Shri Shivaji Science and Arts College, Chikhli.
Department of Computer Science
Program Outcome, Program Specific Outcome and Course Outcome

Program Outcome

- PO-1. Demonstrate, solve and understanding of major concepts in alldisciplines of Computer Science.
- PO-2. Solve the problems and also think methodically, independentlyand draw a logical conclusion.
- PO-3. Employ critical thinking in programming way and the scientificknowledge to design, carry outs, records and analyze the results of practically done with the help of programming language.
- PO-4. Create an awareness of the impact of programming languageand use of computer science with in it on in a useful way on the environment, society and development outside the scientific community and as well as in a society.
- PO-5. Demonstrate proficiency in problem-solving techniques usingthe computer.
- PO-6. Able to do Hands on work in proficiency in at least two high-level programming languages and two operating systems.
- PO-7. In depth proficiency in the analysis of complex problems andthe synthesis of solutions to those problems.
- PO-8. Practically gaining the knowledge of comprehension of modernsoftware engineering principles.
- PO-9. Major impact on a breath and depth of knowledge in the discipline of Computer Science. PO-10. The revised and restructuredcurriculum for the Three-year integrated course is

systematically designed considering the current industry needs in terms of skills setsdemanded under new technological environment.

- PO-11. The proposed curriculum is more contextual, industry affable and suitable to cater the needs of society and nation in present daycontext.

B. Sc. I (Sem I)

CO

- To understand the computer fundamentals like operating systems, peripheral devices, internet and it's types.
- To understand various DOS commands and features of windows
- To understand internet and Types of Internet connections.
- To understand Programming Concept like Algorithm flowcharting programming languages.
- To understand assembler, interpreter and compiler
- To understand C language its history, features and structure of C program
- To understand keywords, identifiers, constants, variables, basic data types of C.
- To understand I/O Operations like Formatted I/O and Unformatted I/O
- To understand Control structure and conditional operator with its Applications

CO Practicals

- Execution of various DOS commands.
- Application of number systems.
- To understand and applications of web browser and E-mail.
- Design, implement, test, debug, and documents programs in C

  Implementation of Arithmetic, relational, logical operaotrs.
- Demonstration & use of various I/O operations.
- To understanding about writing algorithms and step by step approach in solving problems with the help of flowchart.
- Demonstration of Control structure and conditional operator.

B. Sc. I (Sem II)

CO

- To understand the fundamentals of data structure like list, array, stack, queue.
- To understand algorithms of traversing, insertion and deletion operation.
- To understand Linked list, circular queue & there implementation.
- To understand Tree, tree Traversing, sorting and searching Techniques.
- To understand Function in C language function prototype, local

&global variable.

- To understand and implementation of array.
- To understand String Handling in C language.
- To understand and implementation Pointers and Pointer and array.
- To understand and implementation of Structure and Union.
- To understand and implementation of File Handing and I/O Operationson file.

CO Practicals

- To implement data structure like array, stack, queue.
- To demonstration of algorithms of traversing, insertion and deletion operation on Linked list and circular queue.
- Demonstration of tree, tree Traversing, sorting and searching Techniques.
- Demonstration of function in c language
- To implementation of function prototype, local & global variable.
- To demonstration and implementation of array.
- To demonstration of String Handling in C language.
- To demonstrate and implementation Pointers and Pointer and array.
- To demonstration and implementation of Structure and Union

B. Sc. II (Sem III)

CO

- Understand basic data structures such as arrays, linked lists, stacksand queues and their applications.
- Apply Algorithm for solving problems like sorting, searching, insertionand deletion of data.
- To design and implement various data structure algorithms.
- To impart a thorough understanding of non-linear data structuressuch as trees and there applications.
- Familiarity with various sorting, searching techniques and theirperformance comparison.
- Determine and demonstrate bugs in program, recognize needed basicoperations with data structures.
- Describe the procedural and object-oriented paradigm with concepts of streams, classes, functions, data and objects.
- Students should be able to write, compile and debug programs in C++ language.
- Use different data types in a computer program.
- Basic idea about Classes and object with specifies, data members and member functions.
- Understanding the pure concept of managing console I/O with manipulators and operators.
- Using functions with their features and design programs involving decision, structures, loops and using different functions.
- **C**onstructor and Destructor types and usage.
- Operator overloading usage in a way that unary and binary operators

- With the help of Inheritance understand the visible mode and
- virtual as well as abstract base classes for object-oriented programming.

