DEPARTMENT OF COMPUTER SCIENCE AND APPLICATION

FYUGP OF BACHELOR of COMPUTER APPLICATION (Based on NEP)

COURSE OUTCOME (COs)

SEMESTER	COURSE NAME	COURSE TYPE	CREDIT	COURSE OUTCOME
I	Computer Fundamentals	Compulsory	4(3+1)	Upon completion of this course, students will be knowledgeable about the fundamentals of Computer Systems and Software. (a) Obtain a comprehensive understanding of the basic components and operation of computer systems. (b) Gain insight into the principles underlying software development and its role in computer systems. (c) Familiarize themselves with the essential concepts of computer hardware and software interaction. (d) Develop proficiency in identifying and explaining the functions of various hardware and software components within a computer system. (e) Acquire the foundational knowledge necessary for further exploration and specialization in the field of computer science and engineering.
I	Introduction to C Programming	Compulsory	4(3+1)	At the conclusion of the course, students will possess the following abilities: (a) Comprehend the fundamental concepts of C programming, including data types and operators. (b) Demonstrate proficiency in writing C programs that incorporate conditions, loops, and functions. (c) Apply knowledge to work with arrays, strings, and perform basic file operations in C. (d) Adapt and modify code effectively to achieve desired functionality.
I	Mathematics II	Compulsory	4	Upon successful completion of this course, students will achieve the following: (a) Grasp the concepts of set, relation, and function from a Computer Science perspective. (b) Acquire the ability to perceive a table/database as an n-ary relation. (c) Gain insight into matrices and establish connections with arrays utilized in programming.

				(d) Develop a comprehensive understanding of determinants and their application in solving simultaneous equations. (e) Familiarize themselves with statistical and probabilistic measures commonly employed in computational software/packages.
II	Data Structure and Algorithm Using C	Compulsory	4(3+1)	Upon completion of the course, students will be proficient in the following areas: (a) Comprehend and effectively utilize fundamental data structures and algorithms, including arrays, linked lists, stacks, queues, trees, and sorting and searching algorithms, utilizing the C programming language. (b) Conduct analysis on the time and space complexities of diverse algorithms, enabling them to select the most suitable algorithm for a given problem. (c) Demonstrate the ability to develop efficient algorithms to tackle various computational challenges by leveraging the data structures and algorithms discussed throughout the course.
II	Digital Logic Fundamentals	Compulsory	4	Upon completion of this course, students will possess a firm understanding of the foundational concepts of digital logic. This comprehension will serve as the cornerstone for their comprehension of computer architecture and organization. (a) Acquire a solid grasp of the fundamental principles underlying digital logic. (b) Develop a strong foundation that facilitates the comprehension of computer architecture and organization. (c) Gain proficiency in applying digital logic concepts to practical scenarios encountered in computer systems. (d) Demonstrate the ability to analyze and solve problems related to digital logic, setting the stage for more advanced studies in computer science and engineering.
II	Mathematics II	Compulsory	4	Upon successful completion of this course, students will achieve the following learning outcomes: (a) Gain proficiency in understanding the basic concepts of limit, continuity, and derivatives in mathematics. (b) Acquire knowledge of graph theory and its applications in computer science, including different representations of graphs and modelling real-life problems using graphs.

				(c) Learn fundamental graph traversal algorithms to navigate and analyze graph structures efficiently. (d) Understand the principles of counting and apply them to count under various constraints, fostering problem-solving skills. (e) Comprehend Mathematical Logic from an algorithmic perspective, enabling the development of logical reasoning and problem-solving abilities.
III	Computer Organization & Architecture	Compulsory	4	Upon completion of this course, students will acquire the following competencies: (a) Comprehend the structure, function, and key characteristics of computer systems. (b) Demonstrate understanding of the design principles governing various functional units and components within computer architectures. (c) Identify and analyze elements of modern instruction sets, discerning their implications on processor design and performance. (d) Develop an understanding of the purpose and operation of each element comprising a memory hierarchy. (e) Compare and contrast different methods for computer input/output (I/O), evaluating their respective advantages and disadvantages. (f) Attain proficiency in the basics of assembly language programming, facilitating deeper comprehension of computer system operation and design principles.
III	System Software	Compulsory	4(3+1)	Upon completing this course, students will possess a comprehensive understanding of various types of system software. (a) Gain insight into the functions and purposes of different categories of system software, such as operating systems, device drivers, utilities, and firmware. (b) Develop an understanding of the roles and responsibilities of each type of system software in facilitating efficient and effective operation of computer systems. (c) Explore the interaction between system software and hardware components, grasping how they collaborate to execute tasks and manage resources. (d) Analyze the features and characteristics of diverse system software solutions, including their strengths, limitations, and suitability for different computing environments. (e) Acquire practical knowledge and skills in configuring, managing, and troubleshooting system

