Department of Computer Science
Program Offered: B.Sc. (H) Computer Science
Course Outcome

This course is designed for students to get exposure on all aspects of computers right from the basic fundamentals of computers to the recent and advanced courses based on programming languages. Degree in Computer Science combines theoretical study and practical projects, teaching of subject-specific skills including programming languages, hardware architecture, network programming, software engineering, web application tools and packages and database concepts. This course is useful to develop a personal portfolio of your own projects, such as those involving programming, building a website or carrying out tasks online for example, your initiative and ability in fixing bugs, improving functionality or building an app will help show your skills and interest in the subject.

After this course students can pursue post graduate courses like M.C.A., M.Sc. Computer Science, M.Sc. (Information Technology), M.I.T., M.B.A., M.Sc. (OR) and M.Sc. (AOR) etc. Various Job Opportunities after completion of this course include: Information Systems Manager, IT Consultant, Multimedia Programmer, Systems Developer, Web Designer and Web Developer.

Given below is a brief description of expected learning outcomes of the various papers taught as part of this three-year under-graduate program. The course structure is the same as for all Honours Courses in Science.

Semester I

1. **Programming Fundamentals Using C++ (C-I)**
   This Paper will help students to get knowledge of object oriented programming concepts to develop solutions to problems demonstrating usage of control structures, modularity, inheritance, I/O and other standard language constructs. This paper also includes a practical component which is designed to give the student hands on experience with the programming concepts. This paper will give an insight to students to use object oriented paradigm in program designing.

   Learning Outcome: By the end of this course, students should be able to:

- Describe the procedural and object oriented paradigm with concepts of streams, classes, functions, data and objects.
- Understand dynamic memory management techniques using pointers, constructors, destructors, etc
- Describe the concept of function overloading, operator overloading, virtual functions and polymorphism.
- Classify inheritance with the understanding of early and late binding, usage of exception handling, generic programming.
- Demonstrate the use of various OOPs concepts with the help of programs.

2. **Computer System Architecture (C-II)**

This paper imparts the knowledge of the fundamental concepts of modern Computer System Architecture i.e. the structure and behavior of the various functional modules of the computer and how their interaction amongst themselves to meet various processing needs of the user. Its main objective is to give insight on the hardware design of the system providing the basic knowledge on how different functions and modules interact to process user's instruction.

Learning Outcome: By the end of this course, students should be able to:

- understand the basics of computer hardware and how software interacts with computer hardware
- Analyze how system performance is evaluated.
- Understand how data is represented and manipulated in computers.
- Understand computer arithmetic operations and conversion between different number systems
- Understand the basics of Instruction Set Architecture.
- Simulate hardware in any simulation software depending upon required configuration by including data format, instruction format, instruction set, addressing modes, bus structure, input/output, memory, Arithmetic/Logic unit, control unit, and data, instruction and address flow
- Use Boolean algebra as related to designing computer logic, through simple combinational and sequential logic circuits

Semester –II

1. **Programming in JAVA (C-III)**

This Paper will impart students with the knowledge of the structure and model of the Java programming language. It will also enable them to demonstrate programs on advance concepts like inheritance, packages, exceptions, multithreading and applets and will help them to understand how to model real world scenario.

Learning Outcomes: By the end of this course, students should be able to:

- Use Java programming language in object-oriented paradigm
- Understand the Java architecture and use the Java Application Programmable Interfaces.
- Understand and use inheritance and polymorphism.
- Understand and use the exception handling mechanism of Java
- perform standard input-output operations
- Understand and use GUI components

2. **Discrete Structures (C-IV)**

This Paper helps the students to be familiar with constructing proofs, elementary formal logic, set algebra, recurrence relations, graphs and trees, relations and functions. It also helps in learning fundamental mathematical concepts and terminology.

Learning Outcomes: By the end of this course, students should be able to:

- Have a complete knowledge on various discrete structures available.
- Understand set notation and set theory, recognize the connection between set operations and logic, and prove results involving sets.
- Formulate and interpret statements presented in disjunctive normal form and determine their validity by applying the rules and methods of propositional calculus.
- Reformulate statements to formal logic using the rules of propositional and predicate calculus, and assess the validity of arguments.
- Formulate short proofs using the following methods: direct proof, indirect proof, proof by contradiction, and case analysis.
- Construct elementary proofs using recursion, relations, and graph theory, and identify fallacious inductive arguments.
- Apply discrete probability's axioms and laws.
- Determine when a relation is reflexive, symmetric, anti-symmetric, or transitive and explain the connection between equivalence relations and partitioning a set.
- Explain basic definitions and properties associated with graphs and trees and describe the difference between Eulerian and Hamiltonian graphs.