CO Practicals

- Programs to demonstrate fundamental algorithmic problems include tree.
- Implement various searching and sorting algorithms.
- Programs to demonstrate the implementation of various operations onstack and queue.
- To understanding about writing algorithms and step by step approach in solving problems with the help of fundamental data structures.
- The strengths of C++, which provide the students with the means of writing modular, efficient, maintainable, and portable code.
- Implement the use of various OOPs concepts with the help of programs.
- Identify with the help of classes and objects defining data member and member functions, accessing members in creation of program and usage of the supported factors needed for a finding the solution to specific problem.
- To learn how to overload functions and operations in C++.
- Demonstrates how to achieve reusability using inheritance, interfaces and packages and describes faster application development can be achieved.

B. Sc. II (Sem IV)

CO

- To introduce the concept of DBMS with respect to the relational model, to specify the functional and data requirements for a typical database application and to understand creation, manipulation and querying of data in databases.
- Identify the basic concepts and various data model used in database design ER modeling concepts and architecture use and design queries using SQL.
- Apply relational database theory and be able to describe relational algebra expression, tuple and domain relation expression to the form of queries.
- Identify the basic database storage structures and access techniques such as file organizations, indexing methods.
- An understanding of normalization theory and apply such knowledge to the normalization of a database.
- Recognize the purpose of query processing and optimization and also to demonstrate the basic of query evaluation.
- Design entity relationship and convert entity relationship diagrams into RDBMS and formulate SQL queries on the respect data.
- Understanding the concept of SQL DDL, DML commands and Clauses.
- Apply and relate the concept of Function with the supported format of Number, Character, Conversion and Date the in the database.
- In database management system and SQL commands along with PL/SQL using Oracle gives the student mastery on an open source-based toolkit,

which has more scope in the job market.

- Perform PL/SQL programming using concept of Cursor Management and Triggers.

- Deep Knowledge about the Transaction in processing PL/SQL.
  Able to understand the concept of securities in database which is mostly dependent on PL/SQL.

CO
Practicals

- Transform an information model into a relational database schema and to use a data definition language and utility to implement the schema using a DBMS.
- Using an SQL interface of a multi-user relational DBMS packageto create, secure, populate, maintain, and query a database.
- Formulate query, using SQL, solutions to a broad range of query and data update problems.
- Use a desktop database package to create, populate, maintain, and query a database.
- Demonstrate a rudimentary understanding of programmatic interfaces to a database and be able to use the basic functions of one such interface.
- Analyze an information storage problem and derive an information model expressed in the form.
- Understand query processing and techniques involved in query optimization.
- Improve the database design by normalization.
- Execute various PL/SQL queries related to Transaction Processing.
- Understand the PL/SQL architecture and write PL/SQL code for procedures, Cursor Triggers.

B. Sc. III (Sem V)
CO

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

- To explore .NET technologies for designing and developing dynamic, interactive and responsive web applications. To build Windows applications using structured and object-based programming techniques.
- Learn about .NET framework developed by Microsoft.
- Design and develop professional console and window-based .NET application.
- Be able to understand use of VB.NET basics, Objects and Types, Inheritance.
- To develop, implement, and demonstrate Component Services, Threading, Remoting, Windows services, web.
- To understand and be able to explain Security in the .NET frameworkand Deployment in the .NET.
- To develop Assemblies and Deployment in .NET, Mobile Application Development.
- Demonstrate knowledge of object-oriented concepts Design user experience and functional requirements VB.NET application.
- Construct classes, methods and assessors also the instantiate objects.
- Understand and implement string manipulation, events and exception handling within .NET application environment.
- Use Object Oriented paradigm to develop code and understand the concepts of Core Java and to cover-up with the pre- requisites of

Core java. The introduction to Java programming is done through the framework of object-oriented systems.

- Understand fundamentals of object-oriented programming in Java, including defining classes, invoking methods, using class libraries.
- Be able to use the Java SDK environment to create, debug and run simple Java programs.
- Identify Java language components and how they work together inapplications.
- Apply the above to design, implement, appropriately document andtest a Java application of medium complexity, consisting of multipleclasses.
- The knowledge about basic Java language syntax and semantics to writeJava programs and use concepts such as variables, conditional and iterative execution methods.
- Understand the fundamentals of object oriented programming in Java, including defining classes, objects, invoking methods and exception handling mechanisms.
- Understand the principles of inheritance, packages and interfaces.

