

DEPARTMENT OF COMPUTER SCIENCE

B. Sc. Computer Science

Programme Outcomes

After successfully completing BSc Comp. Sci. Programme students will be able to:

PO1: By the end of B.Sc. Computer Science programme, students will be able to formulate algorithms for effective programming, involve in data analysis, and create standalone and web-based applications using Visual Programming languages.

PO2: Work effectively both individually and as member of team.

PO3: Gain the knowledge about software engineering fundamentals, including software analysis and design, evaluation and testing, and software engineering processes.

PO4: Describe mathematics fundamentals, including discrete structures, statistics and calculus.

PO5: Illustrate the concepts of Microprocessors and microcontrollers.

PO6: Able to design and develop fun and practical IoT devices.

Programme Specific Outcomes

PSO1: Apply fundamental principles and methods of Computer Science to a wide range of applications.

PSO2: Impart an understanding of the basics of our discipline

PSO3: An ability to design, implements, and evaluate a computer based system to solve computational problems to meet desired needs of real life.

PSO4: An ability to function effectively in teams and individually to accomplish a common goal.

PSO5: An ability to use new computing technologies, tools and skills for attaining professional excellence.

Course Outcomes

F.Y.B.Sc. (Computer Science)

Course (CS-101): Problem solving using Computer and C-Programming

After successfully completing this course, students will be able to:

- CO1: Explore algorithmic approaches to problem solving.
- CO2: Understand 'C' Programming language tokens.
- CO3: Explain input, output, conditional and iterative statements in C programming.
- CO4: Interpret C programs using array and functions.
- CO5: Explain string and pointer concepts of C programming.
- CO6: Implement various data types and control structures for problem solving
- CO7: Develop modular programs using control structures and arrays in 'C'.

Course (CS-102): File Organization and Fundamental of Databases

After successfully completing this course, students will be able to:

- CO1: Define the suitable Heap, Sorted, Indexed, Hashed File Organization technique;
- CO2: Solve real world problems using appropriate set, function, and relational models.
- CO3: Design E-R Model for given requirements and convert the same into database tables.
- CO4: Illustrate the basics of Structured Query Language and construct queries using SQL;
- CO5: Populate and query a database using SQL DML/DDL commands.
- CO4: Use SQL.

Course (CS-103): Computer Science Practical Paper- I

After successfully completing this course, students will be able to:

- CO1: List the basic UNIX general purpose commands, data types and Operators in C-Language.
- CO2: Devise pseudocodes and flowchart for computational problems.
- CO3: Write, debug and execute simple programs in 'C'.
- CO4: Create database tables in postgre SQL.
- CO5: Write and execute simple, nested queries.

Course (CS-201): Advanced 'C' Programming

After successfully completing this course, students will be able to:

- CO1: Understand advanced concepts of programming using the 'C' language.
- CO2: Understand code organization with complex data types and structures.
- CO3: Develop modular programs using control structures, pointers, arrays, strings and structures
- CO4: Design and develop solutions to real world problems using C.

Course (CS-201): Relational Database Management Systems

After successfully completing this course, students will be able to:

CO1: Understand fundamental concepts of RDBMS (PL/PgSQL)

CO2: Understand database management operations

CO3: Learn the basic issues of transaction processing and concurrency control

CO4: Able to design E-R Model for given requirements and convert the same into database tables.

CO5: Use database techniques such as SQL & PL/SQL.

CO6: Explain transaction Management in relational database System.

CO7: Use advanced database Programming concepts

Course (CS-203): Computer Science Practical Paper

After successfully completing this course, students will be able to:

CO1: Write, debug and execute programs using advanced features in 'C'.

CO2: Interpret the integrity constraints on a database;

CO3: Use of SQL DDL/DML commands to perform query on a database.

CO4: Able to perform advanced database operations.

S. Y. B. Sc. (Computer Science)

Course (CS-231): Data Structures and Algorithms – I

After successfully completing this course, students will be able to:

CO1: To use well-organized data structures in solving various problems.

CO2: To differentiate the usage of various structures in problem solution.

CO3: Implementing algorithms to solve problems using appropriate data structures

Course (CS-232): Software Engineering

After successfully completing this course, students will be able to:

CO1: Compare and chose a process model for a software project development.

CO2: Identify requirements analyze and prepare models.

CO3: Prepare the SRS, Design document, Project plan of a given software system.

Course (CS-223): Practical course on CS 231 (Data Structures and Algorithms I) and CS 232 (Software Engineering)

After successfully completing this course, students will be able to:

CO1: Implement searching and sorting algorithms for basic data structures programs

CO2: Implement Singly Linked and doubly Linked Lists
CO3: Implementation of stack to perform various operations
CO4: Implementation of Linear, Circular and Priority Queue to perform various operations
CO5: Prepare the SRS, Design document, Project plan of a given software system.

Course CS-241: DATA STRUCTURES AND ALGORITHMS-II

After successfully completing this course, students will be able to:

CO1: Implementation of different data structures efficiently
CO2: Usage of well-organized data structures to handle large amount of data
CO3: Usage of appropriate data structures for problem solving

Course CS-242: Computer Networks-I

After successfully completing this course, students will be able to:

CO1: Have a good understanding of the OSI and TCP/IP Reference Models and in particular have a good knowledge of Layers.
CO2: Understand the working of various protocols.
CO3: Analyze the requirements for a given organizational structure and select the most appropriate networking architecture and technologies

Course CS-243: Practical course on CS 241(Data Structures and Algorithms II) and CS 242 (Computer Networks I)

After successfully completing this course, students will be able to:

CO1: Implementation of appropriate data structures for problem solving
CO2: Implementation of different methods of organizing large amount of data
CO3: Implementation of static and linked hash table with Linear Probing and chaining.
CO4: Independently understand basic computer network technology.
CO5: Identify the different types of network topologies and protocols.
CO6: Implement a simple LAN with hubs, bridges and switches.

T. Y. B. Sc. (Computer Science)

Course (CS-331): System Programming and Operating Systems-I

After successfully completing this course, students will be able to:

CO1: Describe the different types of Programming Environment, purpose of editors and types of Editors.
CO2: Discuss the data structures of Assembler
CO3: Explain Data Structures of Macro pre-processor
CO4: Illustrate the concepts of Interpreter, Compiler Linker and Loader
CO5: Explain types of Debugger and demonstrate how to debug the program
CO6: Describe the Operating system as system software and types of system calls.

Course (CS-341): System Programming and Operating Systems-II

After successfully completing this course, students will be able to:

- CO1: Discuss the operating system structure and issues related to process management.
- CO2: contrast the different CPU scheduling algorithms
- CO3: Explain the multithreading models and synchronization techniques.
- CO4: Interpret the different strategies of deadlocks.
- CO5: Describe the different issues related to memory management
- CO6: Discuss file access methods, directory structure and file allocation methods.

Course (CS-347): System Programming and Operating Systems Practical

After successfully completing this course, students will be able to:

- CO1: Perform the different Line editor commands
- CO2: Illustrate the SMACO program
- CO3: Demonstrate the concepts of Assembler and Macro
- CO4: Use concept DFA to check particular Language accepts or not
- CO5: Illustrate different the shell commands
- CO6: Perform the different CPU scheduling algorithms
- CO7: Demonstrate deadlock avoidance algorithm to find the Safe Sequence.
- CO8: Use the different page replacement algorithms to find page fault.

Course (CS-331): Theoretical Computer Science and Compiler Construction – I

After successfully completing this course, students will be able to:

- CO1: Explain how to generate formal language & regular expressions.
- CO2: Express concepts of finite automata
- CO3: Describe knowledge of regular languages
- CO4: Discuss context free languages & different types of grammar
- CO5: Explain concepts of pushdown automata
- CO6: Summarize concepts of Turing machine.

Course (CS-342): Theoretical Computer Science and Compiler Construction – II

After successfully completing this course, students will be able to:

- CO1: Explain phases of compiler & Lexical analyzer.
- CO2: Illustrate types of parsers
- CO3: Express use of YACC tool.
- CO4: Describe Syntax Directed Definitions & its application.
- CO5: Discuss memory allocation in block structure languages, code optimization & code generation.

Course (CS-333): Computer networks –I

After successfully completing this course, students will be able to:

- CO1: Define goals and importance of computer networks.
- CO2: Demonstrate network infrastructure according to various topologies and network type

(LAN, WAN and MAN)

CO3: Describe OSI reference model and TCP/IP model.

CO4: Explain various hardware and software used in network design.

CO5: Discuss various terminologies and protocols used in physical layer.

CO6: Discuss various design issues and various protocols used in data link layer.

Course (CS-343): Computer networks –II

After successfully completing this course, students will be able to:

CO1: Define Wired LAN (Standard Ethernet MAC Layer)

CO2: Discuss standards of IEEE 802.11 architecture and Bluetooth architecture used in Wireless LAN;

CO3: Explain IPV4 protocol used in the network layer

CO4: Explain protocols- ARP, UDP and TCP

CO5: Discuss WWW architecture, E-mail and HTTP

CO6: Illustrate Cryptography and firewall used in network security.

Course (CS-334): Internet Programming- I

After successfully completing this course, students will be able to:

CO1: Interpret fundamental concept of web techniques

CO2: Discuss concept of user define function & predefine functions of strings.

CO3: Explain types of array & predefine function of array.

CO4: Illustrate object oriented concepts in PHP script.

CO5: Describe file & directory handling operation & predefine function of file & directory.

CO6: Explain the database enable web pages.

Course (CS-344): Internet Programming-II

After successfully completing this course, students will be able to:

CO1: Explain content used in web technology.

CO2: Discuss PHP framework & email handling using PHP;

CO3: Explain XML programs, its advantages & different XML parser

CO4: Interpret the concept of JavaScript for web designing

CO5: Describe functioning of Ajax model.

Course (CS-348): Internet Programming, Networking Practical and Project

After successfully completing this course, students will be able to:

CO1: Illustrate a form to implement functions and predefine functions

CO2: Demonstrate the array concepts and its predefine functions

CO3: Apply the predefine functions of files and directories

CO4: Solve problems using object oriented concept

CO5: Demonstrate database enabled web pages using PostgreSQL

CO6: Apply JavaScript in web pages

CO7: Demonstrate dynamic web pages by using Ajax

CO8: Illustrate various concepts of web development in project

CO9: Demonstrate various networking commands in Unix.

Course (CS-335): Programming in Java-I

After successfully completing this course, students will be able to:

- CO1: Define simple java programs using data types, final variable and arrays
- CO2: Explain classes using constructor and array of objects
- CO3: perform java programs using classes and objects
- CO4: Illustrate the concept of inheritance and interfaces
- CO5: implements exception handling techniques in java programs;
- CO6: Demonstrate GUI using Swing and AWT (Abstract Window Toolkit) methods
- CO7: Interpret basic applet using java.

Course (CS-345): Programming in Java-II

After successfully completing this course, students will be able to:

- CO1: Explain programs using java collection API as well as java Standard Library
- CO2: Discuss GUI Applications with JDBC (Java Database Connectivity)
- CO3: Define concept of Servlet
- CO4: Interpret simple Java Server Pages (JSP) Application
- CO5: Describe multithreading using java
- CO6: Demonstrate simple application for client and server communication
- CO7: Illustrate java concept for solving simple business problem

Course (CS-348): Programming in Java Practical

After successfully completing this course, students will be able to:

- CO1: Define simple classes using IDE – Eclipse
- CO2: Explain examples of classes using array of objects and packages
- CO3: implement inheritance and interfaces in java
- CO4: Solve problems using exception handling mechanism in java
- CO5: perform Input/output operations using console and files
- CO6: Apply AWT and Swing to create GUI in java
- CO7: Execute queries on tables using JDBC (Java Database Connectivity)
- CO8: Define and execute simple servlet program
- CO9: Illustrate the JSP (Java Server Pages) programs
- CO10: Demonstrate multithreading using Java.

Course (CS-336): Object oriented software engineering

After successfully completing this course, students will be able to:

- CO1: Recall fundamental principles underlying Object-Oriented software design like class, Object, Instance Polymorphism and inheritance.
- CO2: Give the original examples of basic and advance structural modelling things like class, Objects.
- CO3: Explain basic behavioural things like use case diagram, interaction diagram and state chart Diagram.

- CO4: Explain methods of object oriented analysis and object oriented designing
- CO5: Use architectural modelling like component and deployment diagram
- CO6: Define object oriented testing strategies.

Course (CS-346): Computer Graphics

After successfully completing this course, students will be able to:

- CO1: Define computer graphics, components of computer graphics, and Open GL
- CO2: List input and output devices, graphical user interfaces in Open GL, graphics presentation
- CO3: Explain raster scan graphics methods of line drawing algorithms, polygon filling algorithms, scan conversion,
- CO4: Describe basic transformation and window to viewport co-ordinate transformation. Setting window and viewport in OpenGL,
- CO5: Use line clipping and polygon clipping algorithms,
- CO6: Describe 3-D transformations hidden surface elimination methods.



HEAD

**DEPARTMENT OF COMPUTER SCIENCE
G.M.D. Arts, B.W. Commerce
and Science College, Sinnar**