Semester –III

1. **Data Structures (C-V)**

   This paper teaches about efficient data storage mechanisms for easy access; design and implementation of various basic and advanced data structures. It helps students to know about various techniques for representation of the data in the real world as well as improves the logical ability. Students will be able to choose appropriate data structure as applied to specified problem definition and will be able to handle operations like searching, insertion, deletion, traversing mechanism etc. on various data structures.

   Learning Outcomes: By the end of this course, students should be able to:

- Understand the advantages of different data structures in representing different types of information.
- Understand the concept of a linked list data structure and its different implementations.
- Understand the concept of a stacks and LIFO structure.
- Understand the concept of Queues and FIFO structure.
- Understand the concept of the tree data structure and its use in different applications.
- Understand the concept of hash tables.
- Understand the concept of graphs and its applications.

2. Operating Systems (C-VI)

This course covers the functions, structures and history of operating systems and gives the understanding of design issues associated with operating systems. The objective of the course is to understand scheduling, synchronization, deadlocks, multithreading, system resources sharing among the users, file system interface and implementation, disk management, protection and security mechanisms.

Learning Outcomes: By the end of this course, students should be able to:

- Understand the role of operating systems and the interfaces they offer to application programs and to users.
- Understand the structure of OS, design and implementation issues of an operating system.
- Understand and apply fundamental concepts for process management in operating systems.
- Understand concepts for memory management.
- Understand and apply significant concepts of storage management, file systems, and input/output in operating systems.
- Understand and compare different operating systems.

3. Computer Networks (C-VII)

This course covers the terminology and concepts of the OSI reference model and the TCP-IP reference model, concepts of protocols, network interfaces, and design/performance issues in local area networks and wide area networks. The objective of the course is to make students familiar with wireless networking concepts, contemporary issues in networking technologies, tools and network programming.

Learning Outcomes: By the end of this course, students should be able to:

- Understand the fundamental concepts of data communications and networking.
- Identify the different components of a communication system.
- Solve problems in networking by referring to problems solving steps through relevant information by choosing suitable techniques and protocols.
- Understand the configuring of networking devices and understand their functionality.
- Know the strategies for securing network applications.

4. Android Applications Development Programming (SEC-1)

This Paper will help students in learning, letting students develop competence and confidence in android programming and in understanding the entire Android Application Development Cycle, as well as it would also enable the students to independently create new Android Applications.

Learning Outcomes: By the end of this course, students should be able to:

- Understand how to create and publish their own Apps for Google Android devices.

- Understand the entire project management framework.

Semester –IV

1. **Design and Analysis of Algorithms (C-VIII)**
   This paper will help the students to learn good principles of algorithm design along with the knowledge of how to analyze algorithms, estimate their worst-case and average-case behavior and become familiar with fundamental data structures and with the manner in which these data structures can best be implemented.
   Learning Outcomes: By the end of this course, students should be able to:

- Argue the correctness of algorithms using inductive proofs and loop invariants.
- Analyze worst-case, best case, average case running times of algorithms using asymptotic analysis.
- Compare the asymptotic behaviors of functions- polynomials, exponentials, and logarithmic functions.
- Describe the divide-and-conquer paradigm. Derive and solve recurrences describing the performance of divide- and-conquer algorithms.
- Explain important algorithms for sorting. Derive lower bounds on the running time of comparison-sorting algorithms.
- Describe Breadth First Search and Depth First Search graph algorithms and shortest path algorithms like Prims, Kruskal and Dijikstra.
- Describe the greedy programming paradigm.

2. **Software Engineering (C-IX)**
   This paper includes the knowledge of basic Software engineering methods and practices, and their appropriate application with a general understanding of software process models such as the waterfall and evolutionary models and understanding of the role of project management including planning, scheduling, risk management, etc.

   Learning Outcomes: By the end of this course, students should be able to:

- Understand software engineering topics and relate them to recent software developments.
- Understand software development methodologies.
- Understand different software process models and its applications.
- Understand how requirements are gathered, verified and validated.
- Understand different techniques of project management like process modeling, planning, process metrics, risk management etc.
- Understand the importance of software testing and apply different test cases of validation.

3. **Database Management Systems (C-X)**
   This Paper helps the student to understand, appreciate and effectively explain the underlying concepts of database technologies, to design and implement a database

schema for a given problem-domain. The practical sessions impart them with the knowledge of normalization nd query a database using SQL DML/DDL commands.

