DEPARTMENT OF COMPUTER SCIENCE

B. Sc. (Computer Science)

Programme Outcomes

After successfully completing **B. Sc. (Computer Science)** Programme students will be able to:

- PO1: Use creativity, critical thinking, and analysis and research skills to solve theoretical and real-world problems in computer science
- PO2: Work effectively both individually and as member of team.
- PO3: Discuss software development fundamentals, including programming, data structures, algorithms and complexity.
- PO4: Illustrate the concepts of systems fundamentals, including architectures and organization, operating systems, networking and communication.
- PO5: Gain the knowledge about software engineering fundamentals, including software analysis and design, evaluation and testing, and software engineering processes.
- PO6: Communicate effectively for different purposes and in different situations.
- PO7: Gain self-discipline in everyday aspects of life and work.
- PO8: Describe mathematics fundamentals, including discrete structures, statistics and calculus.
- PO9: Illustrate the concepts of Microprocessors and microcontrollers.
- PO10: Make use of fundamentals of Application, including information management and **intelligent** applications.

Programme Specific Outcomes

After successfully completing **B. Sc. (Computer Science)** Programme students will

- PSO1: Apply knowledge of computing and mathematics appropriate to the discipline
- PSO2: Develop problem-solving abilities using computer.
- PSO3: Design the application using programming languages.
- PSO4: Ability to understand the principles and development methodologies of computer systems.

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: List the flow chart and algorithm for given problem;
- CO2: Discuss the programming language tools and history of C programming;
- CO3: Define C Tokens like keywords, identifiers and operators;
- CO4: Explain input, output, conditional and iterative statements in C programming;
- CO5: Interpret C programs using array and functions;
- CO6: Explain string and pointer concepts of C programming;
- CO7: Illustrate user defined data types including structures and unions to solve the problems;
- CO8: Discuss command line arguments and files handling in C programming.

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: Express Relational, Hierarchical and Network Data Model and structure of Database Management System;
- CO3: Discuss the conceptual modelling tools like E-R diagram and Relational Data Model;
- CO4: Explain the Integrity Constraints on a database schema;
- CO5: Describe Relational Algebraic operations and construct the queries to write Relational Algebra expression;
- CO6: Illustrate the basics of Structured Query Language and construct queries using SQL;
- CO7: Explain the basics of query evaluation techniques and query optimization;
- CO8: Explain the concepts of Functional Dependency and design the database;
- CO9: Interpret the concepts of Normalization and Normal Forms used to improve database design;

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: Use the decision making statements like if, if-else, nested if and switch case in C program;
- CO3: Demonstrate while, do-while, for, nested loops of C-Program;
- CO4: Apply standard library functions in menu driven program in C- Language;
- CO5: Solve C Program using array, pointer, string and functions;
- CO6: Illustrate C-program using structure and Union;
- CO7: Discuss the concepts of file handling command line arguments in C-Programming.

Course (CS-104): Computer Science Practical Paper- II

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

- CO1: Explain the HTML pages using lists, table and hyperlinks;
- CO2: Apply advanced features like CSS and its types on web pages;
- CO3: Demonstrate simple forms layout with HTML;
- CO4: Solve a small case study to create simple application;
- CO5: Define a database schema for a given problem domain;
- CO6: Interpret the integrity constraints on a database;
- CO7: Use of SQL DDL/DML commands to perform query on a database.

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

Course (CS-211): Data Structures Using 'C'

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

- CO1: Discuss fundamental concepts of Data Structure, abstract data type, and algorithm analysis;
- CO2: Summarize different searching and sorting techniques using array;

CO3: Describe linear data structure Stack and its application;
CO4: Explain linear data structure Queue and its types (Linear Queue, Circular Queue, and Priority Queue);
CO5: Summarize different types of Linked List (singly linked list, doubly linked list, linear and circular linked list);
CO6: Discuss non-linear data structure Tree using operations like searching, insertion, deletion, and traversing mechanism;
CO7: Explain non-linear data structure Graph using operations like traversing mechanism;

Course (CS-221): Object Oriented Concepts using C++

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

CO1: depict the applications and need of Object Oriented Programming language;
CO2: Discuss basic concepts of C++ programming language;
CO3: Describe the concepts of classes, objects, member function, constructors and destructor;
CO4: Explain the need of operator overloading, inheritance, polymorphism, and virtual functions;
CO5: Explain managing input- output, and file in C++;
CO6: Explain exceptions handling and templates in C++.