CO

Practicals

- To aware with complete the all introductory part of to .Net IDE Component Framework.
- To develop, implement and creating Applications with VB.NET.
- Programming concepts in .Net Framework.
- Understand .NET framework and can realize some of the major enhancements in the new version of VB.
- Experience to using the VB .NET environment and how to develop small programs.
- Develop programs using Decisions, loop and Arrays in VB .NET.
- To design and program stand-alone Java applications.
- Understand the basic concepts such as Classes, methods, function Overloading, array and string manipulation in Java.
- Apply the types of inheritance in Java.

Implement Strings, packages, and Interface techniques

B. Sc. III (Sem VI)

CO

- Design and develop GUI applications using Abstract WindowingToolkit (AWT), Swing and Event Handling.
- Explore Exception Handling and design the Multithreading application in Java.
- Understand the concept of applets by how to create and run applets and Graphics programming by various classes in the graphics class.
- Interact with the concept of applets life cycle and creating with supported methods in Java
- Event handling with the application of AWT in Java.
- Design and implement windows application using windows forms, control library
- Single and Multiple form-based and menu -based technique shows in a .Net applications using basic and advanced control.
- Implementation of GUI application with Form Controls and its Event.
- Handle controls in Forms (message Box, Input Box), Windows MDI forms and Controls (Textbox, Creating Multiline, Word Wrap textboxes)
- Connect database by using ADO.NET and manipulate the database
- ADO.net based database driven .Net application.
- Understand ADO.NET and develop database applications.

CO

Practicals

- Working on the concepts of multi-threading by using thread class and implementing Run able interface.
- Find out the errors and exceptions, keywords that are used to manage exceptions and various ways in Java application.
- Develop Applet programs and manipulate the IO streams.
- Create a full set of UI widgets and other components, including windows, menus, buttons, checkboxes, text fields, scrollbars and scrolling lists, using Abstract Windowing Toolkit (AWT).

  Apply event handling on AWT with Java applications.
- Understand the VB .NET environment and how to develop smallprograms.
- Develop menu-based program for text manipulation.
- Connect database by using ADO.NET and manipulate the database.
- Develop the applications using Data Grid for displaying records.

Program Outcome (PO)
&
Program Specific Outcome(PSO)
M. Sc. Computer Science

PO

- PO-1. An ability to apply knowledge of computing and mathematics appropriate to the discipline.
- PO-2. An ability to identify, formulate, and develop solutions to computational challenges.
- PO-3. An ability to design, implement, and evaluate a computational system to meet desired needs within realistic constraints.
- PO-4. An ability to function effectively on teams to accomplish shared computing design, evaluation, or implementation goals.
- PO-5. An understanding of professional, ethical, legal, security, and social issues and responsibilities for the computing profession.
- PO-6. An ability to communicate and engage effectively with diverse stakeholders.
- PO-7. An ability to analyze impacts of computing on individuals, organizations, and society.
- PO-8. Recognition of the need for and ability to engage in continuing professional development.
- PO-9. An ability to use appropriate techniques, skills, and tools necessary for computing practice.

- PO-10. An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computational systems in a way that demonstrates comprehension of the trade offs involved in design choices.
- PO-11. An ability to apply design and development principles in the construction of software systems of varying complexity.

PSO

- PSO1. Communicate computer science concepts, designs, and solutions effectively and professionally.
- PSO2. Apply knowledge of computing to produce effective designs and solutions for specific problems
- PSO3. Use software development tools, software systems, and modern computing platforms

M. Sc. I (Sem I)
CO-I

- Apply the principles of number system, binary codes and Boolean algebra to minimize logic expressions.
- Develop K-maps to minimize and optimize logic functions up to 5 Variables.
- Acquire knowledge about various logic gates and logic families and analyze basic circuits of these families.
- Design various combinational and sequential circuits such as encoders, decoders and counters using multiplexers, and flip–flops.
- Describe and compare various memory systems, shift registers and analog to digital and digital to analog conversion circuits.
- Understand the taxonomy of microprocessors and knowledge of contemporary microprocessors. Describe the architecture, bus structure and memory organization of 8086 as well as pin diagram, signal description, register organization.
- Explore techniques for interfacing I/O devices to the microprocessor 8086 Demonstrate programming using the various addressing of 8086 microprocessor.