Learning Outcomes: By the end of this course, students should be able to:

- Describe the features of database management systems.
- Construct database models and translate between such models.
- Implement, manipulate and query relational databases.
- Understand querying from multiple tables.
- Understand different database objects.
- Manipulate database tables
- Create databases and do normalization on this database to eliminate anomalies.

4. **PHP Programming (SEC-2)**
   This Paper helps the students to get the basic knowledge of how server-side programming works on the web, its basic syntax for variable types and calculations along with the knowledge of using PHP built-in functions, custom functions and reading and writing cookies. Students will also learn the PHP programming skills needed to successfully build interactive, data-driven sites.

   Learning Outcomes: By the end of this course, students should be able to:

- Applying the W3C standard syntax and structure for XHTML documents
- Formatting Web pages using Cascading Style Sheets (CSS)
- Performing client-side scripting using JavaScript
- Learning the basic structure and syntax of the PHP scripting language
- Understand the use of open source software PHP and MySQL.
- Understand and build professional database driven interactive websites.
- Understand how to integrate PHP and MySQL with XHTML, CSS.

Semester -V

1. **Internet Technologies (C-XI)**
   This course covers the JAVA, JDBC, JAVA BEANS, JSP, and Java Script for developing programming applications and major Internet and web developments. The objective of this course to understand the concept of Object Oriented Programming through Java programming language and JavaScript and JSP scripting language have been used for the development of web application.

   Learning Outcomes: By the end of this course, students should be able to:

- Understand design of JDBC and its configuration.
- Understand the concept of object oriented programming through Java Programming language.

- Learn database connectivity, web based applications which support database connectivity and creating dynamic web pages.
- Understand the concept of scripts, creating java script programs.
- Develop server side programs using Servlets.
- Develop Java Server Pages applications using JSP Tags
- Have an in depth knowledge of Advance Java.

2. Theory of Computation (C-XII)

This paper is introduced at the undergraduate level because of the deeper insights it yields on specific topics in computer science and serves to establish essential mathematical paradigms that permeate computer science.

Learning Outcomes: By the end of this course, students should be able to:

- Understand basic properties of formal languages and formal grammers.
- Understand properties of deterministic and non-deterministic automata.
- Develop basic understanding of machine design problems.
- Understand basic properties of PDA, turing machines.
- Learn and understand functioning, capabilities and complexities of different mathematical models.

3. System Programming (DSE-1A)

To make the student to understand the process involved in a compiler, create an overall view of various types of translators, linkers, loaders, and phases of a compiler, understand what is syntax analysis, various types of parsers especially the top down approach, awareness among students the various types of bottom up parsers, understand the syntax analysis and, intermediate code generation, type checking, the role of symbol table and its organization, Code generation, machine independent code optimization and instruction scheduling.

Learning Outcomes: By the end of this course, students should be able to:

- Understand the basics of system programs like editors, compiler, assembler, linker, loader, interpreter and debugger.
- Describe the various concepts of assemblers and macro processors.
- Understand the various phases of compiler and compare its working with assembler.
- Understand how linker and loader create an executable program from an object module created by assembler and compiler.
- Know various editors and debugging techniques.

4. Microprocessor (DSE-2A)

By studying this paper, students acquire thorough knowledge of Assembly Language Programming and interfacing of the Intel family of Microprocessors. This knowledge

enables students to develop control software to control an application interface microprocessor.

Learning Outcomes: By the end of this course, students should be able to:

- Understand the taxonomy of microprocessors and knowledge of contemporary microprocessors.
- Describe the architecture, bus structure and memory organization of 8085 as well as higher order microprocessors.
- Explore techniques for interfacing I/O devices to the microprocessor 8085 including several specific standard I/O devices such as 8251 and 8255.
- Demonstrate programming using the various addressing modes and instruction set of 8085 microprocessor.
- Design structured, well commented , understandable assembly language programs to provide solutions to real world control problems

Semester –VI

1. **Artificial Intelligence (C-XIII)**

This paper helps the students to understand the following concepts:
The different types of AI agents
How various AI search algorithms (uninformed, informed, heuristic, constraint satisfaction, genetic algorithms) work
Understand the fundamentals of knowledge representation (logic-based, frame-based, semantic nets), inference and theorem proving.

Learning Outcomes: By the end of this course, students should be able to:

- Demonstrate working knowledge in Lisp in order to write simple Lisp programs.
- Understand different types of AI agents.
- Know various AI search algorithms (uninformed, informed, heuristic, constraint satisfaction, genetic algorithms).
- Understand the fundamentals of knowledge representation (logic-based, frame-based, semantic nets), inference and theorem proving.
- Demonstrate working knowledge of reasoning in the presence of incomplete and/or uncertain information.
- Ability to apply knowledge representation, reasoning, and machine learning techniques to real-world problems.

