M.Sc. COMPUTER SCIENCE

Program Objective/Outcomes

- The M.Sc Computer Science program prepares students to take up positions as system analysts, systems designers, Programmers and managers in any field related to information technology.
- The program, therefore, aims at imparting comprehensive knowledge with equal emphasis on theory and practice.
- However, the course curriculum will have enough flexibility to enable a student to undertake advance studies in computer science later on.
- Providing basic foundations of different areas of Computer Applications
- Providing specializations in the fields of Networking, Web applications, Database Management Systems.
- Cultivating the students who have keen interest in the area of Computer Science and to help them to start their career in their desired field by awarding them with the Masters' Degree when they complete.
- The broad objective of the M.Sc Computer Science programme is to prepare students for careers in software industry understanding and skills related to the use of Computers and its application.

COURSE OUTCOMES

DISCRETE MATHEMATICAL STRUCTURES

- To introduce a number of Discrete Mathematical Structures (DMS) found to be serving as tools even today in the development of theoretical computer science.
- Course focuses on of how Discrete Structures actually helped computer engineers to solve problems occurred in the development of programming languages.
- Also, course highlights the importance of discrete structures towards simulation of a problem in computer science and engineering.
- Introduction of a number of case studies involving problems of Computer Technology.
- Realization of some satisfaction of having learnt that discrete structures are indeed useful in computer science and engineering and thereby concluding that no mistake has been done in studying this course.
- Gaining of some confidence on how to deal with problems which may arrive in computer science and engineering in near future.
- Above all, students who studied this course are found to be better equipped in a relative sense as far as preparation for entrance examinations involving placement opportunities.

DATA STRUCTURES & FILE STRUCTURES

- To introduce the fundamental concept of data structures and to emphasize the importance of data structures in developing and implementing efficient algorithms.
- To develop effective software engineering practice, emphasizing such principles as decomposition, procedural abstraction
- Describe how arrays, records, linked structures, stacks, queues, trees, and graphs are represented in memory and used by algorithm

- Describe common applications for arrays, records, linked structures, stacks, queues, trees

COMPUTER ORGANIZATION AND ARCHITECTURE

- Conceptualize the basics of organizational and architectural issues of a digital computer.
- Analyze processor performance improvement using instruction level parallelism.
- Learn the function of each element of a memory hierarchy.
- Study various data transfer techniques in digital computer.
- Articulate design issues in the development of processor or other components that satisfy design requirements and objectives.
- Learn microprocessor architecture and study assembly language programming.
- Describe basic organization of computer and the architecture of 8086 microprocessor.
- Implement assembly language program for given task for 8086 microprocessor.
- Demonstrate control unit operations and conceptualize instruction level parallelism.
- Demonstrate and perform computer arithmetic operations on integer and real numbers.
- Categorize memory organization and explain the function of each element of a memory hierarchy.
- Identify and compare different methods for computer I/O mechanisms.

OBJECT ORIENTED PROGRAMMING USING C++ & JAVA

- To understand Object Oriented Programming concepts Abstraction, Inheritance, and Polymorphism.
- To understand the Basics of C++. Implement concepts of OOPS Classes & Objects, Inheritance & Polymorphism in C++ and able to write Programs in C++
- To understand the class hierarchy, characteristics of Java, inheritance and polymorphism and become familiar with the relationship between classes and objects in a Java program.
- Learn programming based on JAVA 7 and above.
- To write efficient and effective applications in Java, Java's event handling model, graphical user interface (GUI), swing component set, understand the relationship between the AWT and Swing.
- Have a better understanding of Java's event model and design, build simple Graphical User Interfaces (GUI)s, Networking, Java Database Connectivity with JDBC™, Servlets, JavaServer Pages (JSP).
- The course aims to make the students learn programming in Java. Java language elements and characteristics, including data types, operators, and control structures are discussed in order to make the students develop Java applications.
- The course also intended for students who would like to learn how to develop internet based applications, graphical user interface (GUI), and graphics in both AWT and SWING.
- Advanced Java topics discussed helps students writing programs for Java database connectivity with JDBC; Manipulating databases with JDBC; Programming for Internet, JavaServer pages.

ADVANCED COMPUTER NETWORKS

DATA & FILE STRUCTURES LAB.

- Develop programs that use arrays, records, linked structures, stacks, queues, trees
- Demonstrate different methods for traversing trees
- Compare alternative implementations of data structures with respect to performance
- Compare and contrast the benefits of dynamic and static data structures implementations
- Design the concept of recursion, give examples of its use, describe how it can be implemented using a stack
- Design and implement an appropriate hashing function for an application
- Design the computational efficiency of the principal algorithms for sorting, searching, and hashing .

