



Government City College (A)
Hyderabad-500002
(Affiliated to Osmania University)
Accredited with B⁺⁺ Grade by NAAC



Computer Science COURSE OUTCOMES

Course-C Programming

CO1	Remember: Students will be able to recall and identify key syntax and basic programming concepts in C. Demonstrate memory management skills in C, including variables, arrays, and pointers
CO2	Understand: Students will be able to explain the principles of procedural programming using C. Interpret and understand C code, including control structures and loops.
CO3	Apply: Students will be able to apply their knowledge by implementing solutions to simple programming problems using C. Developing and debugging C programs that involve decision-making and looping structures.
CO4	Analyze: Upon completion of the course, students will be capable of analyzing and identifying errors in C code through debugging techniques. Evaluating the efficiency of algorithms and code segments.
CO5	Create: By the end of the C Programming course, students will be able to develop and implement larger C programs that solve real-world problems. Design modular and reusable code using functions and libraries. Innovate and propose improvements to existing C code for enhanced functionality.

Course C++ Programming

CO1	Remember: Students will be able to identify and explain key C++ language constructs, such as variables, data types, loops, and conditional statements. Students will demonstrate the ability to recall and apply basic programming concepts, including input/output operations and function calls.
CO2	Understand: Students will analyze and interpret C++ code snippets, explaining the purpose and functionality of each segment. Demonstrate comprehension of control flow structures, such as if statements, loops, and switch statements, by predicting program outcomes.
CO3	Apply: Students will design and implement small-scale programs that solve real-world problems, incorporating concepts like functions, arrays, and structures. Apply object-oriented programming principles to design and implement classes and objects in C++.
CO4	Analyze: Students will analyze and debug existing C++ code, identifying and fixing logical errors, syntax issues, and potential improvements. Critically assess the efficiency of algorithms and data structures used in C++ programs, suggesting optimizations where applicable.
CO5	Create: Students will be able to design and implement larger-scale C++ projects that integrate multiple concepts, such as file handling, dynamic memory allocation, and advanced data structures. Synthesize knowledge gained throughout the course to propose and implement innovative solutions to programming challenges.

Course- Data Structures

CO1	Remember: Students will be able to recognize and name essential data structures such as arrays, linked lists, stacks, queues, trees, and graphs. They will demonstrate memory recall by describing the basic properties and use cases of each data structure.
CO2	Understand: Students will explain the core concepts behind data structures, such as time and space complexity, and how they impact algorithmic efficiency. They will be able to compare and contrast the characteristics of different data structures to make informed decisions on their usage.
CO3	Apply: Students will be proficient in coding solutions using data structures to solve real-world problems. They will apply their knowledge to design and implement algorithms that leverage specific data structures for optimal performance.
CO4	Analyze: Students will be capable of analyzing and predicting the time and space complexity of algorithms involving different data structures. They will assess trade-offs between different data structures and make informed decisions based on the requirements of a given problem.
CO5	Create: Students will be able to design novel data structures and algorithms to address complex computational problems. They will demonstrate creativity in optimizing existing data structures or creating new ones to meet specific performance requirements.

Course- Database Management System

CO1	Remember: Recall fundamental concepts and terminology related to database management systems. Identify and define key terms such as tables, fields, primary keys, foreign keys, normalization, and indexing in a relational database.
CO2	Understand: Students will be able to comprehend the principles and theories that govern the design and functioning of database management systems. Explain the relational model and its components, illustrating the relationships between tables and the importance of data normalization.
CO3	Apply: Students will be able to apply database design principles to create a well-structured and normalized relational database. Design a database schema for a given scenario, including entity-relationship diagrams, tables, and appropriate constraints.
CO4	Analyze: Students will be able to evaluate the efficiency and effectiveness of different database management strategies and optimization techniques. Compare and contrast various indexing methods (e.g., B-tree, hash) and analyze their impact on query performance in different scenarios.
CO5	Create: Students will be able to develop and implement database solutions to address real-world problems, considering both functional and performance requirements.

Course- Java Programming

CO1	Remember: Students will be able to recall and explain fundamental Java programming concepts, such as variables, data types, control structures, and object-oriented principles.
CO2	Understand: Students will be able to demonstrate a deep understanding of Java syntax and be able to interpret and explain code written by others. They will also be able to articulate the logic behind their own code.
CO3	Apply: Students will be able to apply Java programming skills to solve real-world problems. They will create programs that demonstrate their ability to design and implement solutions using appropriate algorithms and data structures.
CO4	Analyze: Students will be able to analyze complex Java code to identify errors, inefficiencies, and potential improvements. They will also evaluate different approaches to solving a given problem, considering factors such as performance and code maintainability.
CO5	Create: Students will be able to synthesize their knowledge by designing and implementing larger-scale Java projects. This may involve integrating multiple concepts and techniques learned throughout the course to create cohesive and efficient software solutions.

Course- Web Programming

CO1	Remember: Students will be able to recall and identify fundamental concepts and syntax of web programming languages such as HTML, CSS, and JavaScript.
CO2	Understand: Students will demonstrate comprehension of web development principles, including the DOM (Document Object Model) and responsive design.
CO3	Apply: Students will be able to apply their knowledge of web programming languages to create interactive and dynamic web pages incorporating user input, data retrieval, and dynamic content using HTML, CSS, and JavaScript.
CO4	Analyze: Students will analyze and debug code, identifying and fixing common errors and issues in web applications.
CO5	Create: Students will be able to critically evaluate and compare different web development frameworks and tools, making informed decisions based on project requirements.