2. **Computer Graphics (C-XIV)**

This course will introduce students to all aspects of computer graphics including hardware, software and applications. Students will gain experience using a graphics application programming interface by completing several programming projects.

Learning Outcomes: By the end of this course, students should be able to:

- Understand the basics of computer graphics, different graphics systems and applications of computer graphics.
- Discuss various algorithms for scan conversion and filling of basic objects and their comparative analysis.
- Use of geometric transformations on graphics objects and their application in composite form.
- Extract scene with different clipping methods and its transformation to graphics display device.
- Explore projections and visible surface detection techniques for display of 3D scene on 2D screen.
- Render projected objects to naturalize the scene in 2D view and use of illumination models for this.

3. **Machine Learning (DSE-3A)**

This course introduces several fundamental concepts and methods for machine learning. The objective is to familiarize the students with some basic learning algorithms and techniques and their applications, as well as general questions related to analyzing and handling large data sets.

Learning Outcomes: By the end of this course, students should be able to:

- understand complexity of Machine Learning algorithms and their limitations;
- understand modern notions in data analysis oriented computing;
- be capable of confidently applying common Machine Learning algorithms in practice and implementing their own;
- be capable of performing distributed computations;
- be capable of performing experiments in Machine Learning using real-world data.

4. Project Work/ Dissertation (DSE-4C)

Learning Outcomes: By the end of this course, students should be able to:

- Manage the selection and initiation of individual projects and of portfolios of projects in the enterprise.
- Conduct project planning activities that accurately forecast project costs, timelines, and quality. Implement processes for successful resource, communication, and risk and change management.
- Demonstrate effective project execution and control techniques that result in successful projects.
- Conduct project closure activities and obtain formal project acceptance.
- Demonstrate a strong working knowledge of ethics and professional responsibility.

- Demonstrate effective organizational leadership and change skills for managing projects, project teams.

Department of Computer Science
Program Offered: Generic Electives offered by Department of Computer Science to other Disciplines
Course Outcome

Semester 1

Introduction to Programming (GE-I)

The paper provides students with an introduction to Object Oriented Programming concepts: classes, member functions, stream I/O and inheritance. It also provides an exposure to problem-solving through programming and aims to train the student to the basic concepts of the C++ programming language.

Learning Outcomes: By the end of this course, students should be able to:

- Examine high-level programming languages
- Determine what is a compiler and what it does
- Examine ways to output results using output statements
- Use preprocessor directives and why they are necessary
- Debug syntax errors
- Explore how to properly structure a program, including using comments to document a program
- Learn how to write a C++ program
- Explore how to read data from the standard input device
- Learn how to use predefined functions in a program
- Examine relational and logical operators
- Explore how to form and evaluate logical (Boolean) expressions
- Understand how to use the selection control structures if, if…else, and switch in a program
- Explore how to construct and use count-controlled, sentinel-controlled, flag-controlled, and EOF–controlled repetition structures
- Examine break and continue statements
- Discover how to form and use nested control structures
- Explore how to construct and use a value-returning, user-defined function in a program
- Discover the difference between value and reference parameters
- Learn function overloading
- Explore the string data type and learn how to use the various string functions to manipulate strings
- Explore how to declare and manipulate data into arrays

- Explore how to sort an array using the bubble sort, selection sort, and insertion sort algorithms
- Learn about classes
- Learn about private, protected, and public members of a class
- Explore how classes are implemented

Semester 2

Introduction to Database System (GE-2)

This paper helps the student to understand, appreciate and effectively explain the underlying concepts of database technologies, to design and implement a database schema for a given problem-domain. The practical sessions impart them with the knowledge of normalization and query a database using SQL DML/DDL commands.

Learning Outcomes: By the end of this course, students should be able to:

- Describe the features of database management systems.
- Construct database models from informal descriptions, and translate between such models.
- Purpose of a database system, data models, database languages. Design issues, Entity-Relationship diagrams, mapping to tables Relational database design concepts, decomposition and normalization, integrity constraints.
- Retrieve Data Using the SQL SELECT Statement
- Restrict and Sort Data
- Use Single-Row Functions to Customize Output
- Use Conversion Functions and Conditional Expressions
- Report Aggregated Data Using the Group Functions
- Display Data from Multiple Tables
- Use Sub queries to Solve Queries
- Use the Set Operators
- Manipulate Data
- Use DDL Statements to Create and Manage Tables
- Implement, manipulate and query relational databases.

