

DEPARTMENT OF COMPUTER SCIENCE

Program Specific Outcomes (PSO)

PSO 1: Discipline knowledge:

- **Outcome:** Acquiring knowledge on basics of Computer Science and ability to apply to design principles in the development of solutions for problems of varying complexity

PSO 2: Problem Solving:

- **Outcome:** Improved reasoning with strong mathematical ability to Identify, formulate and analyze problems related to computer science and exhibiting a sound knowledge on data structures and algorithms.

PSO 3: Design and Development of Solutions:

- **Outcome:** Ability to design and development of algorithmic solutions to real world problems.

PSO 4: Programming a computer:

- **Outcome:** Exhibiting strong skills required to program a computer for various issues and problems of day-to-day scientific applications.

PSO 5: Application Systems Knowledge:

- **Outcome:** Possessing a minimum knowledge to practice existing computer application software.

PSO 6: Communication:

- **Outcome:** Must have a reasonably good communication knowledge both in oral and writing.

PSO 7: Ethics on Profession, Environment and Society:

- **Outcome:** Exhibiting professional ethics to maintain the integrity in a working environment and have concern on societal impacts due to computer-based solutions for problems.

PSO 8: Lifelong Learning:

- **Outcome:** Should become an independent learner. So, learn to learn ability.

Department of Computer Science

Course Outcomes

❖ B. Sc. Semester-I (DSC-I) PROGRAMMING IN 'C'

CO1: Write simple algorithms for arithmetic and logical problems.

CO2: Write the C code for a given problem

CO3: Perform input and output operations using programs in C

CO4: Write programs that perform operations on arrays, strings, structures, unions, functions and file handling.

❖ B. Sc. Semester-I (DSC-II) COMPUTER FUNDAMENTALS

CO1: Confidently operate computers to carry out computational tasks

CO2: Understand working of Hardware and Software and the importance of operating systems

CO3: Understand number systems, peripheral devices, networking, multimedia and internet concepts

CO4: Understand various network topologies, protocols, and networking devices, and understand their role in modern communication systems.

❖ B. Sc. Semester-I (OE-I) COMPUTER FUNDAMENTALS

CO1: Confidently operate computers to carry out computational tasks

CO2: Understand working of Hardware and Software and the importance of operating systems

CO3: Understand number systems, peripheral devices, networking, multimedia and internet concepts

CO4: Understand various network topologies, protocols, and networking devices, and understand their role in modern communication systems.

❖ B. Sc. Semester-I (OE-II) OFFICE AUTOMATION

CO1: understand functionality of Operating Systems and its applications.

CO2: Working with the user interface.

CO3: prepare documents, letters and do necessary formatting of the document.

CO4: Worksheet creation, inserting and editing data in cells.

CO5: Opening/saving a presentation and printing of slides and handouts.

❖ **B. Sc. Semester-II (DSC-III) OBJECT ORIENTED PROGRAMMING USING 'C++'**

CO1: Realize the need and features of OOP and idealize how C++ differs from C.

CO2: Infer knowledge on various types of overloading.

CO3: Choose suitable inheritance while proposing solution for the given problem.

CO4: Handle pointers and effective memory management.

CO5: Illustrate application of pointers in virtual functions.

❖ **B. Sc. Semester-II (DSC-IV) OPERATING SYSTEMS**

CO1: Describe the various OS functionalities, structures and layers.

CO2: Usage of system calls related to OS management and interpreting different stages of various process states.

CO3: Design CPU scheduling algorithms to meet and validate the scheduling criteria.

CO4: Apply and explore the communication between inter process and synchronization techniques.

CO5: Implement memory placement strategies, replacement algorithms related to main memory and virtual memory techniques.

CO6: Differentiate the file systems; file allocation, access techniques along with virtualization concepts and designing of OS with protection and security enabled capabilities.

❖ **B. Sc. Semester-II (OE-I) PROGRAMMING IN 'C'**

CO1: Write simple algorithms for arithmetic and logical problems.

CO2: Write the C code for a given problem

CO3: Perform input and output operations using programs in C

CO4: Write programs that perform operations on arrays, strings, structures, unions , functions and file handling.

❖ **B. Sc. Semester-II (OE-II) WEB TECHNOLOGIES**

CO1: Differentiate web protocols and web architecture.

CO2: Apply HTML and DHTML effectively to create websites.

❖ **B. Sc. Semester-III (Paper-I) DATA STRUCTURES**

CO1: Able to write the algorithms and implementing the algorithms based on Single linked List and Double Linked list.

CO2: Able to write algorithms and implemented based on Stack and also understood the concept and Application of stack

CO3: Able to write the algorithms and implementing the algorithms based on Queue and sorting Techniques.

CO4: Able to understand the concept of Binary tree and Graphs and Traversing methods of Trees and Graphs.

❖ **B. Sc. Semester-III (Paper-II) OPERATING SYSTEM**

CO1: Able to understand structure of OS, Process management, Schedulers and different Scheduling algorithms.

CO2: Able to understand the concept related to Deadlock and Starvation.

CO3: Able to understand the concept related to memory management, partition Description table, Paging and Segmentation.

CO4: Able to understand the concept of I/O management, File management and Protection mechanism

❖ **B. Sc. Semester-IV (Paper-I) JAVA PROGRAMMING**

CO1: Students will be able to:

1. Understand timeline, features of Java programming language.
2. Understand the concept of variables and their role in storing and manipulating data in Java.
3. Differentiate between different data types in Java, such as primitive types (int, double, Boolean, etc.) and reference types (classes, arrays, etc.).

CO2: Students will be able to:

1. Understand the concept of classes and objects in object-oriented programming (OOP) and their significance in Java.
2. Implement access specifiers to enforce encapsulation and data hiding principles in class design.
3. Develop well-structured, efficient, and scalable Java programs that effectively utilize method overloading to enhance code flexibility and reusability.
4. Understand the structure and organization of the Java Class Library, including the core packages (java.lang, java.util, etc.) and their respective functionalities.
5. Understand the concepts of decision making and conditional statements in Java.
6. Develop well-structured, efficient, and maintainable Java programs that effectively utilize arrays for data storage and manipulation.
7. Understand the concept of inheritance in object-oriented programming and its significance in code reuse and hierarchy establishment.
8. Develop well-structured, efficient, and scalable Java programs that effectively utilize inheritance for code reuse and maintainable design.
9. Develop well-structured, efficient, and scalable Java programs that effectively utilize interfaces for abstraction, code reusability, and maintainable design.

CO3: Students will be able to:

1. Understand the package naming conventions and guidelines for creating and naming packages.
2. Implement basic applet functionality, including rendering graphics, handling user input, and responding to events.
3. Develop well-structured, efficient, and scalable Java programs that effectively utilize threads for concurrent programming and multitasking.
4. Understand the concept of exceptions and errors in Java and their role in handling exceptional and error conditions.

CO4: Students will be able to:

1. Understand the event-driven programming model and utilize event listeners and handlers to respond to user actions and events.
2. Design and create GUI components using AWT, such as buttons, labels, text fields, checkboxes, radio buttons, and list boxes.
3. Implement event handling for user actions, such as button clicks, menu selections, and keyboard events.
4. Develop well-structured, efficient, and user-friendly GUI applications in Java using AWT that meet specific requirements and usability standards.

❖ **B. Sc. Semester-IV (Paper-II) LINUX OPERATING SYSTEM**

CO1: Understand the basic commands of Linux operating system.

CO2: Create file systems and directories and managing hardware.

CO3: Working on files (Sharing files with others, granting and revoking file access). Managing the user's account. Understand the processes background and fore ground by process and signals system calls.

CO4: Create shared memory segments, pipes, message queues and can exercise inter processcommunication. Working with different graphical user interfaces (KDE & GNOME)

❖ **B. Sc. Semester-V (Paper-I) VISUAL BASIC PROGRAMMING**

CO1: Able to program in VB using controls and able to understand to handle data types, loops and control structures

CO2: Able to handle homogeneous data and able to apply code reusability with procedures, functions and modules

CO3: Able to design interface using Menus and able to apply DAO to handle database

CO4: Apply ADO to handle database and able to handle errors in program

❖ **B. Sc. Semester-V (Paper-II) DATABASE MANAGEMENT SYSTEM**

CO1: Concept related to DBMS, Comparative differences with traditional file system and Nonprocedural concept and different Data models

CO2: Able to construct Entity Relationship diagram, understanding the concept of strong and Weak Entity sets.

CO3: Able to perform different operations on Database and concept related to Aggregate functions.

CO4: Ability to perform Normalized the database using different normal forms.

❖ **B. Sc. Semester-VI (Paper-I) COMPILER CONSTRUCTION**

CO1: Able to understand the concept of Compilers and Translators and different phases of Compiler.

CO2: Able to understand the definitions of programming language's structure operations and storage management.

CO3: Able to understand the role of Lexical Analyzer, syntax Analyzer, Context free Grammar, Ambiguous Grammar.

CO4: Able to understand the role of Lexical Analyzer, syntax Analyzer, Context free Grammar, Ambiguous Grammar.

❖ **B. Sc. Semester-VI (Paper-II) SQL and PL/SQL**

CO1: Able to understand creating table, constraints, different Data types, functions, operations, different Database Sublanguages.

CO2: Able to understand creating views, PL/SQL programming Data types, Iterative and conditional statements and problem based on PL/SQL.

CO3: Able to understand Exceptions, writing cursors and types creating procedures and Examples on procedures.

CO4: Able to understand the concept of Functions, Purity levels in functions, Triggers, Types of Triggers, enabling disabling Triggers and problems based on Triggers.