CO-II

- To Understanding .net, the C# environment
- To understand and implement framework base classes, user and program Interfaces.
- Be able to understand use of C# basics, Objects and Types, Inheritance, program structure; Literals, variables and data types, operators, Expressions, Decision making and branching, looping, methods in c#
- To develop, implement and creating Applications with C#.
- To understand and implement Operator overloading.
- To understand and be able to explain Multithreading in c# with Implementation.
- To develop or implement Data Access with .Net: ADO.net.

CO-III

- After completing this course, students will be able to: Allocate Main Memory based on various memory management techniques.
- Compare Memory allocation using Best fit, Worst fit, and first fit policies. Apply page replacement policies for dynamic memory management.
- Schedule CPU time using scheduling algorithm for processors.
- Compare various device scheduling algorithms.

CO-IV

- Understand computer network basics, network architecture, TCP/IP and OSI reference models.
- Identify and understand various techniques and modes of transmission
- Describe data link protocols, multi-channel access protocols and IEEE 802 standards for LAN Describe routing and congestion in network layer with routing algorithms and classify IPV4 addressingscheme,
- To understand Routing; Internet Protocol: IP Addressing, IPv4: Classes and Packet format, DHCP; ICMP; Routing in the Internet: RIP, OSPF, BGP.
- Discuss the elements and protocols of transport layer
- Understand network security and define various protocols such as FTP, HTTP, Telnet, DNS
- To understand Network Management Architecture; Internet Network Management Framework; SMI, MIB, SNMP.

CO-Practicals (Lab-I)

- Study of logic gates and realization of OR,AND,NOT AND XOR Functions using universal gates Design and implement combinational circuits like half adder/full adder, half subtractor /full subtractor, code converters, comparators, MUX / DEMUX
- Design and implement sequential circuits like flip-flops, counters and shift registers
- Demonstration of 8-bit DAC and 8-bit ADC Solve basic binarymath operations using the instructions of microprocessor 8086.
  Apply programming knowledge using the capabilities of the stack, the program counter. Design, code and debugs
- Assembly Language programs to implement simple programs
  Execute a machine code program on the trainingboards.
- Student can understand internal structure and operations of OS along with various processes including threading, inter process communication and synchronization with I/O operations.

CO-Practicals (Lab-II)

- Understand code solutions and compile C# projects within the .NET framework
- Demonstrate knowledge of object-oriented concepts Design user experience and functional requirements C#.NET application.
- Construct classes, methods, and assessors, and instantiate objects Understand and implement string manipulation, events within .Net application environment.
- Identify and resolve problems (debug /trouble shoot) in C#.NET window based application Design and Implement database connectivity using ADO.NET in window based application Identify and resolve problems (debug /trouble shoot) in C#.NET window based Application.

M. Sc. I (Sem II)
CO-I

- To learn graphics and animation on the web pages, using Java Applets.
- To learn and design a full set of Event driven UI widgets and othercomponents, including windows, menus, buttons, checkboxes, text, fields scrollbars and scrolling lists, using Abstract Windowing Toolkit (AWT) & Swings
- To learn Java Data Base Connectivity (JDBC) so as to retrieve and manipulate the information on any relational database through Java programs.
- To learn the server side programming using Servlets and JSP.
- To learn Java Bean so as to make the reusable software components.
- To learn invocation of the remote methods in an application using RMI
- To learn the development of Enterprise based applications, using EJB: Stateful, Stateless and Entity Beans
- To make the students familiar with Struts frameworks, which gives the opportunity to reuse the codes for quick development
- To learn Hibernate for the mapping of Java classes and objects associations to the relational database tables

CO-II

- After successfully completing this course, students will be able to know:
- Learn the notions of data structure, Abstract Data Type.
- Understand basic data structures such as arrays, linked lists, stacks and queues.
- To evaluate various methods of linked list formulation. Also explore different kinds of linked lists and their applications in day to day problem solving.
- To evaluate various formulation of queues. Also explore different kinds queues and their applications and implementations in simulations.
- To learn Sorting: Insertion sort, merge sort, Heaps and heap sort. Quick sort, Linear sort, priority queue, order statistics, lower bounds for sorting.
- To learn Searching: Balanced tree, red-black tree, lower bounds for Searching.
- To learn Graph: representation and algorithms, Breadth-first search (BFS), Depth-first search (DFS), topological sorting, Shortest Paths, Single source shortest paths problem, Minimum spanning tree, topological ordering, sparse matrices, linked list implementation of graph and graph traversal.
- To explore hashing, and various implementations of searching and hashing algorithms.

