DEPARTMENT OF COMPUTER SCIENCE

B. Sc. (Computer Science)

Programme Outcomes

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

PO1: To build the necessary skill set and analytical abilities for developing computer based solutions for real life problems.

PO2: To train students in professional skills related to Software Industry.

PO3: To prepare necessary knowledge base for research and development in Computer science.

PO4: To help students' build-up a successful career in Computer Science and to produce entrepreneurs who can innovate and develop software products.

PO5: To provide knowledge of technological and practical aspects of electronics.

PO6: To familiarize with current and recent technological developments.

PO7: To enrich knowledge through activities such as industrial visits, seminars, projects.

PO8: To train students in skills related to computer industry and market.

PO9: To create foundation for research and development in Electronics/ Computer Science.

PO10: To develop analytical abilities towards real world problems

PO11: To help students to build-up a progressive and successful career.

PO12: To develop problem solving abilities using a computer.

PO13: A student should be able to recall basic facts about mathematics and should be able to display knowledge of conventions such as notations, terminology and recognize basic geometrical figures and graphical displays, state important facts resulting from their studies.

PO14: A student should get a relational understanding of mathematical concepts and concerned structures, and should be able to follow the patterns involved, mathematical reasoning.

PO15: A student should get adequate exposure to global and local concerns that explore them many aspects of Mathematical Sciences.

PO16: A student be able to apply their skills and knowledge ,that is,translate information presented verbally into mathematical form, select and use appropriate mathematical formulae or techniques in order to process the information and draw the relevant conclusion.

PO17: A student should be made aware of history of mathematics and hence of its past, present and future role as part of our culture.

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) CBCS: 2019-2020

SEM I:

Course (CS-101): Problem Solving Using Computer and 'C' Programming – I

Course Type: Core Credit

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

CO1: To introduce the foundations of computing, programming and problem- solving using computers.

CO2: To develop the ability to analyze a problem and devise an algorithm to solve it

CO3: To formulate algorithms, pseudo codes and flowcharts for arithmetic and logical problems operators;

CO4: To understand structured programming approach.

CO5: To develop the basic concepts and terminology of programming in general.

CO6: To implement algorithms in the 'C' language.

CO7: To test, debug and execute programs problems.

CO8: Explore algorithmic approaches to problem solving

CO9: Develop modular programs using control structures and arrays in 'C'.

Course (CS-102): Database Management Systems

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

CO1: To understand the fundamental concepts of database.;

CO2: To understand user requirements and frame it in data model.

CO3: To understand creations, manipulation and querying of data in databases

CO4: Solve real world problems using appropriate set, function, and relational models

CO5: Design E-R Model for given requirements and convert the same into database tables.

CO6: Use SQL

Course (CS-103): Practical course on Problem Solving using Computer and 'C' programming and Database Management Systems

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

CO1: To understand the program development life cycle.
CO2: Solve simple computational problems using modular design and basic features of the 'C' language.
CO3: Understand basic database management operations.
CO4: Design E-R Model for given requirements and convert the same into database tables
CO5: Devise pseudocodes and flowchart for computational problems.
CO6: Write, debug and execute simple programs in 'C'.
CO7: Create database tables in postgreSQL.
CO8: Write and execute simple, nested queries.

Course (ELC-111): Semiconductor Devices and Basic Electronic Systems

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

CO1: To study various types of semiconductor devices
CO2: To study elementary electronic circuits and systems

Course (ELC-112): Principles of Digital Electronics

CO1: To get familiar with concepts of digital electronics
CO2: To learn number systems and their representation
CO3: To understand basic logic gates, Boolean algebra and K-maps
CO4: To study arithmetic circuits, combinational circuits and sequential circuits

SEM II:

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

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

CO1: To study advanced concepts of programming using the 'C' language.
CO2: To understand code organization with complex data types and structures.
CO3: To work with files
CO4: Develop modular programs using control structures, pointers, arrays, strings and structures.
CO6: Design and develop solutions to real world problems using C

Course (CS-202): Relational Database Management Systems

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

CO1: To teach fundamental concepts of RDBMS (PL/PgSQL).
CO2: To teach database management operations
CO3: Be familiar with the basic issues of transaction processing and concurrency control **CO4:** To teach data security and its importance
CO5: Design E-R Model for given requirements and convert the same into database tables.
CO6: Use database techniques such as SQL & PL/SQL.
CO7: Explain transaction Management in relational database System.
CO8: Use advanced database Programming concepts

Course (CS-203): Practical Course on Advanced 'C' Programming and Relational Database Management Systems

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

CO1: To solve real world computational problems
CO2: To perform operations on relational database management systems.
CO3: Write, debug and execute programs using advanced features in 'C'.
CO4: To use SQL & PL/SQL.
CO5: To perform advanced database operations

Course (ELC 121): Instrumentation Systems

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

CO1: To study Instrumentation System
CO2 To study various blocks of Instrumentation System
CO3: To study Smart Instrumentation System

Course (ELC 122): Basics of Computer Organization

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

CO1: To get familiar digital sequential circuits
CO2 To study Basic computer Organization
CO3: To study Memory architecture
CO4: To use SQL & PL/SQL.
CO5: To perform advanced database operations

Course (CSST 111) :Descriptive Statistics

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

CO1: To tabulate and make frequency distribution of the given data.