Course (CS-223): Data structures Practical and C++ Practical (Lab Course- I)

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

CO1: Use different searching and sorting methods for basic data structures programs;
CO2: Solve simple mathematical problems using data structure;
CO3: Implement various data structures viz. Stack, Queues and Linked Lists;
CO4: Implement complex data structures like trees and graphs;
CO5: Demonstrate programs by using basic object oriented concepts in C++;
CO6: Apply to overload functions and Operators in C++;
CO7: Illustrate programs by applying the object oriented concepts such as (Inheritance, Virtual Function.)
CO8: Apply of file handling in C++ programs.

Course CS-212: Relational Database Management System

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

CO1: Recall the integrity constraints on a database using RDBMS;
CO2: Explain the concepts of stored procedures, stored functions, and cursors triggers in PL/PGSQL programming language;
CO3: Explain the concepts of transactions processing, concurrency control and recovery;
CO4: Interpret the concurrency control techniques;
CO5: Describe the concepts of crash recovery;
CO6: Discuss the data security methods for database protection;
CO7: Summarize the knowledge about client server architecture.

Course CS-222: Software Engineering

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

CO1: Explain the characteristics of system, elements of system, and types of system;
CO2: Discuss software, its application domain and, software engineering principles;
CO3: Describe the activities of system development life cycle;
CO4: Illustrate different software process models used in practice;
CO5: Summarize the requirement engineering tasks;
CO6: Discuss the methods used to build structure analysis model.

Course CS-224: Database Practical's and Mini Project using Software Engineering Techniques (Lab Course- II)

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

CO1: Solve the simple and nested queries using PL/PGSQL;
CO2: Demonstrate stored functions, cursors, triggers and views;
CO3: Illustrate queries using loops and conditional statements;
CO4: Use error and exception handling methods;
CO5: Describe the software engineering processes such as gathering data and functional requirements in the software project;
CO6: Apply feasibility study techniques for the software project;
CO7: Discuss the existing system, and explain the proposed system;
CO8: Determine the entities, attributes and draw E-R diagram.

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 applications;
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.

M. Sc. (Computer Science)

Programme Outcomes :

After successfully completing M**. Sc. (Computer Science)** Programme students will be able to:

PO1: Use creativity, critical thinking, and analysis and research skills to solve theoretical and real-world problems in computer science

PO2: Work effectively both individually and as member of team to design and implement solution to computational problems.

PO3: Discuss various algorithms and analysis of algorithms with its complexity

PO4: Illustrate the concepts of networking and communication, and distributed computation, and security.

PO5: Gain the knowledge about programming languages like PROLOG, LISP, Dot Net.

PO6: Communicate effectively for different purposes and in different situations.

PO7: Gain self-discipline in everyday aspects of life and work.

PO8: Make use of Application fundamentals, including information management and intelligent applications.

PO9: Apply current technical concepts and practices in the core information Technologies

PO10: Apply the knowledge about software engineering fundamentals, including software analysis and design, evaluation and testing, and software engineering processes.

Programme Specific Outcomes

After successfully completing M**. Sc. (Computer Science)** Programme students will

PSO1: Design and develop the software applications using programming languages

PSO2: Apply knowledge of computing and mathematics appropriate to the discipline

PSO3: Use knowledge of computer science in various computer domains

PSO4: Pursue higher studies and research in the computer Science discipline.

Course Outcomes

M. Sc. (Computer Science) (Part – I)

Course CS-101: Principles of Programming Languages

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

CO1: Summarize & Categorize Programming Language Spectrum;

CO2: Discuss Functional & Logic-based language;

CO3: Interpret the concepts of Object lifetime, Storage Management, Scope rules, binding rules and solve the problems using scope & binding rules;

CO4: Summarize the various control flow alternatives;

CO5: Illustrate the various data types & its implementations;

CO6: Describe the fundamental concept of Subprogram & implementation of subprograms and solve the problems using parameter passing methods;

CO7: Explain data abstraction & Object Oriented Features;

CO8: Discuss Multiprocessor architecture and categories of concurrencies.

Course CS-102: Advanced Networking

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

- CO1: Recall the basic concepts of Networking;
- CO2: Discuss the Internet Layer. Routing and Transport Layer protocol;
- CO3: Describe the multimedia concepts and it's protocols;
- CO4: Explain the fundamental concepts of Cryptography and different algorithms;
- CO5: Interpret the Symmetric, Asymmetric key cryptography and digital certificate;
- CO6: Discuss internet security protocol.

Course CS-103: Distributed Database Concepts

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

- CO1: Describe the principles of distributed databases and what distinguishes them from centralized databases;
- CO2: Discuss the different architectures used in implementing distributed databases;
- CO3: Demonstrate how horizontal and vertical fragmentation work in dividing relations;
- CO4: Illustrate the allocation algorithms in distributing relation fragments among sites;
- CO5: Explain the techniques used in securing distributed database entities, such as views and authorization controls;
- CO6: Discuss the three known distributed query optimization algorithms namely, Distributed INGRES, R*, and SDD-1;
- CO7: Explain principle behind distributed concurrency control algorithms, especially the two-phase locking (2PL) and timestamp-based algorithms;
- CO8: Describe the deadlock problem that is associated with distributed transactions and the techniques used in detecting, preventing, and avoiding them.

Course CS-104: Design & Analysis of Algorithms

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

- CO1: Describe how to analyze the algorithms & estimate their best case, worst case & average case behaviour;
- CO2: Discuss divide and conquer design strategy and solve the problems using this strategy;
- CO3: Explain Greedy method and solve the problems using this method;
- CO4: Discuss Dynamic programming and solve the problems using this strategy;
- CO5: Interpret the concepts of Decrease and conquer strategy and solve the problems using this method;
- CO6: Discuss Backtracking and solve the problems using this method;
- CO7: Explain Branch and Bound Technique and solve the problems using this method;
- CO8: Describe Transform and conquer strategy and solve the problems using this method;
- CO9: Analyze & classify the different class of problems.

Course (CS-105): UNIX Network Programming

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

- CO1: Explain basic socket address structures and different conversion functions;
- CO2: Discuss UNIX systems programming, signals, forking, and unp libraries;
- CO3: Illustrate the simple TCP Client/Server applications using Posix C sockets library;

CO4: Contrast I/O multiplexing models Client-Server Functions;
CO5: Describe different Socket Options;
CO6: Illustrate the simple UDP Client/Server applications using Posix C sockets library;
CO7: Explain elementary name and address conversion functions;

Course (CS-201): Digital Image Processing

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

CO1: Discuss basic image processing techniques for solving real problems;
CO2: Explain the fundamental concepts of a digital image processing system;
CO3: Describe various types of images, intensity transformations and spatial filtering;
CO4: Classify the techniques for image enhancement and image restoration;
CO5: Explain morphological image processing algorithms;
CO6: Interpret image segmentation and representation techniques.

Course (CS- 202): Advanced Operating Systems

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

CO1: Recall the functions of Operating System;
CO2: Explain the system calls related to files and directory;
CO3: Describe the process environment and it's relationship;
CO4: Interpret the different memory management schemes;
CO5: Discuss the system calls related to signals;
CO6: Explain the thread management in windows operating system.

Course (CS-203): Data mining & Data Warehousing

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

CO1: Interpret the basic concepts of data mining and its applications;
CO2: Discuss data warehousing and its architecture;
CO3: Explain various data mining techniques and solve the problem using those techniques;
CO4: Describe classification and prediction algorithms and solve the problem using those techniques;
CO5: Summarize the various accuracy measures;
CO6: Discuss clustering algorithms and solve the problem using those techniques;
CO7: Explain the advanced techniques such as text mining and web mining.

Course: CS-204 Project / CS-304 Project

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

CO1: Describe the phases of Software development project life cycle;
CO2: Apply the various project management tools and techniques;
CO3: Implement software systems that meet specified design & performance requirements;
CO4: Use Team Management to effectively design & implement the project;
CO5: Demonstrate effective project execution & Control techniques that results in successful project.

Course (CS-205): Programming with Dot Net

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

CO1: Discuss Dot net framework;

CO2: Explain various c# language features;
CO3: Demonstrate window programming using different controls like Common Controls, Container Controls, Menus and Toolbars;
CO4: Illustrate Data Access with file system data and ADO.net;
CO5: Discuss web Programming, Web Services and dot net assemblies;
CO6: Describe Web Request, Web Response, TCP Client, TCP Listener and GDI+ libraries;
CO7: Discuss ASP.net Server Controls, Control Structures & Functions;
CO8: Demonstrate Event Driven Programming and Post Back;
CO9: Illustrate state management and session events;

Course (CS-206): Artificial Intelligence

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

CO1: Describe problems, problem spaces & search for AI problems;
CO2: Interpret heuristic search techniques;
CO3: Express knowledge representation & reasoning using predicate logic;
CO4: Explain knowledge representation using semantic network;
CO5: Illustrate game playing & learning concepts.

Course: (CS-301) Software Metrics and Project Management

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

CO1: Define the development phases of project life cycle;
CO2: Describe the project integration management and their knowledge areas throughout the project life cycle;
CO3: Explain the knowledge of various tools and techniques used for elements of Project management like Scope, Time, Cost, Quality, Human Resources, Communication, Risk and Procurement;
CO4: Use of metrics in software project;
CO5: Explain Software Reliability characteristics, tools and methods used for Software Reliability;
CO6: Discuss the software process assessment models like Capability Maturity Model, TSP, PSP.

Course (CS-302): Mobile Computing

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

CO1: Interpret the basic concepts of mobile computing & its applications;
CO2: Discuss general concepts of multiplexing, modulation and spread spectrum;
CO3: Describe specialized MAC for wireless environment and Compare SDMA, TDMA, FDMA & CDMA;
CO4: Explain Mobile IP & different mobile routing protocols;
CO5: Summarize the problems of Traditional TCP & Discuss Classical TCP improvements;
CO6: Explain system architecture of GSM and various value added services;
CO7: Discuss UMTS, UTRAN system architecture;
CO8: Describe Wireless application protocol architecture;
CO9: Discuss Introduction to Android Operating System, its architecture and features.

Course (CS-303): Soft Computing

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

- CO1: Illustrate the concept of Fuzzy sets, knowledge representation using fuzzy rules, Fuzzy Inference System, Fuzzy Logic and various operations on it;
- CO2: Discuss the fuzzy system simulation and classification;
- CO3: solve the problems using fuzzy arithmetic;
- CO4: Describe Artificial Neural Network and applications of it;
- CO5: Explain Genetic Algorithms and differentiate Genetic algorithms from Traditional methods.

Course (CS-305): Web Services

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

- CO1: Discuss standards and bind architecture of Web services;
- CO2: Demonstrate web service by using Java Language;
- CO3: Explain SOAP (Simple object access protocol);
- CO4: Describe WSDL (Web Services Description Language);
- CO5: Discuss various service models used in cloud computing;
- CO6: Illustrate case study based on cloud computing.

Course (CS-307): Functional Programming

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

- CO1: Interpret the concept of Python languages;
- CO2: Illustrate the concept of string, list, tuple, set and dictionary in python;
- CO3: discus the concept of files and directories in python;
- CO4: Explain the concept of object oriented concept in python;
- CO5: Describe concept of functional programming and varieties of functional programming language;
- CO6: Explain semantics of function language using precise formal specification;
- CO7: Describe different reduct.

Course: CS-401 Industrial Training Project

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

- CO1: Select comprehensive learning platform students can enhance their employ ability skills and become job ready along with real corporate exposure;
- CO2: Apply the theory knowledge to get hands-on experience in the field of computer science;
- CO3: Appreciate the ethical basis of professional practice in relevant industry;
- CO4: Describe with all the latest changes in technological world;
- CO5: Interpret options in career plans and goals.