CO-III

- Learn the phases of software development LO2. LO3. LO4. LO5.
- Develop process models and process system models.
- Gather, understand, analyze and specify requirements, Analyze and translate a specification into a design, and then realize that design practically, using an appropriate Software engineering methodology.
- Know how to develop the code from the design and effectively apply relevant standards and perform testing, and quality management and practice
- Develop architectural diagram, and implement by following coding principles
- Able to use modern engineering tools necessary for software project Management, time management and software reuse.
- Apply testing strategies and handle software product maintenance Issues.

CO-IV

- Ability to understand Sets and their algebra, duality, power sets and partitions. Principle of Strong Mathematical Induction, set theory.
- Ability to analyze various binary relations characteristic function and Recursive functions.
- Ability of Counting, Algebraic Structures, Algebraic systems, and Solving cosets and Lagrange's theorem
- Ability to understand logical operators, Implications, Lattice as POSETs and properties,
- Lattice as algebraic systems, sublattices, Direct product and homomorphism, Special lattices, Boolean algebra
- Ability to model problems using Graphs , connectivity, Rooted trees, simple precedence grammars-syntax terminology, a view of parsing, notion and use of precedence relations, formal definition of precedence relations.
- To understand Turing machines and partial recursive functions. Ability to learn the notions of languages, finite state automata, phrase structure grammars, finite state machines.

CO-V

Understand the structure of compilers

- Understand the basic techniques used in compiler construction such as lexical analysis, top-down, bottom-up parsing, context-sensitive analysis, and intermediate code generation.
- Understand the Memory Allocation like Static and dynamic memory allocation, array allocation and access, allocation for strings, structure allocation.
- Understand the basic data structures used in compiler, Compilation of control structures: Control transfers, procedural calls, conditional execution, iteration control constructs Understand Error detection, indication and recovery. Compilation of I/O statements: Compilation of I/O list, compilation of FORMAT list, the I/O routine, file control.
- Understand Code optimization, program flow analysis, Global optimization, writing compilers.

CO Practicals (Lab-III)

- To build software development skills using java programming for real world applications.
- To implement frontend and backend of an application.
- To implement classical problems using java programming.
- Read and make elementary modifications to Java programs that solvereal-world problems.
- Validate input in a Java Program.

CO Practicals (Lab-IV)

- Able to prepare SRS document, design document, test cases and software configuration management and risk management related document.
- Apply various white box and black box testing techniques.
- Develop function oriented and object oriented software design using tools like rational rose.
- Able to perform unit testing and integration testing.
- Able to track the progress of a project using Openproj tool.
- Implement basic data structures such as arrays and linked list.
  Programs to demonstrate fundamental algorithmic problems including Tree Traversals, Graph traversals, and shortest paths.
- Implement various searching and sorting algorithms.
- Programs to demonstrate the implementation of various operations on stack and queue.

M. Sc. II (Sem III)

CO-I

- 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.

CO-II

- Describe and Synthesise concepts of programming for networking, including, multithreading, delegate and event handling, remote files I/O and database connectivity.
- Develop Code for basic network and Internet protocols including sockets, stream and packet protocols such as TCP, UDP, HTTP, FTP and SMTP protocols for creating simple two tier client server applications.
- Program multi-tier client server computing systems with remote and web services protocols for creating distributed client server Systems.
- Design and develop specialized client server systems with better security, scalability, queuing, and optimal performance and bandwidth utilization;
- Program different network programming tools, network monitoring, tracking and analyzing advanced client server systems

CO-III

- To provide hardware and software issues in modern distributed systems.
- To get knowledge in distributed architecture, naming, synchronization, consistency and replication, fault tolerance, security, and distributed file systems.
- To analyze the current popular distributed systems such as peer-to-peer (P2P) systems will also be analyzed.
- To know about Shared Memory Techniques
- Have Sufficient knowledge about file access.
- Have knowledge of Synchronization and Deadlock.

CO-IV

- Understanding of the basic kinds of finite automata and their capabilities.
- Understanding of regular and context-free languages.
- Understanding of the key results in algorithmic complexity, computability and solvability of problems.
- Ability to describe and transform regular expressions and grammars.
- Using the Prolog language as an experimental tool for testing properties of basic computational structures.
- Understanding the key notions of computation, such as algorithm, computability, decidability, reducibility, and complexity, throughproblem solving.
- Understanding Natural Language.