COMPUTER ORGANIZATION LAB.

- Minimize the Boolean algebra and design it using logic gates.
- Analyse and design combinational circuit.
- Realise given function using combinational circuit.
- Design and develop sequential circuits
- Implement digital systems using programmable logic devices
- Translate real world problems into digital logic formulations
- Assembly Language Programming using 8086 Microprocessor

FORMAL LANGUAGES & AUTOMATA THEORY

The following can be learned upon completion of this course.

- Understand the formal languages, its properties and relation between the language classes.
- Understand the formal grammars, its properties and relation between them.
- Understand the Automata theory which recognizes the formal languages, its properties, relation between them.
- Understand the types of formal languages and types of Automata.
- Understand the construction of Formal Languages, Formal Grammars and Automata which recognize them
- Proving and disprove theorems for properties of automata theory.
- Understand the concepts of decidability

RELATIONAL DATABASE MANAGEMENT SYSTEMS

The *RDBMS Concepts* WBT is an end user course that provides an overview of the architecture, functions, and benefits of a database management system and discusses various database models. The course describes the data structure of a relational database model in detail. It extensively covers the normalization process and compares SQL with other popular relational databases. After completing this course, the student will be able to:

- Describe a database management system and trace its historical development
- Identify the advantages of the database approach over the file-based data storage system
- Describe the architecture of a DBMS
- List the functions of the database system components
- List the features of distributed and object-oriented databases
- Describe different types of logical-based data models
- List the features of the entity-relationship modeling technique
- Identify the features of hierarchical and network record-based models
- Describe a relational record-based model in terms of its data structure
- List the operators that work on relations
- List the features of the first, second, and third normal forms
- Describe the Boyce-Codd Normal Form (BCNF) and the need for denormalization
- List the features of SQL as the relational database language
- Compare different RDBMSs

ADVANCED OPERATING SYSTEMS

- Analyze the structure and basic architectural components involved in OS
- Understand the process management policies and scheduling of processes by CPU By studying the process management and scheduling.
- To understand the working of an OS as a resource manager, file system manager, process manager, memory manager and I/O manager and methods used to implement the different parts of OS
- To understand the concepts and implementation Memory management policies and virtual memory.
- Ability to Identify use and evaluate the storage management policies with respect to different storage management technologies.
- Ability to evaluate the requirement for process synchronization and coordination handled by operating system
- To study the need for special purpose operating system with the advent of new emerging technologies

Elective - I Embedded Systems

- Explain the embedded system concepts and architecture of embedded systems
- Describe the architecture of 8051 microcontroller and write embedded programs for 8051 microcontroller.
- Design the interfacing for 8051 microcontroller.
- Understand the concepts of ARM architecture.
- Demonstrate the open source RTOS and solve the design issues for the same.
- Select elements for an embedded systems tool.

ELECTIVE - II WEB TECHNOLOGIES

- To understand the Design of Fundamentals in Internet Concepts like Web Page Design
- To Learn the Design of Web Pages using HTML, CSS
- To Learn JavaScript and perform Client side validations

- To understand the basics of JDBC and ODBC and the architecture of JDBC API and its objects, different Types of JDBC drivers & their advantages and disadvantages
- To learn how to install the Servlet Engine / Web Server
- basics of Servlet and how it is better than other server extensions;
- how Servlet engine maintains the Servlet Life Cycle;
- where do we use HttpServletRequest Interface and some of its basic methods;
- where do we use HttpServletResponse Interface and some of its basic methods;
- what is session tracking;
- Different ways to achieve Session Tracking like HttpSession & persistent cookies, and different ways to achieve InterServlet communication.
- understand the need of JSP;
- understand the functioning of JSP;
- understand the relation of applets and servlets with JSP;
- know about various elements of JSP;
- explain various scripting elements of JSP;
- explain various implicit objects of JSP, and
- understand the concept of custom tags and process of creating custom tag
- understand how to connect JSP with a database;
- understand how to select, insert and delete records in database using JSP;
- understand how to include files and applets in JSP documents;
- understand how to include the output of JSP, HTML or plain text pages at the time the client requests the page;
- understand how to include JSP files at the time the main page is translated into a servlet;
- understand how to include applets that use the Java Plug-In, and
- understand how to integrate servlet and JSP.
- Understand MVC Design Pattern

ADVANCED JAVA PROGRAMMING LAB

Student learn and develop the programs in Basic concepts in Java

- Develop the Java Applications using Multi threading
- Develop the Java Applications using Files & String Manipulations
- Develop the Java Applications using Multi threading
- Develop the Java Applications to demonstrate Applet Program using Various Controls and perform Font animation
- Develop the Java Application to demonstrate Menus, sub Menus, Popup Menus, Shortcut Keys, Check Boxes and Separators
- Develop the Java Applications to demonstrate Java to DataBase Connectivity (JDBC)
- Develop the Java Applications to demonstrate Servlets , JSP, Java Beans, RMI

RELATIONAL DATABASE MANAGEMENT SYSTEMS LAB.