CO2: To use various graphical and diagrammatic techniques and interpret.

CO3: To compute various measures of central tendency, dispersion, Skewness and kurtosis.

Course (CSST 112) :Mathematical Statistics

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

CO1: To fit the Binomial and Poisson distributions.

CO2: To compute the measures of attributes.

CO3: The process of collection of data, its condensation and representation for real life

CO4: To study free statistical softwares and use them for data analysis in project.

Course(MTC-111): Matrix Algebra and Course(MTC 112): Discrete Mathematics

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

CO1: A students should be able to work with graphs and identify certain parameters and properties of the given graphs.

CO2: A students should be able to perform certain algorithms, justify why these algorithms work, and give some estimates of the running times of these algorithms.

CO3: A students should be able to solve basic exercises of the type: given a graph with properties X, prove that the graph also has property Y.

CO4: A students should develop an appreciation for the literature on the subject and be able to read and present results from the literature.

CO5: A students should be able to write cohesive and comprehensive solutions to exercises and be able to defend their arguments.

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

SEM III

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

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

CO1: To learn the systematic way of solving problem;
CO2: To understand the different methods of organizing large amount of data
CO3: To efficiently implement the different data structures
CO4: To efficiently implement the different data structures;
CO5: To apply linear data structures.
CO6: To use well-organized data structures in solving various problems.
CO7: To differentiate the usage of various structures in problem solution.
CO8: To differentiate the usage of various structures in problem solution.

Course(CS 232): Software Engineering

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

CO1: To get knowledge and understanding of software engineering discipline.
CO2: To learn analysis and design principles for software project development.
CO3: Compare and chose a process model for a software project development.
CO4: Identify requirements analyze and prepare models.
CO5: Prepare the SRS, Design document, Project plan of a given software system

Course(ELC 231):Microcontroller Architecture & Programming

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

CO1: To study the basics of 8051microcontroller
CO2: To study the Programming of8051microcontroller
CO3: To study the interfacing techniques of 8051microcontroller
CO4: To design different application circuits using 8051microcontroller
CO5: To write programs for 8051 microcontroller
CO6: To interface I/O peripherals to 8051 microcontroller
CO7: To design small microcontroller based projects

Course(ELC 232):Digital Communication and Networking

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

CO1: Define and explain terminologies of data communication
CO2: Understand the impact and limitations of various digital modulation techniques
CO3: To acknowledge the need of spread spectrum schemes.
CO4: Identify functions of data link layer and network layer while accessing communication link
CO5: To choose appropriate and advanced techniques to build the computer network.

SEMIV

Course(CS 241): DATA STRUCTURES AND ALGORITHMS-II

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

CO1: To learn the systematic way of solving problems
CO2: To design algorithms
CO3: To understand the different methods of organizing large amount of data
CO4: To efficiently implement the non-linear data structures
CO5: Implementation of different data structures efficiently
CO6: Usage of well-organized data structures to handle large amount of data
CO7: 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(ELC-241): Paper I : Embedded System Design On completion of the course, student will be able

CO1: To understand the difference between general computing and the Embedded systems.
CO2: To know the fundamentals of embedded systems.
CO3: Understand the use of Single board Computer (Such as Raspberry Pi) for an embedded
system application.
CO4: Familiar with the programming environment to develop embedded systems and their
interfaces with peripheral devices.
CO5: To develop familiarity with tools used to develop in an embedded environment.

Course(ELC-242): Paper II: Wireless Communication and Internet of Things

Course Outcomes: Students will be able to

CO1:Know working of wireless technologies such as Mobile communication, GSM, GPRS

CO2: Become familiar with 3G and 4G Cellular Network Technologies for Data Connections.

CO3:Understand working principles of short range communication application

CO4: Get introduce to upcoming technology of Internet of Things

CO5: Explore themselves and develop new IoT based applications

Course(ELC-243): Paper III, Practical Course

Course Outcomes : On completion of the course, students will be able

1. To design and develop own smart applications using Rasberry-Pi

2. To write Python program for simple applications

3. To build own IoT based system

Course (MTC-231): Groups and Coding Theory and Course(MTC-232 :Numerical Techniques and MTC-233: Mathematics Practical: Python Programming Language-I

(i) A student should be able to recall basic facts about mathematics and should be able to display knowledge of conventions such as notations, terminology and recognize basic geometrical figures and graphical displays, state important facts resulting from their studies.

(ii) A student should get a relational understanding of mathematical concepts and concerned structures, and should be able to follow the patterns involved, mathematical reasoning.

(iii) A student should get adequate exposure to global and local concerns that explore them many aspects of Mathematical Sciences.

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.