CO-Practicals (Lab-V)

- Demonstrate the use of ARFF files taking input and display the output of the files.
- Implement the Preprocess and classify Customer dataset.
- Perform Preprocessing, Classification techniques on Agriculture dataset.
- Preprocess and classify Weather dataset
- Perform Clustering and association techniques on Customer/agriculture dataset.
- Compare various Data Mining techniques available in WEKA
- Understand the basic concepts of computer graphics.
- Design scan conversion problems using C++ programming.
- Apply clipping and filling techniques for modifying an object.
- Understand the concepts of different type of geometric transformation of objects in 2D and 3D.
- Understand the practical implementation of modeling, rendering, viewing of objects.

CO-Practicals (Lab-V)

- Design and Set up a client /server environment using LAN and WAN Scenarios.
- Design and build client server applications with network programming exposure. Understand basic networking concepts using sockets.
- Examine the techniques which are required to develop network application/internet-based application.
- Compare various application deployment mechanisms and the useof digital certificates
- Outline the prevention strategies for network attacks with at least one prevention technique providing in two or three tier environments.

M. Sc. II (Sem IV)

CO-I

- Represent Knowledge using various knowledge representation schemes.
- Understand Artificial Neural Networks and its applications
- Understand the basic knowledge acquisition methods.
- Understand the theoretical base of the expert system and its development process.
- Differentiate between different knowledge representation techniques and describe methods of knowledge acquisition and extraction.
- Develop expert systems using various available tools
- Analyze the development process of expert system through various case studies.

CO-II

- Understand the basic concepts of algorithms and analyze the performance of algorithms.
- Discuss various algorithm design techniques for developing algorithms.
- Discuss various searching, sorting and graph traversal algorithms.
- Understand NP completeness and identify different NP complete problems.
- Discuss various advanced topics on an algorithm.

CO-III

- Describe network security services and mechanisms
- Understanding Breaking an Encryption Scheme, Types of Cryptographic Function, Respective Algorithms of cryptography, Cryptographic Authentication Protocols,
- Understanding Kerberos V4: Tickets, Kerberos V5: ASN.1, Names, Delegation of Rights, Ticket Lifetimes, Key Versions, Optimizations..
- Various network security applications, IPSec, Firewall, IDS, Web security, Email security, and Malicious software etc.
- Understanding Firewalls.

CO-IV

- Understanding the basic principles of mobile communication systems AND Satellite Systems Understanding an analysis of mobile communications with the interpretation of the call prints, Wireless LAN: Infrared Versus Radio Transmission, Infrastructure and Adhoc Network.
- Understanding the basic principles of the modern mobile and wireless communication systems and Mobile Network Layer.
- Understanding the operation of mobile communications systems and their generation divisions Understanding Support for Mobility like File Systems, World Wide Web, Wireless

  Application Protocol, i-Mode, SyncML.

CO-V

- Examine various types of images, intensity transformations and
- Develop Fourier transform for image processing in frequency domain.
  Learn different techniques employed for the enhancementof images.
  Evaluate the methodologies for image segmentation, restoration etc
  Implement image process & analysis of algorithms
- Learn different causes for image degradation and overview of image restoration techniques.
- Understand the need for image compression and to learn the spatial and frequency domain techniques of image compression.
- Learn different feature extraction techniques for image analysis and recognition.

CO-VI

- Have an ability to apply software testing knowledge and engineering methods.
- Understanding the software test outline to test cases, creating test cases, documentation short cuts, introduction to using tables and spreadsheet.
- Have an ability understand and identify various software testing problems, and solve these problems by designing and selecting software test models, criteria, strategies, and methods.
- Create test strategies and plans, design test cases, prioritize and execute them.
- Able to develop the Testing Web Applications
- Manage incidents and risks within a project.

CO-Practicals (Lab -VII)

- Apply various AI search algorithms (uninformed, informed, heuristic, constraint satisfaction,) Implement fundamentals of knowledge representation, inference and theorem proving using AI tools
- To implement advance learning techniques-search
- Ability to write programs in java to solve problems using algorithm design techniques such as Divide and Conquer, Greedy, Dynamic programming, and Backtracking.
- Able to implement Quick sort, Merge sort algorithm, BFS and DFS algorithms
- Able to implement backtracking algorithm for the N-queens problem.
- Able to implement greedy algorithm for job sequencing with deadlines.
- Get awarded with Dijkstra's , Prim's algorithm , Kruskal's algorithm on spanning tree.
- Able to implement Floyd's algorithm for the all pairs shortest path problem.