				software components, preparing them for roles in IT support and system administration.
III	Object Oriented Programming Using C++	Compulsory	4(3+1)	Upon successful completion of this course, students will attain the following competencies: (a) Demonstrate the ability to conceptualize real-life concepts as objects, extracting their attributes and behaviours for computational modelling. (b) Proficiently develop programs utilizing object-oriented features such as data abstraction, polymorphism, inheritance, and exception handling for efficient problem-solving and code organization. (c) Acquire knowledge of C++ streams and operators, enabling effective input/output operations and manipulation of data streams. (d) Master file handling techniques in C++, including reading from and writing to files, managing file streams, and implementing file-related operations to store and retrieve data persistently.
IV	Database Management System	Compulsory	4(3+1)	Upon successful completion of this course, students will achieve the following objectives: (a) Gain proficiency in understanding database concepts and their architectural components. (b) Describe various data models utilized for database design. (c) Demonstrate the ability to create databases using relational models and entity relationship concepts. (d) Apply normalization techniques to ensure databases adhere to various normal forms. (e) Design and implement SQL queries proficiently to manipulate and manage relational databases effectively.
IV	Operating System	Compulsory	4(3+1)	Upon completing this course, students will have a comprehensive understanding of the internal structure and utilization of various components associated with an operating system. (a) Develop insight into the internal architecture of operating systems, including kernel, file system, process management, and memory management. (b) Understand the functionalities and interactions of different components within an operating system, such as device drivers, schedulers, and system calls.

				(c) Gain knowledge of the role and importance of process management in multitasking environments, including process creation, scheduling, synchronization, and communication. (d) Explore memory management techniques, including virtual memory, paging, and segmentation, to efficiently manage system resources. (e) Learn about file system structures, access methods, and file management operations, enabling effective storage and retrieval of data. (f) Acquire proficiency in the usage of operating system tools and utilities for system monitoring, performance optimization, and troubleshooting.
IV	Automata Theory and Languages	Compulsory	4	Upon completing this course, students will achieve the following learning outcomes: (a) Develop a thorough understanding of the mathematical model of a finite state machine, encompassing both deterministic and non-deterministic versions of Finite Automata. (b) Gain proficiency in the mathematical concepts underlying languages and grammar, including regular and context-free languages. (c) Acquire knowledge of Pushdown Automata and its associated grammar and language, enabling the recognition of more complex language structures. (d) Familiarize themselves with the properties of Regular languages and Context-free languages, facilitating the analysis and manipulation of languages within computational contexts.
IV	Python Programming	Compulsory	4(3+1)	Upon completion of this course, students will have a solid understanding of the fundamentals of Python Programming and Problem Solving. (a) Grasp the foundational concepts of Python programming language, including syntax, data types, control flow, and functions. (b) Develop proficiency in problem-solving techniques, such as algorithm design, decomposition, pattern recognition, and abstraction, using Python as a tool. (c) Gain practical experience in solving a variety of problems through hands-on coding exercises and projects. (d) Learn how to utilize built-in Python libraries and modules to enhance problem-solving capabilities and streamline development tasks. (e) Understand best practices for writing clean, efficient, and maintainable Python code, including

				code documentation, error handling, and code optimization strategies. (f) Acquire the skills necessary to analyze and solve real-world problems using Python programming, preparing students for further studies or careers in software development, data science, and other related fields.
V	Software Engineering	Compulsory	4	Upon successful completion of this course, students will attain the following competencies: (a) Identify the core challenges that affect all stages of software development processes. (b) Select appropriate software development process models, methodologies, and strategies tailored to specific project requirements, and justify their selections based on project constraints and objectives. (c) Implement various software estimation metrics, including cost, effort, size, and staffing, to effectively plan and manage software development projects. (d) Describe different software design approaches and coding/testing strategies employed in software engineering principles, ensuring the development of robust and maintainable software solutions. (e) Understand the concept of software reliability and its significance, along with the ability to calculate software maintenance costs, thereby ensuring the sustainability and longevity of software products.
V	Web Technologies	Compulsory	4(3+1)	Upon completion of the course, students will achieve the following objectives: (a) Grasp the fundamental concepts of web applications and web services, including their architecture and functionalities. (b) Design well-structured web pages proficiently using HTML and CSS, ensuring proper layout and styling. (c) Develop the skills to incorporate interactive elements and dynamic content into web pages using basic JavaScript programming. (d) Cultivate a foundational understanding of server-side scripting by learning to implement basic functionalities using PHP, enabling dynamic server-side processing in web applications.

V	Java Programming	Compulsory	4(3+1)	Upon completing this course, students will become familiar with the core concepts of Java programming and classes of the Swing package. (a) Gain proficiency in Java programming fundamentals, including syntax, data types, control structures, and object-oriented principles. (b) Acquire knowledge of the Swing package classes, enabling the development of graphical user interfaces (GUIs) in Java applications. (c) Understand how to utilize Swing components such as buttons, labels, text fields, and panels to create interactive GUIs. (d) Learn techniques for event handling and listener implementation to respond to user interactions within Swing-based GUI applications. (e) Develop the ability to design and build graphical interfaces for Java applications, enhancing their usability and functionality.
V	Computer Networks	Compulsory	4(3+1)	Upon completing this course, students will achieve the following learning objectives: (a) Gain understanding of the general principles of data communication, including transmission methods, protocols, and network architectures. (b) Learn how computer networks are structured and organized using a layered approach, such as the OSI (Open Systems Interconnection) or TCP/IP (Transmission Control Protocol/Internet Protocol) model. (c) Understand the principles of signal transmission and how data is transferred between nodes in a network. (d) Learn about the process of packet delivery in the Internet, including packet switching and routing. (e) Gain knowledge of routing protocols and how they facilitate efficient data transmission in computer networks. (f) Understand the functions of the transport layer, including segmentation, error detection, and flow control mechanisms. (g) Learn about the functions of the application layer, including data formatting, encryption, and application-specific protocols.
VI	Information Security and Cyber Laws	Elective I	4	Upon completion of the course, students will achieve the following learning outcomes: (a) Develop a foundational understanding of cyber security principles, including concepts related to confidentiality, integrity, and availability of data.

				(b) Gain insight into cryptographic techniques used to secure data transmission and storage, including encryption, decryption, hashing, and digital signatures. (c) Learn about common system attacks, such as malware, phishing, and denial-of-service attacks, and understand the methods used to detect and mitigate them. (d) Explore defensive strategies and security measures to protect against system vulnerabilities and potential cyber threats, including access control, intrusion detection systems, and security policies. (e) Acquire practical knowledge through hands-on exercises and case studies, allowing for the application of cyber security concepts in real-world scenarios.
VI	Artificial Intelligence	Elective II	4(3+1)	Upon completion of this course, students will have acquired knowledge of the fundamentals of artificial intelligence (AI) and be capable of applying basic principles in problem-solving scenarios. (a) Understand the core concepts and principles of artificial intelligence, including problem-solving, inference, perception, knowledge representation, and learning. (b) Identify scenarios where artificial intelligence techniques are applicable, recognizing problem domains suitable for AI solutions. (c) Develop the ability to apply basic AI principles to devise solutions for diverse problem-solving tasks, leveraging techniques such as search algorithms, logical reasoning, pattern recognition, and machine learning. (d) Demonstrate proficiency in implementing and evaluating AI algorithms and models using programming languages and frameworks commonly used in the field. (e) Apply critical thinking and creativity to design and optimize AI solutions that address real-world challenges across various domains, including healthcare, finance, and autonomous systems.
VI	Graph Theory	Elective III	4	Upon completion of this course, students will possess an understanding of graph theoretic concepts, problems, and associated algorithmic solutions. (a) Gain comprehension of fundamental graph theory concepts, including vertices, edges, paths, cycles, and connectivity. (b) Explore various graph problems such as shortest path, minimum spanning tree, graph colouring, and network flow.

				(c) Learn algorithmic solutions for graph problems, including Dijkstra's algorithm, Prim's algorithm, Ford-Fulkerson algorithm, and graph traversal algorithms like Depth-First Search (DFS) and Breadth-First Search (BFS). (d) Develop the ability to analyze and solve graph problems using algorithmic techniques, fostering critical thinking and problem-solving skills. (e) Apply graph theoretic concepts and algorithmic solutions to real-world scenarios, such as network optimization, route planning, and social network analysis.
VI	Project		4	Projects in computer science provide students with a holistic learning experience that prepares them for careers in technology, research, academia, and beyond. They not only acquire technical expertise but also develop essential soft skills and a growth mindset necessary for success in the rapidly evolving field of computer science.

PROGRAM OUTCOME (POs)

1) Students can demonstrate a strong understanding of core computer science principles, including algorithms, data structures, and computer architecture.

2) Students will possess proficient programming skills across various languages and paradigms, enabling them to design, implement, and debug complex software systems.

3) Students will be adept at applying mathematical and theoretical concepts to solve computational problems and analyze algorithms for efficiency and correctness.

4) Students can demonstrate proficiency in utilizing contemporary tools, technologies, and methodologies prevalent in the field of computer science, including but not limited to software development frameworks, version control systems, and agile methodologies.

5) Students will exhibit effective communication and teamwork skills, enabling them to collaborate with multidisciplinary teams, articulate technical concepts to diverse audiences, and contribute effectively to project planning and execution.

6) Students will possess critical thinking and problem-solving skills, allowing them to adapt to evolving technologies, identify innovative solutions to complex problems, and make informed decisions in the design and development of computer systems and applications.

7) Students can demonstrate an understanding of ethical, legal, and social issues related to computing, and apply ethical principles in their professional practice, including considerations for privacy, security, and societal impact of technology.

8) Students can engage in lifelong learning and professional development, staying abreast of emerging trends and advancements in computer science, and actively contributing to the advancement of the field through research, innovation, and continuous learning.