Semester 3

Computer Networks and Internet Technologies (GE-3)

This paper introduces the basic knowledge on Computer Networks i.e. how the computers are connected in a network, what are the different layers, how data is travelled and exchanged between two computer systems. The second part deals with introduction of creating web pages, forms using HTML, and usage of little bit of JavaScript.

Learning Outcomes: By the end of this course, students should be able to:

- Understand the fundamental concepts of data communications and networking
- Identify different components of communication system.
- Apply the knowledge, concepts and terms related to data communication and networking.
- Solve problems in networking by referring to problems solving steps through relevant information by choosing suitable techniques.
- Explain the history of the internet and related internet concepts that are vital in understanding web development.
- Discuss the insights of internet programming and implement complete application over the web.
- Demonstrate the important HTML tags for designing static pages.
- Utilize the concepts of JavaScript and Java in brief.

Semester 4

Information Security and Cyber Laws (GE-4)

Students will learn the basics of Information Security, Anatomy of information Security Attacks their countermeasures and Fundamentals of Cyber Law through Virtual Training Environment.

Learning Outcomes: By the end of this course, students should be able to:

- Understand information security governance, and legal and regulatory issues.
- Understand external and internal threats.
- Familiarize themselves with information security awareness and a clear understanding of its importance.
- Familiarize them with how threats are discovered, analyzed, and dealt with.
- Understand fundamentals of secret and public cryptography.
- Understand protocols for security services.
- Familiarize themselves with network security threats and its countermeasures.
- Familiarize with network security designs using available secure solutions (such as PGP, SSL, IPSec, etc).
- Understand advanced security issues and technologies (such as DDoS attack detection and containment, and anonymous communications,).

Semester 5: BA (Program) Generic Elective

IT Fundamentals

This course will provide the fundamental skills and concepts required to maintain, support, and work efficiently with personal computers. This course is designed to teach the students basic concepts of computer system, Networks and Internet. It includes computer hardware, computer software, networking, security, and basic IT literacy.

Learning Outcomes: By the end of this course, students should be able to:

- Develop a vocabulary of key terms related to the computer and to software program menus.
- Understand the concept of input and output devices of Computers and how it works and recognize the basic terminology used in computer programming
- Identify the components of a personal computer system.
- Demonstrate mouse and keyboard functions.
- Demonstrate window and menu commands and how they are used.
- Demonstrate how to organize files and documents on a USB/hard drive.
- Compose, format and edit a word document.
- Send email messages (with or without attachments).
- Navigate and search through the internet.

Program Outcome of B.Sc.(Hons.) Computer Science

- Serve as the Programmers or the Software Engineers with the sound knowledge of practical and theoretical concepts for developing software.
- Serve as the Computer Engineers with enhanced knowledge of computers and its building blocks.
- Work as the Hardware Designers/Engineers with the knowledge of Networking Concepts.
- Work as the System Engineers and System integrators
- Serve as the System Administrators with thorough knowledge of DBMS.
- To Give Technical Support for the various systems.
- Work as the Support Engineers and the Technical Writers
- Work as Consultant and Management officers for system management.
- Work as IT Sales and Marketing person.
- Serve as the IT Officers in Banks and cooperative societies.
- Work as DTP Operator in small-scale industries.
- work as: Programmer or Software Engineer, Computer Engineer, Web Designer, Hardware Designer/Engineer, Systems Engineer, System integrator, System Administration, Technical Support, Support Engineer, Technical Writer, Consultant, IT Sales and Marketing IT Officer, Computer Scientist, Research Staff Member, Systems Analyst, Logic Designer, Computer Scientist in research and R & D laboratories. Serve as the Web Designers with latest web development technologies.

Program Specific Outcomes

Students seeking admission to B.Sc. (Hons.) Computer Science is proposed to get equipped with following outcomes:

A. An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.
B. Effectively communicating computing concepts and solutions to channel the gap between computing industry and business to create and initiate innovation.
C. Ability to use and apply system design engineering process in order to design, plan and implement software systems.
D. To prepare students for a career in an information technology industry or for higher studies in computer science or other scientific or technical fields.
E. Effectively employing the knowledge of computing principles to develop viable solutions to existing and future computing problems.
F. An ability to function effectively on teams to accomplish a common goal.
G. An understanding of professional, ethical, legal, security and social issues and responsibilities
H. An ability to communicate effectively with a wide range of audiences
I. An ability to use current techniques, skills, and tools necessary for computing practice
J. An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices.