks. ● Perform tokenization, stemming, and lemmatization on a given text dataset. Handle common text preprocessing tasks, such as removing stop words, punctuation, and special characters. ● Train a basic language model (e.g., n-gram model) and generate text based on the learned language model.	**08**
	Week 5-9 ● Train word embeddings using Word2Vec or GloVe on a small corpus. Utilize pre-trained word embeddings and explore semantic relationships between words. ● APIs for Social Media Web Scraping, Implement a text classification task (e.g., sentiment analysis) using a machine learning algorithm (e.g., Naive Bayes, SVM) and evaluate its performance. ● Build a simple named entity recognition model using a pre-trained model or a custom model on a labelled dataset.	**10**
	Week 10-15 ● Implement a part-of-speech tagging system using a rule-based or machine-learning approach. ● Build a basic information retrieval system using techniques like TF-IDF and evaluate its effectiveness on a dataset. ● Fine-tune a pre-trained BERT model on a specific NLP task, such as text classification or named entity recognition. ● Create a text generation model using recurrent neural networks (RNNs) or transformers and generate coherent text based on a given prompt.	**12**
Pedagogy	**Suggested strategies to use to accelerate the attainment of the various course outcomes.** 1. The lecture method need not be only a traditional lecture method, but alternative effective teaching methods could be adopted to attain the outcomes. You may use a. Video/Animation to explain various concepts.	

	b. Collaborative, Peer, Flipped Learning etc. 2. Ask at least three HOT (Higher-order Thinking) questions in the class, which promotes critical thinking. 3. Adopt Problem-Based Learning (PBL), which fosters students' Analytical skills, and develops design thinking skills such as the ability to design, evaluate, generalize, and analyze information rather than simply recall it. 4. Introduce Topics in manifold representations. 5. Show the different ways to solve the same problem and encourage the students to come up with creative ways to solve them. 6. Discuss how every concept can be applied to the real world - and when that's possible, it helps improve the students' understanding 7. To promote self-learning give at least one assignment (equivalent to 50% assignment weightage) where they can complete atleast one MOOCs (certificate or equivalent) course out of lecture hour. Test their understanding through quizzes or presentations. 8. One assignment in the form of a mini-project collecting data and using analytic tools may be given to the students.
References / Readings	Main Reading: 1. Jurafsky, Dan and Martin, James, (2008),"*Speech and Language Processing*", Second Edition, Prentice Hall. 2. Allen, J. (1995), "*Natural language understanding*",Benjamin-Cummings Publishing Co., Inc.. Additional Reading: 1. Bird, S., Klein, E., &Loper, E. (2009), "*Natural language processing with Python: analyzing text with the natural language toolkit.*", O'Reilly Media, Inc.". 2. Eisenstein, J. (2019), "*Introduction to natural language processing",*MIT press. 3. McEnery, T. (2019) "*Corpus linguistics",* Edinburgh University Press.
Course Outcomes	**On completion of the course, students will be able to:** 1. Define fundamental concepts in NLP, including tokenization, stemming, lemmatization, and syntactic and semantic analysis. 2. Interpret and compare representing and encoding language using various techniques such as bag-of-words, TF-IDF, and word embeddings. 3. Use the necessary tricks for making their models work on practical problems. 4. Connect NLP techniques to real-world problems and datasets, demonstrating the ability to choose appropriate methods and evaluate model performance.

Name of the Programme : B.Sc. Computer Science
Course Code : CSC - 414
Title of the Course : Introduction to Quantum Computing
Number of Credits : 4 (3T + 1P)
Effective from AY : 2024-25

Pre-requisites for the Course:	Basic Knowledge of Boolean Algebra, Data Structures, Computational Complexity, and Algorithm Analysis	
Course Objectives:	1. To introduce students to the fast-growing field of quantum computing 2. To create an understanding of the differences between quantum bits and classical bits 3. To familiarize with the basic quantum logical operations and algorithms 4. To provide an initial overview of quantum computing, emphasizing the shift in paradigm from classical computing and introducing fundamental quantum algorithms. 5. To equip students with future-proof skills, enable them to tackle complex problems, enhance critical thinking abilities, and promote interdisciplinary learning.	
Content:		**No. of hours**
	Unit I **Introduction to Quantum Computing** One Quantum Bit ● Superposition- superposition, complex numbers ● Measurement- measurement in Z-basis, normalization, measurement on other basis, consecutive measurements ● Bloch Sphere Mapping- global and relative phases, Bloch sphere ● Physical qubits ● Quantum Gates- linear maps, classical reversible gates, common one-qubit quantum gates, General one-qubit gates ● Quantum Circuits **Linear Algebra** ● Quantum States- Column and row vectors ● Inner Products- Inner products, Orthonormality, Projection, Measurement, Change of basis ● Quantum Gates- Gates as matrices, Common one-qubit gates as matrices, sequential quantum gates, Circuit identities, Unitarity, Reversibility ● Outer Products- Outer products, Completeness relation	**15**
	Unit II **Multiple Quantum Bits**	**15**

	● States and Measurement- Tensor product, Kronecker product, Measuring individual qubits, sequential single-qubit measurements ● Entanglement- Product states, Entangled states ● Quantum Gates- One-qubit quantum gates, Two-qubit quantum gates, Toffoli gate ● No-cloning theorem ● Quantum Adders- Classical adders, Converting classical adder to quantum gate, Quantum setup, Quantum sum, Quantum carry, Quantum ripple-carry adder, Circuit complexity, Adding in Superposition ● Universal Quantum Gates- Definition, Components, Examples, Solovay-Kitaev theorem ● Quantum Error Correction- Decoherence, Bit-flip code, Phase-flip code, Shor code	
	Unit III **Entanglement and Quantum Protocol** ● Measurements- Product states, Maximally entangled states, Partially entangled states ● Bell Inequalities- ERP paradox and local hidden variables, Bell inequalities and the CHSH inequality, Quantum processor experiment, No-signaling principle ● Monogamy and Entanglement- Classical correlations, Quantum entanglement ● Superdense Coding- The problem, Classical solution, Quantum solution ● Quantum Teleportation- The problem, Classical solution, Quantum Solution ● Quantum Key Distribution- Encryption, Classical solution: public key cryptography, Quantum solution: BB84	**15**
	Unit IV **PRACTICAL WORK**	
	Week 1-3 ● Parity- The problem, Classical solution, Quantum solution: Deutsch's Algorithm. ● Constant vs Balanced Functions- The problem, Classical solution, Quantum solution: Deutsch-Jozsa Algorithm	**06**
	Week 4-6 ● Secret Dot Product String- The problem, Classical solution, Quantum solution: Bernstein-Vazirani Algorithm, Recursive problem ● Secret XOR Mask: The problem, Classical solution, Quantum solution: Simon's Algorithm	**06**

	Week 7- 9 ● Brute-Force Searching: The problem, Classical solution, Quantum solution: Grover’s Algorithm ● Discrete Fourier Transform(DFT)- An Application, Classical solution(DFT) and Quantum solution(QFT)	**06**
	Week 10-15 ● Eigenvalue Estimation- The problem, Classical solution, Quantum solution ● Period of Modular Exponentiation- The problem, Classical solution, Quantum solution ● Factoring- The problem, Classical solution, Quantum solution (Shor’s Algorithm)	**12**
References	**A) Main Reading** 1. Nielsen, M. A., & Chuang, I. L. (2010). *Quantum computation and quantum information*. Cambridge university press. 2. Bernhardt, C. (2019). *Quantum computing for everyone*. Mit Press. 3. Hidary, J. D., &Hidary, J. D. (2019). *Quantum computing: an applied approach* (Vol. 1). Cham: Springer. **B) Additional Reading:** 1. Nielsen, M. A. (2005). Cluster-state quantum computation. 2. Sutor, R. S. (2019). *Dancing with Qubits: How quantum computing works and how it can change the world*. Packt Publishing Ltd.	
Course Outcomes	**On completion of the course, students will be able to:** 1. Recall the basic concepts and characteristics of classical and quantum computing systems 2. Understand the characteristics of classical & quantum computing systems and quantum algorithms. 3. Describe systems with qubits. 4. Perform basic quantum computing operations and quantum Fourier transform.	