- To introduce to a commercial DBMS such as ORACLE.
- To learn and practice SQL commands for schema creation, data manipulation.
- To learn conceptual and physical database design based on a case study.
- To apply database design stages by studying a case study.

- The student is exposed to a commercial RDBMS environment such as ORACLE.
- The student will learn SQL commands for data definition and manipulation.
- The student understands conceptual through physical data base design.
- The student takes up a case study and applies the design steps.

DATA WARE HOUSING & MINING

This course will introduce the concepts of data warehouse and data mining, which gives a complete description about the principles, uses, architectures, applications, design and implementation of data mining and data warehousing concepts.

- Understand the functionality of the various data mining and data warehousing component
- Appreciate the strengths and limitations of various data mining and data warehousing models
- Explain the analyzing techniques of various data
- Describe different methodologies used in data mining and data ware housing.
- Compare different approaches of data ware housing and data mining with various technologies.

OBJECT ORIENTED SOFTWARE ENGINEERING

- To Analyse, Design, Develop the Software using the Object Oriented Approach
- To design and develop softwares using UML Diagrams
- To understand the nature of software development and software life cycle process models, agile software development, SCRUM and other agile practices.
- To Explain methods of capturing, specifying, visualizing and analyzing software requirements.
- To understand concepts and principles of software design and user-centric approach and principles of effective user interfaces.
- To know basics of testing and understanding concept of software quality assurance and software configuration management process
- To understand the need of project management and project management life cycle.
- To understand project scheduling concept and risk management associated with various type of projects.

NETWORK SECURITY & CRYPTOGRAPHY

- The concepts of classical encryption techniques and concepts of finite fields and number theory.
- And explore the working principles and utilities of various cryptographic algorithms including secret key cryptography, hashes and message digests, and public key algorithms
- And explore the design issues and working principles of various authentication protocols, PKI standards.
- And explore various secure communication standards including Kerberos, IPsec, and SSL/TLS and email.

- The ability to use existing cryptographic utilities to build programs for secure communication.
- The concepts of cryptographic utilities and authentication mechanisms to design secure applications

ELECTIVE - III OPERATIONS RESEARCH

This module aims

- To introduce students to use quantitive methods and techniques for effective decisions–making; model formulation and applications that are used in solving business decision problems.
- To introduce the students how to use variables for formulating complex mathematical models in management science, industrial engineering and transportation science.
- To provide the students with opportunity of using various software package for solving linear programming and integer programming models
- To introduce the students to the use of basic methodology for the solution of linear programs and integer programs.
- To introduce the students to the basic concepts of valid inequalities and how to integrate the theory to the solution methods for integer programming.
- To introduce the students to the advanced methods for large-scale transportation and assignment problems.

MOOCS - I -- INTERNET OF THINGS

- To learn the concepts of IOT.
- To identify the different technology.
- To learn different applications in IOT.
- To learn different protocols used in IOT.
- To learn the concepts of smart city development in IOT.
- To learn how to analysis the data in IOT
- Apply the concepts of IOT.
- Identify the different technology.
- Apply IOT to different applications.
- Analysis and evaluate protocols used in IOT.
- Design and develop smart city in IOT.
- Analysis and evaluate the data received through sensors in IOT.

OOSE LAB.

- To apply Object Oriented Concepts in Project Development
- To prepare Requirement Engineering, Design Such as architecture, User Interface Design, Testing, Preparations User Manuals Etc and also
- To Design of DBMS Schema Including Normalization, Forms design, Report Generation, Linking to Web DataBases Etc. Preferably on Live Projects
- Periodical Presentations and Discussions Among the Groups and their Outputs.

NETWORK PROGRAMMING & WEB PROGRAMMING LAB.

- Student is able to understand the Networking Concepts
- Student design and develop Network Programs using Java Socket Program
- Student can design and develop the programs using the following
- Identifying the Well known Ports for Protocols
- One User to one User Communication
- One to Many User Communication
- Chat Application
- FTP Protocol and FTP Server
- SMTP Protocol
- HTTP Protocol