KAVAYITRI BAHINABAI CHAUDHARI NORTH MAHARASHTRA UNIVERSITY, JALGAON

School of Computer Sciences

MASTER OF SCIENCE in Computer Science
[M. Sc. (Computer Science)]

Syllabus

[*under CBCS and Academic Flexibility*]

Faculty of Science and Technology

With effect from 2019-20

MASTER OF SCIENCE in Computer Science [M.Sc. (Computer Science)]

PROGRAMME OBJECTIVES (POs):

1) **Broadly Educated and Versatile** - Able to draw upon foundational knowledge, learn, adapt and successfully bring to bear analytical and computational approaches on changing societal and technological challenges.
2) **Inspiring and Collaborative** - Able to induce and contribute to diverse teams, expertise, and experiences.
3) **Innovative** - Drives scientific and societal advancement through technological innovation and entrepreneurship.
4) **Engaged** - Is and remains engaged with the academics, technical and scientific professional communities.

PROGRAMME SPECIFIC OUTCOMES (PSOs):

M.Sc. (Computer Science) Programme has been designed to prepare graduates for attaining the following program outcomes:

1) An ability to apply knowledge of computer science appropriate to the discipline.
2) An ability to apply computer science foundations, algorithmic principles, and computer science theory in the modeling and design of computational systems in a way that demonstrates comprehension of the tradeoffs involved in design choices.
3) Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.
4) Communicate effectively in a variety of professional and research contexts.
5) Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.
6) Apply computer science theory and software development fundamentals to produce computing-based solutions.
7) Acquire and apply new knowledge as needed, using appropriate learning strategies.

MASTER OF SCIENCE in Computer Science [M.Sc. (Computer Science)]

Degree Name	**:**	Master of Science in Computer Science [M.Sc. (Computer Science)] [Under Academic Flexibility]
Faculty	**:**	Science and Technology
Duration	**:**	02 years, Full Time course
Medium of Instruction	**:**	English
Pattern	**:**	Semester Pattern (04 semesters)
Examination Pattern	**:**	60% (External Assessment) + 40% (Internal Assessment)
Passing Standard	**:**	Separate Passing for internal as well as external assessment.
Evaluation Mode	**:**	CGPA
Lecture	**:**	Clock hour (60 minutes)

Kavayitri Bahinabai Chaudhari North Maharashtra University, Jalgaon

School of Computer Sciences

Syllabus under CBCS for Master of Science in Computer Science [M.Sc. (Computer Science)]

Course Structure (w.e.f. 2019-20)

COURSE STRUCTURE WITH CREDIT

Semester-I

Course Code	Course Type	Title of the Course	Contact Hours/Week			Distribution of Marks for Examination						Credits
						Internal		External		Total		
			Th(L)	Pr	Total	Th	Pr	Th	Pr	Th	Pr	
CS-101	**Core**	Data Structures and Algorithms	04	-	04	40	-	60	-	100	-	04
CS-102	**Core**	Database Management System (DBMS)	04	-	04	40	-	60	-	100	-	04
CS-103	**Core**	Automata Theory and Computability	04	-	04	40	-	60	-	100	-	04
CS-104	**Core**	Operating Systems	04	-	04	40	-	60	-	100	-	04
CS-105	Skill Based	Object Oriented Programming using JAVA	**04**	**-**	**04**	**40**	**-**	**60**	**-**	100	**-**	**04**
CS LAB-I	**Core**	LAB on Data Structures and Algorithms and JAVA programming	-	04	04	-	40	-	60	-	100	04
CS LAB-II	**Core**	LAB on DBMS	-	04	04	-	40	-	60	-	100	04
AC-101	**Audit Course**	**Practicing Cleanliness**	**-**	**02**	**02**	**-**	**100**	**-**	**-**	**-**	**100**	**02**

Semester-II

Course Code	Course Type	Title of the Course	Contact Hours/Week			Distribution of Marks for Examination						Credits
						Internal		External		Total		
			Th(L)	Pr	Total	Th	Pr	Th	Pr	Th	Pr	
CS-201	**Core**	Compiler Construction	04	-	04	40	-	60	-	100	-	04
CS-202	**Core**	Mathematical Foundations of Computer Science	04	-	04	40	-	60	-	100	-	04
CS-203	**Core**	Artificial Intelligence	04	-	04	40	-	60	-	100	-	04
CS-204	**Core**	Design and Analysis of Algorithms	04	-	04	40	-	60	-	100	-	04
CS-205	Skill Based	Python Programming	**04**	**-**	**04**	**40**	**-**	**60**	**-**	100	**-**	**04**
CS LAB-III	**Core**	LAB on Design and Analysis of Algorithms (DAA)	-	04	04	-	40	-	60	-	100	04
CS LAB-IV	**Core**	LAB on Python Programming	-	04	04	-	40	-	60	-	100	04
AC- 201 (A)/(B)/(C)/(D)	**Elective Audit Course**	**Choose one out of four (AC-201(A)/(B)/(C)/(D))** (Personality and Cultural Development Related)	**-**	**02**	**02**	**-**	**100**	**-**	**-**	**-**	**100**	**02**

List of Elective Audit Courses to be offered in Semester-II:

AC-201 (A) : Soft Skills	AC-201 (B) : Practicing Sports Activities
AC-201 (C) : Practicing Yoga	AC-201 (D) : Introduction to Indian Music

Semester-III

Course Code	Course Type	Title of the Course	Contact Hours/Week			Distribution of Marks for Examination						Credits
						Internal		External		Total		
			Th(L)	Pr	Total	Th	Pr	Th	Pr	Th	Pr	
CS-301	**Core**	Machine Learning	04	-	04	40	-	60	-	100	-	04
CS-302	**Core**	Web Application Development Technology	04	-	04	40	-	60	-	100	-	04
CS-303	**Core**	Computer Graphics and Digital Image Processing	04	-	04	40	-	60	-	100	-	04
CS-304	**Core**	Software Engineering	04	-	04	40	-	60	-	100	-	04
CS-305(A)/(B)/ (C)	Elective	Choose one from CS-305(A), CS-305(B) and CS-305(C)	**04**	**-**	**04**	**40**	**-**	**60**	**-**	100	**-**	**04**
CS LAB-V	**Core**	LAB on Web Application Development Technology	-	04	04	-	40	-	60	-	100	04
CS LAB-VI	**Core**	LAB on Computer Graphics and Digital Image Processing	-	04	04	-	40	-	60	-	100	04
AC-301 (A)/(B)/(C)/(D)	**Elective Audit Course**	**Choose one out of four (AC-301 (A)/(B)/(C)/(D))** (Technology + Value added course)	**-**	**02**	**02**	**-**	**100**	**-**	**-**	**-**	**100**	**02**

List of Elective Courses to be offered in Semester-III:

CS-305(A): Natural Language Processing	CS-305(B): Optimization Algorithms	CS-305(C): Data Warehousing and Data Mining (DWDM)

List of Elective Audit Courses to be offered in Semester-III:

AC-301 (A) : Computer Skills	AC-301 (B) : Cyber Security
AC-301 (C) : Linux (Spoken Tutorial Course)	AC-301 (D) : Advance C++ (Spoken Tutorial Course)

Note: Syllabus for Spoken Tutorial Courses AC-301 (C)/(D) is available at https://spoken-tutorial.org developed at IIT Bombay for MHRD, Government of India.

Semester-IV

Course Code	Course Type	Title of the Course	Contact Hours/Week	Distribution of Marks for Examination					Credits
				Internal		External	Total		
			Six Month Full Time Industrial Training	Th	Pr	Project Evaluation	Th	Pr	
CS-401	**Project**	**Full time Industrial Training**	Students contact to teachers through E-mail, AView Software and other ICT technologies throughout the Semester	-	-	Project Evaluation in the form of Demonstration, Project Report Writing, Confidential Report from the Industry, Viva Voce etc.	-	300	12

Semester-I

*Course Code:***CS-101** **Data Structures and Algorithms** *Clock Hours:* **60**
Total Marks: **100**

Course Objectives:
1 To impart the basic concepts of data structures and algorithms
2 To understand basic concepts about array, stacks, queues,linked lists, trees and graphs
3 To impart knowledge of advance topics like AVL Trees, BTrees, B* and B+ Trees
4 To understand concepts about searching and sorting techniques
5 Apply hashing concepts for a given problem
6 To understanding about writing algorithms and step by step approach in solving problems with the help of fundamental data structures

Unit-I **[08]** Max Marks:**06**
Introduction to Data Structures and Algorithms: Algorithmic Notation: Format Conventions, Statement and Control Structures. Time and Space Analysis: Data types and Abstract data types, Types of Data structures; Primitive, Non primitive, Linear and Nonlinear Data structures

Unit-II **[08]** Max Marks:**15**
Array: Storage representation, operations and applications (Polynomial addition and subtraction) Stack: operations and applications (infix, postfix and prefix expression handling), Queue: operations and applications, Circular Queues: operations and applications, Concept of Double ended Queue and Priority Queue, Linked representation of stack and queue.

Unit-III **[10]** Max Marks:**12**
Linked Lists: Operations and Applications of Linear linked list (Polynomial addition and subtraction), Circular linked list and Doubly linked list.

Unit-IV **[11]** Max Marks:**21**
Trees: Binary Trees, Binary Tree: Representations, Operations (insert/delete), Traversal (inorder, preorder, postorder, level order), Threaded Binary Tree, Search Trees: AVL Tree, single and double rotations, B-Trees: insertion and deletion, Introduction to B+ and B* Trees

Unit-V **[11]** Max Marks:**18**
Graphs and Their Applications: Representation (Matrix/Adjacency) and Traversal (Depth First Search/Breadth First Search), Spanning Trees, Minimal Spanning Tree (Prim's and Kruskals's algorithm), Shortest Paths and All Pair Shortest Path, Dijkstra's, Floyd-Warshall Algorithms.

Unit-VI **[12]** Max Marks:**18**
Hash Table: Hash Function, Collision and its Resolution, Separate Chaining, Open Addressing (linear probing, quadratic probing, double hashing), Rehashing, Extendible Hashing Searching: Linear Search and Binary Search (array/binary tree). Sorting: General Background, Sorting Techniques: Bubble Sort, Insertion Sort, Selection Sort, Quicksort, Mergesort, Heapsort and Radix Sort.

References:

1. Tremblay, J. & Sorenson, P.G., (2001), An Introduction to Data Structures with Application, Mcgraw Hill India, ISBN: 978-0074624715, 0074624717
2. Langsam, Y., Augenstein, M.J. & Tenenbaum A.M., (2015), Data Structures using C

and C++, 2nd Edition, Pearson Education ISBN: 978-9332549319, 9332549311

3. Balagurusamy, E., (2013), Data Structures using C, 1st Edition, Mcgraw Hill Education, ISBN: 978-1259029547, 1259029549
4. Weiss, M.A., (2002), Data Structures and Algorithm Analysis in C, 2nd Edition, Pearson India, ISBN: 978-8177583588, 8177583581
5. Horowitz, E., Sartaj S. & Mehta, D. (2008), Fundamentals of Data Structures in C++, Universities Press ISBN: 978-8173716065, 8173716064 *
6. Lafore, R., (2003), Data Structures & Algorithms in Java, 2nd Edition, Pearson India, ISBN: 978-8131718124, 8131718123
7. Kruse, R., Tondo, C.L., Leung B., & Mogalla S, (2006), Data Structures and Program Design in C, Pearson India, ISBN: 978-8177584233.

Course Outcome:

1. Student will be able to choose appropriate data structure as applied to specified problem definition.
2. Students will be able to use linear and non-linear data structures like stacks, queues, linked list etc.
3. To increase the student‘s intuitive understanding of search trees
4. To learn advanced tree data structures
5. To learn to represent data using graph data structure
6. Student will be able to handle operations like searching, insertion, deletion, traversing mechanism etc. on various data structures.
7. Students will be able to apply concepts learned in various domains like DBMS, compiler construction etc.

*Course Code:***CS-102** **Database Management System (DBMS)** *Clock Hours:* **60**
Total Marks: **100**

Course Objectives:

1) The course emphasizes the understanding of the fundamentals of relational systems including data models, database architectures, and database manipulations.
2) To develop conceptual understanding of database management system
3) To understand how a real-world problem can be mapped to schemas
4) To educate students with different Database Languages.
5) Understand the needs of database processing and learn techniques for controlling the consequences of concurrent data access.

Unit-I **[05]** Max Marks:**08**

Introduction: Database system application and purpose, Characteristics of DBMS, Database Users, 1-tier, 2-tier and 3-tier architecture of DBMS along with its advantages, Levels of Database Architecture, Data Models, Data-schemas and instances, Data Independence, Role and responsibilities of DBA.

Unit-II [**10]** Max Marks:**10**

Database Design and E-R Model: Overviews of Database Design, ER Modeling concepts, ER Diagrams, Reduction to Relational Schemas, Extended ER Features, Alternative notations for Modelling, Cardinality constraints, Atomic Domains and 1NF, Decomposition using Functional

Dependencies (BCNF, 3NF and 4NF).

Unit-III **[12]** Max Marks:**20**

Relational Databases: Structure of Relational Databases, Database Schemas, Keys, Schema diagrams, Relational Query Languages, Relational Operation. Overview of SQL, SQL Data Definition, Basic Structure of SQL Queries, Basic Operations, Set Operations, Null Values, Aggregate Functions, Nested Sub queries, Modification of Databases. Join Expressions, Views, Transactions, Integrity Constraints, SQL data types and Schemas, Authorization, Accessing SQL from Programming Languages, Overview of Dynamic SQL and SQL CLI. Functions and Procedures, Triggers. The relational Algebra fundamental and extended Operations. Tuple and Domain Relational Calculus.

Unit-IV **[10]** Max Marks:**22**

Transaction Management and Query Processing: Transaction Concept, Model, Storage Structure, Atomicity and Durability, Isolation, Levels of Isolation, Overview of Query Processing, Measuring Query Cost, Selection Operation, Sorting, Join Operation, Other Operations and Evaluation of Expression. Overview of Query Optimization, Transformation of Relational Expression, Choice of Evaluation Plan.

Unit-V **[10]** Max Marks:**15**

Concurrency Control and Recovery System: Lock based Protocol, Timestamp based Protocol, Validation based Protocol, Deadlock Handling, Failure Classification, Storage, Recovery and Atomicity, Recovery Algorithms, Buffer Management, Early lock release and logical undo operations, Remote Backup Systems. Case study: ARIES

Unit-VI **[13]** Max Marks:**25**

Advanced Topics in Databases: Type your unit content here. (comma separated) Introduction to Object Databases: Shortcomings of Relational Data Model, The Conceptual Object Data Model, Objects in SQL:1999 and SQL:2003. Introduction to XML and Web Data: Semi-structured Data, Overview of XML, XML Data Definitions, XML Schema, XML Data Manipulation: XQuery, XPath Query Languages: XPath and SQL/XML. Distributed Databases: Overview, Homogeneous and Heterogeneous Databases, Distributed Data Storage, Distributed Transactions, Commit Protocols, Concurrency Control, Cloud based Databases.

References:

1. Michael Kifer, Arthur Bernstein, P.M, Lewis and P.K. Panigrahi, “Database Systems: An Application Oriented Approach”, Second Edition, Pearson Education, ISBN:978-81-317-0374-8.
2. C.J.Date, A. Kannan and S. Swamynathan, “An Introduction to Database Systems”, Eighth Edition, Pearson Education, ISBN:978-81-7758-556-8
3. A. Silberschatz, H.F.Korth, and S.Sudarshan, “Database System Concepts”, TMH Publications, Sixth Edition, ISBN:978-007-132522-6.

Course Outcome:

Upon successful completion of this course, students will be able to

1. To analyze Database design methodology.
2. Acquire knowledge of fundamentals of Database Management System.

3. Analyze the difference between traditional file system and DBMS.
4. To deal with different Database languages.
5.Draw various data models for Database, writing and executing queries to get expected results.

Course Code: **CS-103** **Automata Theory and Computability** *Clock Hours:* **60**
Total Marks: **100**

Course Objectives:

Students will try to learn:

1. To learn fundamentals of Grammars and Languages.
2. To understand the relation between Regular Language and Finite Automata and machines.
3. To learn how to design Automata's and machines as Acceptors, Verifiers and Translators.
4. To understand the relation between Contexts free Languages, PDA and TM.
5. To learn how to design PDA as acceptor and TM as Calculators.
6. To understand the decidability and complexity measures.

Unit-I **[10]** Max Marks: **20**

Grammars: Production systems, Chomskian Hierarchy, Right linear grammar and Finite state automata, Context free grammars, Normal forms, uvwxy theorem, Parikh mapping, Self-embedding property, Subfamilies of CFL, Derivation trees and ambiguity

Unit-II **[10]** Max Marks: **20**

Finite State Automata: Nondeterministic and deterministic FSA, NFSA with ε- moves, Regular Expressions, Equivalence of regular expression and FSA, Pumping lemma, closure properties and decidability, Myhill - Nerode theorem and minimization, Finite automata with output

Unit-III **[08]** Max Marks: **15**

Pushdown Automata: Acceptance by empty store and final state, Equivalence between pushdown automata and context-free grammars, Closure properties of CFL, Deterministic pushdown automata

Unit-IV **[12]** Max Marks:**20**

Turing Machines: Techniques for Turing machine construction, Generalized and restricted versions equivalent to the basic model, Godel numbering, Universal Turing Machine, Recursively enumerable sets and recursive sets, Computable functions, time space complexity measures, context sensitive languages and linear bound automata

Unit-V **[08]** Max Marks: **10**

Decidability: Post's correspondence problem, Rice's theorem, decidability of membership, emptiness and equivalence problems of languages

Unit-VI **[10]** Max Marks: **15**

Complexity Measures: Time and tape complexity measures of Turing machines, Random access machines, the classes P and NP, NP-Completeness, satisfiability and Cook's theorem, Polynomial reduction and some NP-complete problems, Regulated rewriting L systems, Grammar systems

References:

1. K. Krithivasan and R. Rama (2009). Introduction to Formal Languages, Automata Theory and Computation: Pearson Education, ISBN 9788131723562.
2. J. E. Hopcroft, R.Motwani and J.D.Ullman (2001). Introduction to Automata Theory Languages and computation: Pearson Education Asia, ISBN 978-0321455369.
3. Peter Linz (2006). An Introduction to Formal Language and Automata 4th Edition: Narosa Publishing house, ISBN 978-1-4496-1552-9.
4. M.Sipser (1997). Introduction to the Theory of Computation; Singapore: Brooks/Cole, Thomson Learning ISBN, 978-1133187790.
5. John. C. Martin (2003). Introduction to the Languages and the Theory of Computation Third edition Tata McGraw-Hill ISBN 9780070660489.
6. 6. http://nptel.ac.in/

Course Outcome:

Students will able to:

1. Understand, design, construct, analyse and interpret Regular languages, Expression and Grammars.
2. Design different types of Finite Automata and Machines as Acceptor, Verifier and Translator.
3. Understand, design, analyse and interpret languages, Expression and Grammars.
4. Design different types of Push down Automata and Turing Machine.

Course Code: **CS-104** **Operating Systems** *Clock Hours:* **60**
Total Marks: **100**

Course Objectives:

The student should be able to

- understand different types of operating systems and the concepts that underlies operating systems.
- learn the fundamental concepts and algorithms that will be used in existing commercial operating systems.
- understand the issues related to protection and security.

Unit-I **[04]** Max Marks:**08**

Introduction: review of computer organization, introduction to popular operating systems like UNIX, Windows, etc., OS structure, system calls, functions of OS, evolution of OSs.

Unit-II **[03]** Max Marks:**06**

Computer organization interface: using interrupt handler to pass control between a running program and OS.

Unit-III **[08]** Max Marks:**12**

Concept of a process: states, operations with examples from UNIX (fork, exec), Process scheduling, inter-process communication (shared memory and message passing), UNIX signals.

Unit-IV **[04]** Max Marks:**06**
Threads: multithreaded model, scheduler activations, examples of threaded programs.

Unit-V **[06]** Max Marks:**10**
Scheduling: multi-programming and time sharing, scheduling algorithms, multiprocessor scheduling, thread scheduling (examples using POSIX threads).

Unit-VI **[08]** Max Marks:**12**
Process synchronization: critical sections, classical two process and n-process solutions, hardware primitives for synchronization, semaphores, monitors, classical problems in synchronization (producer-consumer, readers-writer, dining philosophers, etc.).

Unit-VII **[06]** Max Marks:**10**
Deadlocks: modelling, characterization, prevention and avoidance, detection and recovery.

Unit-VIII **[07]** Max Marks:**12**
Memory management: with and without swapping, paging and segmentation, demand paging, virtual memory, page replacement algorithms, working set model, implementations from operating systems such as UNIX. Current Hardware support for paging: e.g., Pentium/ MIPS processor etc.

Unit-IX **[07]** Max Marks:**12**
Secondary storage and Input/Output: device controllers and device drivers, disks, scheduling algorithms, file systems, directory structure, device controllers and device drivers, disks, disk space management, disk scheduling, NFS, RAID, other devices. operations on them, UNIX FS, UFS protection and security, NFS

Unit-X **[04]** Max Marks:**06**
Protection and security: Illustrations of security model of UNIX and other OSs. Examples of attacks.

Unit-XI **[03]** Max Marks:**06**
Epilogue: Pointers to advanced topics (distributed OS, multimedia OS, embedded OS, real-time OS, OS for multiprocessor machines).

All above topics shall be illustrated using UNIX as case-studies.

References:

1. Abraham Silberschatz, Peter B. Galvin, Greg Gagne (2009), Operating System Concepts, 8th Ed., John Wiley, *ISBN* 0-471-69466-5.
2. William Stallings (2014), Operating Systems: Internals and Design Principles. Pearson, 8th Ed., *ISBN*-13: 978-0-13-230998-1
3. AS Tanenbaum (2009), Modern Operating Systems, 3rd Ed., Pearson, *ISBN*: 0135013011
4. AS Tanenbaum, AS Woodhull (2006), Operating Systems Design and Implementation, 3rd Ed., Prentice Hall, *ISBN*-10: 0131429388

5. M. J. Bach (1986), Design of the Unix Operating System, Prentice Hall of India, *ISBN* 0 -13-201757-1 025

Course Outcome:

Upon completion of the subject, students will be able to:

- understand different types of operating systems.
- gain extensive knowledge on principles and modules of the operating systems.
- understand key mechanisms in the design of operating systems modules.
- understand process management, thread management, memory management, file management and deadlock handling.
- compare performance of different processor scheduling algorithms.
- produce algorithmic solutions to process synchronization problems
- understand the issues related to protection and security.

Course Code: **CS-105**	**Object Oriented Programming using JAVA**	*Clock Hours:* **60** *Total Marks:* **100**

Course Objectives:

Students will try:

1. To learn fundamentals of Java programming language and its constructs.
2. To understand concept of object-oriented programming concept using Java.
3. To study the concept of the Inheritance, Interfaces, Lambda Expressions, and Inner Classes.
4. To understand the concept of the Exceptions and Generic Programming
5. To learn about the Graphics Programming, Event Handling, Swing Components, and Database Programming

Unit- I **[08]** Max Marks:**12**

An Introduction to Java: Java as a Programming Platform, The Java “White Paper” Buzzwords, Java Applets and the Internet, Common Misconceptions about Java, The Java Programming Environment, Installation, A Simple Java Program, Comments, Data Types, Variables, Operators, Strings, Input and Output, Control Flow, Big Numbers, Arrays.

Unit-II **[08]** Max Marks:**12**

Objects and Classes: Introduction to Object-Oriented Programming, Using Predefined Classes, Defining Your Own Classes, Static Fields and Methods, Method Parameters, Object Construction, Packages, The Class Path, Documentation Comments

Unit-III **[10]** Max Marks:**16**

Inheritance, Interfaces, Lambda Expressions, and Inner Classes: Classes, Super classes, and Subclasses, Object: The Cosmic Superclass, Generic Array Lists, Object Wrappers and Autoboxing, Methods with a Variable Number of Parameters, Enumeration, Classes, Reflection, Interfaces, Examples of Interfaces, Lambda Expressions, Inner Classes, Proxies

Unit-IV **[10]** Max Marks:**16**

Exceptions and Generic Programming: Dealing with Errors, Catching Exceptions, Assertions and Logging, Why Generic Programming? Simple Generic Class, Generic Methods, Bounds for Type Variables, Inheritance Rules for Generic Types, Wildcard Types, Reflection and Generics

Unit-V **[12]** Max Marks:**24**

Graphics Programming, Event Handling and Swing Components: Introducing Swing, Creating a Frame, Positioning a Frame, Displaying Information in a Component, Working with 2D Shapes, Using Color, Using Special Fonts for Text, Displaying Images, Basics of Event Handling, Actions, Mouse Events, The AWT Event Hierarchy, Swing and the Model-View-Controller Design Pattern, Introduction to Layout Management, Text Input, Choice Components, Menus, Sophisticated Layout Management, Dialog Boxes.

Unit-VI **[12]** Max Marks:**20**

Deployment and Concurrency and Database Programming: JAR Files, Storage of Application Preferences, Service Loaders, Applets, Java Web Start, Threads, Interrupting Threads, Thread States, Thread Properties, Synchronization, Blocking Queues, Thread-Safe Collections, Callables and Futures, Executors, Synchronizers, Threads and Swing, The Design of JDBC, The Structured Query Language, JDBC Configuration, Working with JDBC Statements, Query Execution, Scrollable and Updatable Result Sets, Row Sets, Metadata.

References:

1. Cay S. Horstmann Core Java Volume I—Fundamentals (December 2015), Tenth Edition, Prentice Hall, ISBN: 9780134177335
2. Cay S. Horstmann Core Java, Volume II—Advanced Features (December 2016), Tenth Edition, Prentice Hall, ISBN: 9780134177878
3. Herbert Schildt, Java: The Complete Reference, Ninth Edition, McGraw Hill Education, ISBN 978-0-07-180855-2

Course Outcome:

Students will able to:

1. To understands the fundamentals of Java programming language and its constructs.
2. To understand concept of object-oriented programming concept using Java.
3. To implement the applications using the concept of the Inheritance, Interfaces, Lambda Expressions, and Inner Classes.
4. To design and implement the real-world application using the concept of the Exceptions and Generic Programming
5. To understand how to use concept of the Graphics Programming, Event Handling, Swing Components, and JDBC in their application.

Course Code: **CS LAB-I** **LAB on Data Structures and Algorithms and JAVA programming** *Total Marks:* **100**

Course Objectives:

The main objectives of this course are:

Data Structures and Algorithms

- Solve real-world problems by reasoning about data structure choices, choose appropriate implementations.

- Identify the strengths and weaknesses of different data structures
- To make the students write various programs and ADTS for all data structures.
- Students will learn to write, debug, and test large programs systematically.
- Think critically for improvement in solutions.
- Determine which algorithm or data structure to use in different scenarios.

JAVA programming

- Gain knowledge about basic Java language syntax and semantics to write Java programs and use concepts such as variables, conditional and iterative execution methods etc.
- Understand the fundamentals of object-oriented programming in Java, including defining classes, objects, files, invoking methods etc and exception handling mechanisms.
- Understand the principles of inheritance, packages and interfaces
- Using Swing library and various GUI components, Applet programming, JDBC, generic programming and multithreaded programming

Data Structures and Algorithms

1. Implementation of programs based on the following
    - Arrays
    - Multidimensional Arrays, Matrices
    - Stacks, Polish Notation
    - Queues
    - Deques
    - Linear Linked List, Circular Linked List, Doubly Linked List
    - Polynomial Addition/Subtraction
2. Implementation of programs based on Trees
    - Binary Search Tree
    - In-order, Pre-order and Post-order Traversals
    - Heap Tree
    - Balanced Binary Tree (AVL)
    - B-Trees
3. Implementation of programs based on Graphs
    - Depth First Traversal
    - Breadth First Traversal
    - Obtaining Shortest Path (Dijstra and Floyd-Warshall)
    - Minimum spanning tree (Kruskal and Prim)
4. Implementation of programs for Hash Table, Searching and Sorting techniques
    - Hash Table
    - Linear and Binary Search (using array)
    - Bubble sort
    - Selection sort
    - Insertion sort
    - Radix sort
    - Quick sort
    - Merge sort
    - Heap sort

JAVA programming

1. Write a program that demonstrates program structure of java.
2. Write a program that demonstrates string operations.
3. Write a program that demonstrates package creation and use in program.
4. Write a program that demonstrate inner class.
5. Write a program that demonstrates inheritance.
6. Write a program that demonstrates 2D shapes on frames.
7. Write a program that demonstrates text and fonts.
8. Write a program that demonstrates event handling for various types of events.
9. Write a program to illustrate use of various swing components.
10. Write a program that demonstrates use of dialog box.
11. Write a program to create own dialog box.
12. Write a program to create toolbar, menu & popup menu.
13. Write a program to implement file handlings.
14. Write a program that demonstrates Applet programming.
15. Write a program to implement generic programming.
16. Write a program that demonstrates JDBC on applet/application.
17. Write a program that demonstrates multithreading.

Course Outcome:

The above exercise shall make the students competent in the following ways and will be able to learn following parameters at the end of the course.

Data Structures and Algorithms

- Develop solutions for a range of problems using procedure oriented / object oriented programming.
- Apply divide and conquer strategy to searching and sorting problems using iterative And / or recursive solutions.
- Use critical thinking skills and creativity to solve the problems.
- Design scenarios to explain behaviors and demonstrate correctness of programs.
- Determine which algorithm or data structure to use in different scenarios.
- Choose the appropriate data structure and algorithm design method for a specified application.
- Have practical knowledge on the applications of data structures
- Identify classes, objects, members of a class and relationships among them needed for a specific problem

JAVA programming

- Write Java application programs using OOP principles and proper program structuring
- Implementing user interface: 2D shapes, events, dialog box, menu and popup menu
- Developing Applets, multithreaded programs
- Implementing generic and JDBC programming
- Demonstrate the concepts of polymorphism and inheritance
- Write Java programs to implement error handling techniques using exception handling

Course Code: **CS LAB-II** **LAB on DBMS** *Total Marks:* **100**

Course Objectives:

1) The course mainly concentrates on understanding of the fundamentals of Data Definition Language and Data Manipulation Languages.
2) To develop conceptual understanding of database management system
3) To understand how a real-world schema can be implemented
4) To educate students with different Database Languages.
5) Understand the needs of database processing and learn techniques for controlling the consequences of concurrent data access.

1. Creating database tables and using data types.
   - Create table
   - Modify table
   - Drop table
2. Practical Based on Data Manipulation.
   - Adding data with Insert
   - Modify data with Update
   - Deleting records with Delete
3. Practical Based on Implementing the Constraints.
   - NULL and NOT NULL
   - Primary Key Constraint
   - Foreign Key Constraint
   - Unique Constraint
   - Check Constraint
   - Default Constraint
4. Practical for Retrieving Data Using following clauses.
   - Simple select clause
   - Accessing specific data with Where
   - Ordered By
   - Distinct
   - Group By
5. Practical Based on Aggregate Functions.
   - AVG
   - COUNT
   - MAX
   - MIN
   - SUM
   - CUBE
6. Practical Based on implementing all String functions.
7. Practical Based on implementing Date and Time Functions.
8. Practical Based on implementing use of UNION, INTERSECTION, SET DIFFERENCE.
9. Implement Nested Queries & all types of JOIN operation.
10. Practical Based on performing different operations on a view.
11. Practical Based on implementing use of Procedures.
12. Practical Based on implementing use of Triggers
13. Practical Based on implementing Cursor.

14. ++++VB.NET, C#.NET, JAVA, D2K, etc.
15. Practical based on creating Data Reports.
16. Design entity relationship models for a business problem and develop a normalized database structure

Course Outcome:

After successful completion of this course, students will be able to

1.To understand Database design methodology.
2.Acquire knowledge in fundamentals of Database Management System.
3.Work with popular Database languages.
4.Realise various data models for Database and Write queries in SQL.
5. Familiar with basic database storage structures and access techniques.

Semester-II

Course Code: **CS-201** **Compiler Construction** *Clock Hours:* **60**
Total Marks: **100**

Course Objectives:

To cover the major topics in compiler design with emphasis on solving the problems encountered in designing a compiler regardless of the source language or the target machine.

Unit-I **[05]** Max Marks:**10**

Compiler structure: analysis-synthesis model of compilation, various phases of a compiler, tool-based approach to compiler construction.

Unit-II **[06]** Max Marks:**15**

Lexical analysis: Interface with input, parser and symbol table, token, lexeme and patterns, Difficulties in lexical analysis, Error reporting, Implementation, Regular definition, Transition diagrams, LEX.

Unit-III **[15]** Max Marks:**30**

Syntax analysis: CFGs, ambiguity, associativity, precedence, top down parsing, recursive descent parsing, transformation on the grammars, predictive parsing, bottom up parsing, operator precedence grammars, LR parsers (SLR, LALR, LR), YACC.

Syntax directed definitions: inherited and synthesized attributes, dependency graph, evaluation order, bottom up and top down evaluation of attributes, L- and S-attributed definitions.

Type checking: type system, type expressions, structural and name equivalence of types, type conversion, overloaded functions and operators, polymorphic functions.

Unit-IV **[10]** Max Marks:**15**

Run time system: storage organization, activation tree, activation record, parameter passing, symbol table, dynamic storage allocation.

Unit-V **[10]** Max Marks:**15**

Intermediate code generation: Intermediate representations, translation of declarations, assignments, control flow, Boolean expressions and procedure calls. Implementation issues.

Unit-VI **[10]** Max Marks:**15**

Code generation and instruction selection: issues, basic blocks and flow graphs, register allocation, code generation, dag representation of programs, code generation from dags, peep hole optimization, code generator generators, specifications of machine.

References:

1. Aho A.V., R. Sethi and J.D. Ullman. Compiler Principle, Techniques and Tools: Addison Wesley, ISBN 0-321-48681-1.
2. Barret, Couch. Compiler Construction Theory and Practice: Computer Science series, Asian Student Ed, ISBN 978-0574213358
3. Dhamdhere D.M. Compiler Construction Principle and Practice: McMillan India, ISBN 9780333904060
4. Gres D. Compiler Construction for Digital Computer: Wiley, ISBN 047132776X.
5. David Galles (2009). Modern Compiler Design: Pearson Education, ISBN 9788131709412

Course Outcome:

Understanding of basic structure of compiler, concepts and terminology in programming languages, lexical analysis, finite state techniques, scanner generator, parsing, kinds of parsers, designing lexical analyzer, scanner and parsers, principal ideas with intermediate code generation, optimizations.

Understanding of all concepts essential to design compiler in general for programming languages.

Course Code: **CS-202** **Mathematical Foundations of Computer Science** *Clock Hours:* **60**
Total Marks: **100**

Course Objectives:

1. To build the foundation of computer algorithms using mathematical base.
2. To introduce the concepts of induction to prove certain property is true for non-negative integers.
3. To introduce the concepts of recursive data types, linear recurrences, dived and conquer recurrences and solving recurrences.
4. To introduce the concept of generating functions and applications of principle of inclusion and exclusion.
5. To apply statistical measures on the data and represent it graphically.
6. To relate practical examples to the probability theory and probability distributions to build the foundation for machine learning.
7. To understand stochastic processes and apply Markov chain theory to relate real time problems.
8. To understand hidden markov model and Chapman-Kolmogorov equation for solving problems.

Unit-I **[15]** Max Marks:**25**

Induction and Recursion: Mathematical Induction, Strong Induction and Well Ordering, Recursive Definitions and Structural Induction, Recursive Algorithms, Program Correctness, The Towers of Hanoi, Merge Sort, Linear Recurrences, Divide-and-Conquer Recurrences, A Feel for Recurrences

Unit-II **[15]** Max Marks:**20**

Advance Counting Techniques: Recursive Relations, Solving Linear Recurrence Relations, Divide-and-Conquer Algorithms and Recurrence Relations, Generating Functions, Inclusion-Exclusion, Applications of Inclusion-Exclusion.

Unit-III **[05]** Max Marks:**10**

Statistics: Population and sample, parameters and statistics: definition, types: Descriptive and Inferential, applications, Descriptive Statistics: Mean, median, mode and standard deviation, variance, Graphical statistics

Unit-IV **[15]** Max Marks:**20**

Probability: Making decisions under uncertainty, Classical definition of Probability, Events and their Outcomes, Rules of Probability, Probability axioms, Random variables (discrete and continuous), Joint and Conditional probability, independence and Bayes theorem, Distribution of a random vector, Probability mass function, Probability density function and distribution function. Distributions: Binomial, Poisson, Uniform, Exponential, Normal.

Unit-V **[10]** Max Marks:**15**

Stochastic Processes: Definitions and classifications of Stochastic Processes, discrete and continuous Markov models, Hidden Markov Models, Chapman-Kolmogorov equation

References:

1. Kenneth H. Rosen, Discrete Mathematics and its Applications 6th Ed, Tata McGraw-Hill Publishing Company Limited, New Delhi, 2007 *ISBN* 10: 0070681880
2. Michael Baron (2014) Probability and Statistics For Computer Scientists Second Edition, CRC press. ISBN: 978-1-4822-1410-9
3. Bhat B.R, Srivenkatramana T and Rao Madhava K.S. (1996): Statistics: A Beginner's Text, Vol. I, New Age International (P) Ltd.
4. Croxton F.E, Cowden D.J and Kelin S (1973): Applied General Statistics, Prentice Hall of India *ISBN* 10: 0201089947 *ISBN* 13: 9780201089943
5. Hogg, Robert V. & Craig Allen T. (2008). Introduction to Mathematical Statistics, Pearson Education *ISBN* 0-02-978990-7
6. Goon A.M., Gupta M.K., Dasgupta. B. (2001), Fundamentals of Statistics, Volume I and II, World Press, Calcutta.
7. Ross, S. (2005). Introduction to Probability Models, (6th Ed. Academic Press). *ISBN* 978-0-12-375686-2
8. Medhi, J. (1994). Stochastic Processes, (2nd Ed. New Age Publisher) *ISBN* : 978-93-86286-48-2

Course Outcome:

Students are expected to know and be able to

1. demonstrate their understanding of and apply methods of mathematics in computer science to subsequent courses in algorithm design and analysis
2. identify, formulate, and develop solutions to computational challenges.

3. understand and solve a computational problem to meet desired needs within realistic constraints.
4. analyze the behavior of the data, model the data using statistical measures and represent it graphically on paper without using available computerized tools.
5. apply mathematical foundations, probability theory in the modeling and design of computational systems in a way that demonstrates comprehension of the tradeoffs involved in design choices.

Course Code: **CS-203** **Artificial Intelligence** *Clock Hours:* **60**
Total Marks: **100**

Course Objectives:

The primary objective of this course is to introduce the basic principles, techniques, and applications of Artificial Intelligence. Emphasis will be placed on the teaching of these fundamentals, not on providing a mastery of specific software or tools programming environments.

The student should be made to:

1) Gain a historical perspective of AI and its foundations.
2) Study the concepts of Artificial Intelligence.
3) Investigate applications of AI techniques in intelligent agents
4) Learn the methods of solving problems using Artificial Intelligence.
5) Learn various peculiar search strategies for AI

Unit-I **[08]** Max Marks:**10**

Introduction: Overview and Historical Perspective, Turing test, Physical Symbol Systems and the scope of Symbolic AI, Agents.

Unit-II **[06]** Max Marks:**10**

State Space Search: Depth First Search, Breadth First Search, DFID.

Unit-III **[08]** Max Marks:**12**

Heuristic Search: Best First Search, Hill Climbing, Beam Search, Tabu Search.

Unit-IV **[08]** Max Marks:**15**

Randomized Search: Simulated Annealing, Genetic Algorithms, Ant Colony Optimization.

Unit-V **[08]** Max Marks:**12**

Problem Decomposition: Goal Trees, AO*, Rule Based Systems, Rete Net.

Unit-VI **[06]** Max Marks:**12**

Game Playing: Minimax Algorithm, AlphaBeta Algorithm, SSS*.

Unit-VII **[08]** Max Marks:**14**

Planning and Constraint Satisfaction: Domains, Forward and Backward Search, Goal Stack Planning, Plan Space Planning, Graphplan, Constraint Propagation.

Unit-VIII **[08]** Max Marks:**15**

Logic and Inferences: Propositional Logic, First Order Logic, Soundness and Completeness, Forward and Backward chaining.

References:

1. Deepak Khemani (2013). A First Course in Artificial Intelligence, McGraw Hill Education (India), ISBN 9781259029981
2. Elaine Rich and Kevin Knight (1991). Artificial Intelligence, Tata McGraw Hill, ISBN 13: 9780070087705
3. Stuart Russell and Peter Norvig (2009). Artificial Intelligence: A Modern Approach, 3rd Edition, Prentice Hall, ISBN-13: 978-0-13-604259-4.

Course Outcome:

At the end of the course, the student should be able to:

1) Identify problems that are amenable to solution by AI methods.
2) Identify appropriate AI methods to solve a given problem.
3) Design smart system using different informed search / uninformed search or heuristic approaches.
4) Apply the suitable algorithms to solve AI problems.

***Course Code:* CS-204** **Design and Analysis of Algorithms** *Clock Hours:* **60**
Total Marks: **100**

Course Objectives:

To Understand and learn

1. Basic concepts of algorithms and analyze the performance of algorithms.
2. Algorithm design techniques for developing algorithms.
3. Searching and traversal algorithms for graphs.
4. Nondeterministic algorithms and NP class of problem.

Unit-I **[10]** Max Marks:**15**

Introduction: What Is An Algorithm?, Algorithm Specification, Pseudocode Conventions, Recursive Algorithms, Complexity, Asymptotic Notation, Practical Complexities And Performance Measurement

Tree And Graph Representations, Binary Trees Basics, Heaps And Heap Sort, Sets And Disjoint Set Union And Find.

Unit-II **[12]** Max Marks:**15**

Divide and Conquer: General Method, Binary Search, Finding Maximum and Minimum, Merge Sort, Quick Sort, Strassen's Matrix Multiplication.

Unit-III **[08]** Max Marks:**15**

The Greedy Method: General Method, Knapsack Problem, Huffman Code, Minimum-Cost Spanning Trees (Prim's & Kruskal's Algorithm), Optimal Storage On Tapes, Single-Source Shortest Paths.

Unit-IV **[08]** Max Marks:**15**

Dynamic Programming: General Method, All-Pair Shortest Path, Matrix Chain Multiplication, Longest Common Sub Sequence, 0/1knapsack, Flow Shop Scheduling

Unit-V **[06]** Max Marks:**15**

Basic Search and Traversal Techniques: Breadth First Search and Traversal, Depth First Search And Traversal, Spanning Trees.

Unit-VI **[08]** Max Marks:**15**

Backtracking: General Method, Constrains, 8-Queens Problem Graph Coloring

References:

1. Horowitz E. and Sahni S. “Fundamentals of computer Algorithms” Galgotia publications. ISBN:0716783169
2. Horowitz E., Sahni S. and Rajshekaran S(),Computer Algorithms, Computer Science Press, ISBN-10: 8173716129
3. S. Dasgupta, C. H. Papadimitriou, and U. V. Vazirani(2006), Algorithms. McGraw-Hill publications, ISBN 9780073523408
4. Cormen, Leiserson and Rivest, Introduction to Algorithms, Prentice Hall of India, ISBN: 978-81-203-4007-7

Course Outcome:

1. Analyze the asymptotic performance of algorithms.
2. Write rigorous correctness proofs for algorithms.
3. Design and analyze divide-and-conquer based algorithms.
4. Devise and Synthesize greedy and dynamic-programming based algorithms.
5. Employ graphs to model problems solvable using traversal techniques.
6. Able to model problems using backtracking
7. Able to classify nondeterministic polynomial time algorithms.

Course Code: **CS-205** **Python Programming** *Clock Hours:* **60**
Total Marks: **100**

Course Objectives:

The student should be able to

- understand the fundamental concepts of Python programming.
- learn that how python programming supports some constructs of functional programming.
- work with strings, lists, tuples, dictionaries, and files.
- define their own classes, methods and module for solving real world problems.
- use regular expression for searching patterns in given strings.

Unit-I **[12]** Max Marks:**20**

The Python Programming Language, Python Data, Variables, Expressions and Statements, Values and Data Types, Type conversion Functions, Operators and Operands, Input, Order of Operations, Functions, Calling Functions, Passing Functions, Formal Arguments, Variable-length Arguments, Functional Programming, Boolean Expressions, Logical operators, Precedence of Operators, Conditional Execution, Unary Selection, Nested conditionals, Chained conditionals, Boolean Functions, Iteration, The for loop, The while Statement

Unit-II **[08]** Max Marks:**15**

Strings, A Collection Data Type, Operations on Strings, Index Operator: Working with the Characters of a String, String Methods, Length, The Slice Operator, String Comparison, Lists, List Values, List Length, Accessing Elements, List Membership, Concatenation and Repetition, List Slices, Lists are Mutable, List Deletion, Objects and References, Aliasing, Cloning Lists, Repetition and References, List Methods, Append versus Concatenate Lists and for loops, Using Lists as Parameters, Nested Lists, Strings and Lists, List Type Conversion Function, Tuples, Tuple operators and built-in functions, Tuples and Mutability, Tuple Assignment, Tuples as Return Values

Unit-III **[15]** Max Marks:**20**

Dictionaries, Dictionary Operations, Dictionary Methods, Dictionary Keys, Aliasing and Copying, Sparse Matrices, Working with Data Files, Finding a File on your Disk, Reading a File, Iterating over lines in a file, Writing Text Files, Object Oriented Programming, Classes, Instances, Class method Calls, Coding Class Tree, Attributes, Building and Method Invocation, Composition, Inheritance, Operator Overloading, Encapsulation and Information Hiding, Search Algorithms, Sorting Algorithms, Hash Tables

Unit-IV **[10]** Max Marks:**20**

Regular Expressions, Exceptions, Standard Exceptions, Exceptions Syntax, The try/except/else Statement, The try/finally Statement, Unified try/except/finally, The raise Statement, The assert Statement, with/as Context Managers String-Based Exceptions, Class-Based Exceptions, General raise Statement Forms, Nesting Exception Handlers, Exception Idioms, Exception Design Tips. Catch All Exceptions, Catch A Specific Exception, Catch Multiple Specific Exceptions, Clean-up After Exceptions, GUI Programming using TKinter.

Unit-V **[15]** Max Marks:**25**

Advance Function Topics: Anonymous Function Lambda, Mapping Functions over Sequences: map, Functional Programming Tools: filter and reduce, List Comprehensions Revisited: Mappings. Modules: Python Program Architecture, Module Creation, Module usage, Module Namespaces, Reloading Modules, Module Packages. Data Hiding in Modules, Enabling Future Language Features, Mixed Usage Modes, Changing the Module Search Path, The import as Extension, Relative Import Syntax, Module Design Concepts

References:

1. John V Guttag (2013), Introduction to Computation and Programming Using Python, Prentice Hall of India, 2013, ISBN: 9780262525008
2. R. Nageswara Rao(2016), Core Python Programming, Dreamtech Press, 2016, ISBN-13: 9789351199427
3. Wesley J. Chun(2006), Core Python Programming - Second Edition, Prentice Hall, ISBN-13: 978-0132269933, ISBN-10: 0132269937

4. Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser(2013), Data Structures and Algorithms in Pyhon", Wiley, 2013, ISBN : 978-1-118-54958-2, ISBN : 978-1-118-29027-9(HardCover)
5. Kenneth A. Lambert(2011), Fundamentals of Python – First Programs, CENGAGE Publication, 2011, ISBN 1111822700, ISBN 9781111822705
6. Luke Sneeringer(2015), Professional Python, Wiley Inc.,2015, ISBN: 1119070856
7. Mark Lutz (2007), Learning Python, 3rd Edition, O'Reilly Media, Inc., 2007, ISBN-13: 978-0-596-51398- 6, ISBN-10: 0-596-51398-4

Course Outcome:

Upon completion of the subject, students will be able to:

- understand the basic concepts of Python programming.
- write Python programs that supports some constructs of functional programming like map, reduce, filter.
- understand the use of strings, lists, tuples, dictionaries, and files and able to manipulates data available within them with help of various functions.
- understand how to write user defined classes, methods as well as module creation and handle exceptions while implementing python programs.
- use regular expression for validating email address or domain name.

Course Code: **CS- LAB-III** **LAB on Design and Analysis of Algorithms (DAA)** *Total Marks:* **100**

Course Objectives:

Understand and learn

1. To convert the algorithms to code.
2. To measure the complexities at run time.
3. To modify the algorithms for efficiency.
4. To debug and test the programs.
5. To conclude using profile of outcomes.

OS: Windows/Linux, **Programming Language:** C++/Java/C#

1. Write a program for creating max./min. heap using
   - INSERT
   - ADJUST/HEAPIFY
2. Write a program to implement union and find operation.
3. Write a program to find minimum and maximum form a given array.
4. Write a program for searching element form given array using binary search for n=1000,2000,3000 find exact time of execution.
5. Write a program for sorting given array in ascending/descending order with n=1000,2000,3000 find exact time of execution using
   - Heap sort
   - Merge sort
   - Quick sort
6. Write a program for matrix multiplication using Strassen's matrix multiplication.
7. Write a program to find solution of Knapsack instant.

8. Write a program to find shortest path using single source shortest path.
9. Write a program to find Minimum-Cost Spanning Trees (Prim's & Kruskal's Algorithm).
10. Write a program to find shortest path using all pair path.
11. Write a program to find longest common subsequence.
12. Write a program to implement breadth first and depth first search.
13. Write a program to implement breadth first and depth first traversal.
14. Write a program to find all solutions for 8-queen problem using backtracking.

Course Outcome:

1. Able to construct logic for the algorithms designed using designing techniques.
2. Able to do posterior analysis of the algorithms.
3. Able to debug the algorithms.
4. Modify to improve performance of the algorithms.
5. Able to test and profile the algorithms.

Course Code: **CS-LAB-IV** **LAB on Python Programming** *Total Marks:* **100**

Course Objectives:

The student should be able to

- develop the Python programs for searching, sorting, with help of fundamental concepts like lists, dictionary.
- understand the concepts of functions scoping, recursion, list mutability, regular expression in Python programming.
- learn to define their own classes, methods and modules according to the requirement of the problem and use of exception handling concepts.
- define regular expression and develop GUI programs using Tkinter.

1. Develop programs to understand the control structures of python
2. Develop programs to learn different types of structures (list, dictionary, tuples) in python
3. Develop programs to learn concept of functions scoping, recursion and list mutability.
4. Develop programs to understand object oriented programming using python.
5. Develop programs for data structure algorithms using python – searching, sorting and hash tables.
6. Develop programs to learn regular expressions using python.
7. Develop programs to learn GUI programming using Tkinter.
8. Demonstrate the concept of exception handling using try/except/else Statement, Unified try/except/finally, try/finally Statement, raise Statement, assert Statement, catch multiple specific exceptions
9. Demonstrate the concept of String-Based Exceptions, Class-Based Exceptions and Nesting Exception handlers.
10. Demonstrate implementation of the Anonymous Function Lambda.
11. Demonstrate implementation Mapping Functions over Sequences.
12. Demonstrate implementation functional programming tools such as filter and reduce
13. Demonstrate the Module Creation, Module usage, Module Namespaces, Reloading Modules, Module Packages, Data Hiding in Modules.

14. Demonstrate Mixed Usage Modes of modules, Changing the Module Search Path, The import as Extension, Relative Import Syntax, Module Design Concepts

Course Outcome:

Upon completion of the subject, students will be able to:

- implement Python programs that demonstrates all types of sorting and searching techniques.
- write programs that demonstrate the concepts of functions scoping, recursion, list mutability, regular expression and support of function programming constructs through Python programming.
- write Python programs that defines user defined classes, methods and module for solving real world problems as well as use of exception handling concepts whenever necessary.
- implement programs that uses regular expression for searching patterns and validating data.
- develop GUI programs using Tkinter.

Semester-III

Course Code: **CS-301** **Machine Learning** *Clock Hours:* **60**
Total Marks: **100**

Course Objectives:

- To introduce students to the basic concepts and techniques of Machine Learning.
- To have a thorough understanding of the Supervised and Unsupervised learning techniques
- To understand regression, classification and clustering
- To study the various probability based learning techniques
- To understand graphical models of machine learning algorithms

Unit-I **[08]** Max Marks:**10**

Introduction: Basic definitions, types of learning, hypothesis space and inductive bias, evaluation, cross-validation

Unit-II **[08]** Max Marks:**15**

Linear regression, Decision trees, overfitting

Unit-III **[09]** Max Marks:**15**

Instance based learning, Feature reduction, Collaborative filtering based recommendation

Unit-IV **[08]** Max Marks:**15**

Probability and Bayes learning

Unit-V **[09]** Max Marks:**15**

Logistic Regression, Support Vector Machine, Kernel function and Kernel SVM

Unit-VI **[09]** Max Marks:**15**

Neural network: Perceptron, multilayer network, backpropagation, introduction to deep neural network

Unit-VII **[09**] Max Marks**:15**

Clustering: k-means, adaptive hierarchical clustering, Gaussian mixture model

References:

1. Tom Mitchell (1997). Machine Learning. First Edition, McGraw- Hill, ISBN 10: 0070428077 ISBN 13: 9780070428072
2. Ethem Alpaydin (2009). Introduction to Machine Learning, Edition 2, The MIT Press. ISBN 978-0-262-01243-0

Couse Outcome:

Upon completion of the course, the students will be able to:

- Distinguish between, supervised, unsupervised and semi-supervised learning
- Apply the apt machine learning strategy for any given problem
- Suggest supervised, unsupervised or semi-supervised learning algorithms for any given problem
- Design systems that uses the appropriate graph models of machine learning

Course Code: **CS-302** **Web Application Development Technology** *Clock Hours:* **60**
Total Marks: **100**

Course Objectives:

- To learn .Net Framework
- Creating ASP.Net web applications using standard .net controls.
- Develop database applications using ADO.Net
- Use Web Services and develop simple and complex applications using .Net framework
- Develop a data driven web application.
- Connecting to data sources and managing them.
- Maintain session and controls related information for user used in multi-user web applications
- Understand the fundamentals of developing modular application by using object oriented methodologies

Unit-I **[10]** Max Marks:**15**

Desktop Computing vs. Internet Computing, Internet computing infrastructure, Client side scripting vs. Server Side Scripting technologies, Web Server basics and configuration: IIS, Apache etc., Web site hosting basics, Web Publishing, HTML, introduction to .NET framework, Features of .NET frameowork:CTS,CLS,CLR,.NET technologies, languages'C#.NET,VB.NET, basics of ASP.NET page framework, Visual studio .NET IDE, Page Life Cycle, PostBack, Viewstate, Page directives, ASP.Net page execution cycle, HTTP Pipeline, HTTP Application, HTTP Request, HTTP Response classes, HTTP Modules and HTTP Handlers, State Management, Role of Global.asax, Application configuration using web.config file

Unit-II **[15]** Max Marks:**25**

ASP.NET Control hierarchy, HTML Server Controls, Web Server Controls, User and Server controls, Validation Controls, List bound controls: dropdown lists, list boxes, Repeater, DataList, Data Grid, DataGridView, FormsView controls, Data binding to List Bound Controls, Templating and Styling of ASP.NET server controls

Unit-III **[20]** Max Marks:**25**

Web Page Designing principles, CSS anatomy, Anatomy of Master Pages, nesting master pages, Site map file, Web site Navigation controls, properties:TreeView, Sitemap Path, Menu, Other Navigation methods: Response.Redirect(), Server.Transfer(), Personalization through Profiles, Themes/Skins, Web Site security basics: authentication modes:Windows,Forms,passport, authorization, roles/Membership, access rules, login controls,Web services: working, anatomy, hosting

Unit-IV **[15]** Max Marks:**25**

Database technology: ADO.NET, Anatomy/architecture of ADO.NET, working with Connection, Command, Data Adaptor, DataReader, DataSet, DataTable objects, Editing data in Data Tables, concurrency control. Introduction to MVC, Data Reports

References:

1. Richard Anderson, Brian Francis, Alex Homer, Rob Howard, David Sussman, Karli Watson(2002), Professional ASP.NET 1.0, Special Edition, Wrox Press Ltd., 2002, ISBN 1-861007-0-3-5.
2. Chris Hart, John Kauffman, Dave Sussman, and Chris Ullman(2006), Beginning ASP.NET 2.0, Wiley Publishing, Inc., 2006, ISBN-13: 978-0-7645-8850-1, ISBN-10: 0-7645-8850-8.
3. Beginning ASP.NET 4: in C# and VB, Imar Spaanjaars, Wiley Publishing, Inc 2010., ISBN: 978-0-470-50221-1
4. Bill Evjen, Scott Hanselman, Devin Rader (2008), Professional ASP .NET 3.5 in C# and VB, Wiley Publishing Inc.,2008 ISBN:978-0-470-18757-9.
5. Dino Esposito (2008), Programming Microsoft ASP.NET 3.5, Second Edition, Microsoft Press, 2008, ISBN-10: 0735625271, ISBN-13: 978-0735625273

Auxiliary Resources:

Website URLs

- https://www.asp.net/
- http://asp.net-tutorials.com/

Video Links

- https://www.asp.net/web-forms/videos
- https://www.youtube.com/playlist?list=PL6n9fhu94yhXQS_p1i-HLIftB9Y7Vnxlo&feature=view_all

Course Outcome:

The student will be able apply technical knowledge and perform specific technical skills, including:

- Successful students will able to design web applications using ASP.NET
- Successful students will be able to use ASP.NET controls in web applications.
- Successful students will be able to debug and deploy ASP.NET web applications
- Successful students will be able to create database driven ASP.NET web applications and web services.

Course Code: **CS-303** **Computer Graphics and Digital Image Processing** *Clock Hours:* **60** *Total Marks:* **100**

Course Objectives:

1. To equip students with the fundamental knowledge and basic technical competence in the field of Computer Graphics and Digital Image Processing.
2. To provide an understanding of how to scan convert the basic geometrical primitives, basic principles of 2 and 3- dimensional computer graphics.
3. To be able to discuss the application of computer graphics concepts in the development of information visualization, and business applications.
4. Give an in-depth knowledge about the basic theory and algorithms related to Digital Image Processing.

5. Provide awareness about the current technologies and issues specific to Digital Image Enhancement, Restoration, Segmentation, Color Image Processing, and Morphological Image Processing.

Unit-I **[08] Max Marks:12**

Introduction to Computer Graphics and Output Primitives: Overview of Computer Graphics, Its Applications and Software, Representation in Graphics, Vector Graphic Display, Raster Graphics Display, I/O Devices, Representing Image, Straight Line, Line Drawing Algorithm, DDA, Bresenham's Line Algorithm, Circle-generating algorithm, Ellipse-generating Algorithm, Polygon Filling Algorithm.

Unit-II **[06] Max Marks:12**

Two-Dimensional Transformation: Matrix and transformation, 2D Basic transformation, Homogeneous coordinates, Translation, Scaling and Rotation of straight line or polygon, Combined Transformation, Rotation about an arbitrary point/line, Reflection and Shearing Transformation, Viewing Transformation, Clipping, Cohen-Sutherland line clipping.

Unit-III **[06] Max Marks:12**

Three-Dimensional Transformation: Introduction, Matrix representation of 3D Transformation, 3D Translation, Scaling, Rotation, Composition of 3D Transformation, Projection, Orthographic, Isometric, Oblique Projection, Perspective Projection, One-Two-Three point perspective Projection.

Unit-IV **[08]** Max Marks:**12**

Introduction to Digital Image Processing & Applications: Digital Image Processing. Applications of Digital Image Processing, Fundamental Steps in Digital Image Processing, Components of an Image Processing System, Image Sensing and Acquisition. Image Sampling and Quantization. Some Basic Relationships Between Pixels.

Unit-V **[10] Max Marks:15**

Image Enhancement: Background, Some Basic Gray Level Transformations, Histogram Processing, Enhancement Using Arithmetic/Logic Operations, Basics of Spatial Filtering, Smoothing Spatial Filters, Sharpening Spatial Filters, Combining Spatial Enhancement Methods,

Introduction to the Fourier Transform and the Frequency Domain, Smoothing Frequency-Domain Filters, Sharpening Frequency Domain Filters, Homomorphic Filtering

Unit-VI **[10] Max Marks:15**

Image Restoration and Color Image Processing: A Model of the Image Degradation/Restoration Process, Noise Models, Restoration in the Presence of Noise Only-Spatial Filtering, Periodic Noise Reduction by Frequency Domain Filtering, Linear, Position-Invariant Degradations, Estimating the Degradation Function, Inverse Filtering, Minimum Mean Square Error (Wiener) Filtering,

Color Fundamentals, Color Models, Pseudocolor Image Processing, Basics of Full-Color Image Processing, Color Transformations, Smoothing and Sharpening.

Unit VII **[06] Max Marks: 12**

Morphological Image Processing & Segmentation: Detection of Discontinuities, Edge linking & Boundary Detection, Thresholding, Region based segmentation Laplacian of Gaussian, Derivative of Gaussian, Canny Edge Detection, Morphological operation: Dilation erosion, Opening & Closing, Basic Morphological Algorithm, Image representation schemes.

Unit VIII **[06]** **Max Marks: 10**

MATLAB Image processing toolbox: Introduction to MATLAB, Matrix Operations, Introduction to Image Processing Tool Box, Image Read & Write, Filters (spatial and frequency domain), Image Restoration and Reconstruction, Morphological Operations, Edge Detection and linking, Segmentation.

References:

1. Amarendra N Sinha, Arun D. Udai, (2008). Computer Graphics, TMH publication ISBN- 13 : 978-0-07-063437-4.
2. D. Hearn and M. Pauline Baker, Computer Graphics (C Version), Pearson Education, 2nd Edition ISBN-13: 978-0135309247
3. D. F. Rogers and J. A. Adams, Mathematical Elements for Computer Graphics, 2nd Edition, McGraw-Hill International Edition, ISBN-13:978-0-07-0486775
4. R.C.Gonzalez & R.E.Woods, Digital Image Processing, Pearson Education, 3rd edition, ISBN. 13:978-0131687288
5. S. Jayaraman Digital Image Processing TMH (McGraw Hill) publication, ISBN-13:978-0-07-0144798
6. Gonzalez, Woods & Steven, Digital Image Processing using MATLAB, Pearson Education, ISBN-13:978-0130085191

Course Outcome:

1. Developed scientific and strategic approach to solve complex problems Computer in the domain of Computer Graphics and Digital Image Processing.
2. Demonstrated various algorithms for scan conversion and filling of basic primitives objects and their comparative analysis and applied 2-D and 3-D geometric transformations, viewing and clipping on graphical objects.
3. Built the mathematical foundations for digital image representation, image acquisition, image transformation, image enhancement and restoration.
4. Developed a theoretical foundation of fundamental concepts of digital image processing.
5. Exposed students to MATLAB Image Processing Toolbox.

Course Code: **CS-304** **Software Engineering** *Clock Hours:* **60**
Total Marks: **100**

Course Objectives:

Students will try to learn:

- The nature of software development and software life cycle process models.
- Explain methods of capturing, specifying, visualizing and analyzing software requirements.
- 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.
- Understand need of project management and project management life cycle.
- Understand project scheduling concept and risk management associated to various type of projects.

Unit-I **[10]** Max Marks:**10**

Introduction and Process Models: Nature of Software, Software Engineering the process, Software Myths. Process Models: Generic process model, Prescriptive process models, Specialized process models, Unified process, Personal and Team process model, Process Technology, Product and Process. Agility, cost of change, Agile process, Extreme Programming, Agile Process models: Adaptive Software development, Scrum, Dynamic system development model, Crystal, Feature Driven development, Lean Software development, Agile modelling, Agile Unified process. Tool set for Agile process

Unit-II **[10]** Max Marks:**15**

Requirement Analysis and Modelling: Requirement Engineering, Establishing Groundwork, Eliciting Requirements Developing Use cases, Building Requirement model, Negotiating and Validating requirements. Requirement analysis, Scenario based modelling, UML models that supplements use case, Data modelling concepts, class based modelling. Requirement Modelling strategy, Flow oriented modelling, Creating Behaviour model, Pattern for Requirement modelling.

Unit-III **[08]** Max Marks:**15**

Quality Assurance and Change Management: Elements of SQA, SQA Tasks, Goal and Metrics, Formal approaches to SQA, Software Reliability, ISO 9000 Quality standards, SQA Plan. Software Configuration Management, SCM Repository, SCM process

Unit-IV **[11]** Max Marks:**20**

Design Concept: Design process, Design Concept: Abstraction, Architecture, Pattern, Separation of concept, Modularity, Information hiding, Functional independence, Refinement, Aspects, Refactoring. Design Model: Data design element, Architectural design element, Interface design element, Component level design element, Deployment level design element.

Unit-V **[11]** Max Marks:**20**

Architectural and Component Level Design: Software Architectures, Architectural Genres, Architectural styles, Architectural design, Accessing alternatives Architectural design, Architectural mapping using dataflow. Introduction to component, Designing class based component, Conducting component level design, Designing traditional component, component based development.

Unit-VI **[10]** Max Marks:**20**

Software Testing: Strategic approach to software testing, Test strategies for conventional software, Validation Testing, System testing, Software testing fundamentals, Internal and external view of testing, White box testing, Basic path testing, Control structure testing, Black box testing, model based testing, Testing for specialized Environment, Architectures and applications.

References:

1. R. S. Pressman, “Software Engineering: A Practitioner’s Approach”, McGraw-Hill International Edition, Seventh Edition, ISBN:978-007-126782-3.
2. Pankaj Jalote, “Software Engineering: A Precise Approach”, Wiley India Pvt. Limited ISBN: 978-81-265-2311-5.
3. K. K. Aggarwal and Yogesh Singh, “Software Engineering”, Third Edition, New Age International Publishers, ISBN:978-81-224-2360-0.

Course Outcome:

Students will able to:

- Understand and demonstrate basic knowledge in software engineering
- Define various software application domains and remember different process model used in software development.
- Explain needs for software specifications also they can classify different types of software requirements and their gathering techniques.
- Convert the requirements model into the design model and demonstrate use of software and user interface design principles.
- Distinguish among SCM and SQA and can classify different testing strategies and tactics and compare them.
- Justify role of SDLC in Software Project Development
- Generate project schedule and can construct, design and develop network diagram for different type of Projects.

Course Code: **CS-305 (A)** **Natural Language Processing** *Clock Hours:* **60**
Total Marks: **100**

Course Objectives:

i. The prime objective of this course is to introduce the students to the field of Language Computing and its applications ranging from classical to modern context.
ii. Course also aims to provide understanding of various NLP tasks and NLP abstractions such as Morphological analysis, POS tagging, concept of syntactic parsing, semantic analysis etc.
iii. Course provide knowledge of different approaches/algorithms for carrying out NLP tasks.
iv. Course also discusses concepts of Language grammar and grammar representation in Computational Linguistics.

Unit-I **[08]** Max Marks:**12**

Introduction to NLP, brief history, NLP applications: Speech to Text(STT), Text to Speech(TTS), Story Understanding, NL Generation, QA system, Machine Translation, Text Summarization, Text classification, Sentiment Analysis, Grammar/Spell Checkers etc., challenges/Open Problems, NLP abstraction levels, Natural Language (NL) Characteristics and NL computing approaches/techniques and steps, NL tasks: Segmentation, Chunking, tagging, NER, Parsing, Word Sense Disambiguation, NL Generation, **Web 2.0**

Applications : Sentiment Analysis; Text Entailment; Cross Lingual Information Retrieval (CLIR).

Unit-II **[12]** Max Marks:**16**

Text Processing Challenges, Overview of Language Scripts and their representation on Machines using Character Sets, Language, Corpus and Application Dependence issues, Segmentation: word level (Tokenization), Sentence level. Regular Expression and Automata

Morphology, Types, Survey of English and Indian Languages Morphology, Morphological parsing FSA and FST, Porter stemmer, Rule based, and Paradigm based Morphology, Human Morphological Processing, Machine Learning approaches

Unit-III **[12]** Max Marks:**18**

Word Classes ad Part-of-Speech tagging (POS), survey of POS tagsets, Rule based approaches (ENGTOWL), Stochastic approaches(Probabilistic, N-gram and HMM), TBL morphology, unknown word handling, evaluation metrics: Precision/Recall/F-measure, error analysis

Unit-IV **[15]** Max Marks:**22**

NL parsing basics, approaches: Top Down, Bottom Up, Overview of Grammar Formalisms: constituency and dependency school, Grammar notations CFG, LFG, PCFG, LTAG, Feature-Unification, overview of English CFG, Indian Language Parsing in Paninian Karaka Theory, CFG parsing using Earley's and CYK algorithms, Probabilistic parsing

Unit-V **[15]** Max Marks:**22**

Concepts and issues in NL, Theories and approaches for Semantic Analysis, Meaning Representation, word similarity, Lexical Semantics, word senses and relationships, WordNet (English and IndoWordnet), Word Sense Disambiguation: Lesk Algorithm Walker's algorithm, Coreferences Resolution: Anaphora, Cataphora

References:

1. Indurkhya, N., & Damerau, F. J. (Eds.). (2010). *Handbook of Natural Language Processing, 2nd Edition.* New York: CRC Press Taylor and Francis Group, Boca Raton London, New York. ISBN-10: 1420085921, ISBN-13: 978-1420085921
2. Martin, J. H., & Jurafsky, D.(2013), *Speech and Language Processing*, Pearson Education India; 2 edition, ISBN-10: 9332518416, ISBN-13: 978-9332518414
3. Manning, Christopher and Heinrich, Schutze(1999), Foundations of Statistical Natural Language Processing", MIT Press, ISBN-10: 0262133601, ISBN-13: 978-0262133609.
4. Akshar Bharati, Chaitanya, V., Kulkarni, A., & Sangal, R. (July 1997). *Machine translation in Stages* (Vol. 10 no. 3). Mumbai: NCST, Mumbai.
5. Bharati, A., Chaitanya, V., & Sangal, R. (1995). *Natural Language Processing: A Paninian Perspective,* New Delhi: Prentice Hall of India, ISBN 10: 8120309219, ISBN 13: 9788120309210.
6. Steven Bird, Edward Loper (2016),Natural Language Processing With Python, Ed. 2, O'Reilly Media,ISBN 1491913428, 9781491913420

Auxiliary Resources:

a. Web Links
   1. https://see.stanford.edu/Course/CS224N
   2. https://web.stanford.edu/~jurafsky/NLPCourseraSlides.html

b. Video Links
   1. http://www.nptelvideos.in/2012/11/natural-language-processing.html
   2. https://www.youtube.com/playlist?list=PL6397E4B26D00A269

Course Outcome:

i. Students will get idea about know-hows, issues and challenge in Natural Language Processing and NLP applications and their relevance in the classical and modern context.

ii. Student will get understanding of Computational techniques and approaches for solving NLP problems and develop modules for NLP tasks and tools such as Morph Analyzer, POS tagger, Chunker, Parser, WSD tool etc.

iii. Students will also be introduced to various grammar formalisms, which they can apply in different fields of study.

iv. Students can take up project work or work in R&D firms working in NLP and its allied areas

*Course Code:***CS-305 (B)** **Optimization Algorithms** *Clock Hours:***60**

Total Marks: **90**

Course Objectives:

1. To introduce with the branch of OR and its role in decision making.
2. To list out various types of applications of operations research (OR).
3. To explain Linear Programming Problem (LPP) and practice with techniques to solve various types of LPP (transportation problem, assignment problems, special cases of duality, Integer programming problems)
4. Describe the significance, concept of game theory and algorithms to solve game theory problems.
5. Introduce critical path analysis using network problems.

Unit-I **[05]** Max Marks:**08**

Overview of operations Research: Introduction, Applications, Role of OR in Decision Making, Feasible and optimal Solutions

Unit-II **[15]** Max Marks:**20**

Linear Programming: Special Types: Transportation Problem as LPP, Initial Basic Feasible Solution, North West corner Rule, Lowest Cost Method, Vogel’s Approximation Method, MoDi method for optimization, Degeneracy.

Assignment problem, Hungarian Method, Special cases of assignment problem

Unit-III **[18]** Max Marks:**24**

Linear Programming Problems: Introduction, Formulation of Mathematical model of LPP, Standard form of linear programming problems, Solving LPP using Graphical method, Infeasible LPP, Unbounded LPP, Basic feasible solutions, Simplex method for solving LPP, augmentation using Slack and artificial variables, Big M and two phase method, Degeneracy, alternative optima, Interpretation of final Simplex table, Duality: concept, applications and example.

Unit-IV **[06]** Max Marks:**08**

Integer Programming: Introduction, How it differs from LPP, Pure and mixed integer programming problems, Binary IPP, Techniques to solve IPP.

Unit-V **[08]** Max Marks:**15**

Network Models: Definitions, Applications, Representation of a problem in network form, Critical Path Analysis, Resource planning, Giantt Chart.

Unit-VI **[08]** Max Marks:**15**

Game Theory : Concept, Two party zero sum game, Pay off matrix, Pure and mixed strategy games, Rule of Dominance, Subgame method, Brown's Algorithm

References:

1. Hamdy Taha (2010). Operations Research: An Introduction. Pearson Education. ISBN: 978-0132555937
2. L C Jhamb. Quantitative Techniques For Managerial Decisions Vol I, Vol II. Everest Publishing House, ISBN: 8186314628
3. PanneerSelvan R (2006). Operations Research. Prentice Hall of India. ISBN: 978-8120329287

Course Outcome:

After completion of this course students shall be able to-

1. write about OR and decision making.
2. Differentiate between feasible and optimal solution
3. Apply solving techniques to all types of LPP.
4. Apply solving techniques to network problems and game theory problems as well.

Course Code: **CS-305 (C)**	**Data Warehousing and Data Mining (DWDM)**	*Clock Hours:* **60** *Total Marks:* **100**

Course Objectives:

1. To comprehend evolution of decision making, operational vs decision support system and the concept of data warehouse.
2. To understand transactional and analytical processing
3. Significance of analytical processing and importance of data pre-processing.
4. Learn various data pre-processing techniques, methods.
5. Understand and apply various techniques/algorithms to obtain meaningful patterns from data (Association mining, classification and clustering)

Unit-I **[08]** Max Marks:**12**

Introduction to Data Warehousing: Evolution of decision system, Failure of past decision support system, Operational v/s decision support systems, Data warehousing lifecycle, Architecture, Building blocks, Components of DW, Data Marts and Metadata

Unit-II **[08]** Max Marks:**12**

Data Pre-processing: need for pre-processing of the data, Descriptive data summarization, Data cleaning, Data Integration and transformation, Data reduction, Data discretization and concept hierarchy generation.

Unit-III **[10]** Max Marks:**15**

OLAP Analytical Processing: OLAP in Data warehouse, Demand for online analytical processing, need for multidimensional analysis, limitations of other analysis methods, OLAP definitions and rules, OLAP characteristics, major features and functions, OLAP models-ROLAP, MOLAP, HOLAP, Differentiation, Data cubes and operations on cubes.

Unit-IV **[06]** Max Marks:**09**

Data Mining: Introduction-Data Mining functionalities, Classification of Data Mining Systems, basic Data Mining task, Data Mining Issues

Unit-V **[08]** Max Marks:**12**

Association Rule Mining: Efficient and Scalable Frequent Item set Mining Methods – Mining Various Kinds of Association Rules – Association Mining to Correlation Analysis – Constraint-Based Association Mining.

Unit-VI **[10]** Max Marks:**15**

Classification and Prediction : Issues Regarding Classification and Prediction – Classification by Decision Tree Introduction – Bayesian Classification – Rule Based Classification – Classification by Back propagation – Support Vector Machines – Associative Classification – Lazy Learners – Other Classification Methods – Prediction – Accuracy and Error Measures – Evaluating the Accuracy of a Classifier or Predictor – Ensemble Methods – Model Section.

Unit-VII **[10]** Max Marks:**15**

Cluster Analysis : Types of Data in Cluster Analysis – A Categorization of Major Clustering Methods – Partitioning Methods – Hierarchical methods – Density-Based Methods – Grid-Based Methods – Model-Based Clustering Methods – Clustering High-Dimensional Data – Constraint-Based Cluster Analysis – Outlier Analysis

References:

1. Jiawei Han and MichelineKamber "Data Mining Concepts and Techniques" Second Edition, Elsevier, Reprinted 2008.
2. M. H. Dunham. Data Mining: Introductory and Advanced Topics. Pearson Education. 2001.
3. H. Witten and E. Frank. Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann. 2000.
4. D. Hand, H. Mannila and P. Smyth. Principles of Data Mining. Prentice-Hall. 2001
5. Tan Steinbach, Vipin Kumar, Introduction to Data mining, Pearson Eduction
6. Jarke Vassiliou, Fundamentals of Data Warehouses, IInd Edition, Springer
7. Anahory Murray, Data Warehousing in Real World, Pearson Education
8. Paulraj Ponniah , Data Warehousing.

Course Outcomes:

After this course students shall be able to –

1. Explain organization of data warehousing and data marts.
2. Differentiate between OLTAP and OLAP
3. Apply data pre-processing techniques
4. Write basic algorithms for extracting patterns from data (association mining, classification and clustering)
5. Solve problems related with various aspects of data mining.

Course Code: **CS LAB-V** **LAB on Web Application Development Technology** *Total Marks:* **100**

Course Objectives:

i. Students will understand the working of Internet, Types of Web Sites/applications, basics of Web hosting and working of IIS web server.
ii. Students will get practical hands-on experience on Microsoft ASP.NET Web Application Development Technology and required Programming Language (C#.Net/VB.NET)
iii. Basic hands on the C#.Net/VB.NET programming language.

iv. Students will practically understand actual working of the theoretical concepts.
v. Students
vi. Students will undertake Project Work and its Demonstration in Viva-voce.

1. Demonstrate followings in IIS:
    a. Creation of Virtual Directory, Home directory, Home page, hosting of website
2. Demonstrate Page Life Cycle of ASP.NET. Use important page events for your demonstration.
3. Write VB.Net/C# console applications to demonstrate: OO concepts: polymorphism, encapsulation, inheritance, interface inheritance, abstract classes/methods, overloading, overriding, collection classes, properties
4. Demonstrate concept of postback and viewstate using web form server controls of ASP.NET
5. Demonstrate various Web form server controls using sample data entry screen form for registering for a service on website. Also use validation controls to validate input data.
6. Demonstrate DropDown List box, CheckButtonList, RadioButtonList controls.
7. Demonstrate Databinding using Hashtable, ArraryList, DataTable data sources.
8. Demonstrate Repeater control with the help of various templates.
9. Demonstrate paging, sorting, filtering of data in asp:DataGrid/DataGridView.
10. Demonstrate editing process in DataGrid and DataList controls. Make use of necessary templates for proper visual appearance.
11. Demonstrate State Management features of ASP.NET using sample shopping cart application.
12. Create sample website for demonstrating use of Profiles/Themes using skin files.
13. Demonstrate Master Pages and website navigation controls(sitemap path, treeview, menu) using SiteMap file.
14. Demonstrate Properties of website navigation controls.
15. Demonstrate Authorization/Authentication using Login controls and Roles/Membership/AccessRules
16. Demonstrate creation of simple/complex DataReader/DataSet Objects.
17. Demosntrate editing in DataTable objects.
18. Demonstrate Web Service hosting, access in ASP.NET

Course Outcome:

i. Students will get hands-on experience on basic concepts in web applications development using ASP.NET technology.
ii. Students can develop or undertake professional looking real life web sites using ASP.Net technology.
iii. It will help students to grasp other Web Application Development technologies/platforms easily through learn-by-comparison approach so that the learning curve will be smooth and faster.

Course Code: **CS LAB-VI** **LAB on Computer Graphics and Digital Image Processing** *Total Marks:* **100**

Course Objectives:

1. The student will gain a deeper knowledge about a chosen field of computer graphics and image processing while working on one of the more complex projects solved in the Laboratory.
2. To implement line, circle and ellipse drawing algorithms and 2 and 3-dimensional geometric transformations using C++.
3. To be able to design and develop the programs for viewing and clipping on graphical objects.
4. To introduce MATLAB to implement the complex algorithms of Digital Image Processing.
5. Provide hands-on experience to process digital images and expose students to MATLAB Image Processing Toolbox for Digital Image Enhancement, Restoration, Segmentation, Color Image Processing, and Morphological Image Processing.

1. Line drawing algorithm (DDA and Bresenham's Line Algorithm)
2. Circle drawing algorithm
3. Ellipse drawing algorithm
4. Polygon filling algorithm
5. Windowing and clipping algorithm (Point, line and polygon clipping)
6. Composite 2-D transformation, (rotation, scaling & reflection)
7. 3-D geometric transformation (rotation, scaling & reflection)
8. Introduction to Image Processing Toolbox
9. Read an 8 bit image and then apply different image enhancement techniques:
   - Brightness improvement
   - Brightness reduction
   - Thresholding
   - Negative of an image
   - Log transformation
   - Power Law transformation.
10. Implement different interpolation techniques using MATLAB/ SciLab
11. Read an image, plot its histogram then do histogram equalization. Comment about the result.
12. Read an image and apply
    - Implement Gray level slicing (intensity level slicing) in to read cameraman image.
    - Read an 8 bit image and to see the effect of each bit on the image.
    - Read an image and to extract 8 different planes i.e. 'bit plane slicing."
13. Implement various Smoothing spatial filter.
14. Read an image and apply
    - Gaussian 3x3 mask for burring
    - High pass filter mask with different masks
    - Laplacian operator with centre value positive and negative
    - High boost filtering.
15. Write a program to implement various low pass filters and high pass filter in frequency

domain.

16. Write a program for erosion and dilation, opening & closing using inbuilt and without inbuilt function.
17. Implement and study the effect of Different Mask (Sobel, Prewitt and Roberts)
18. Implement various noise models and their Histogram
19. Implement inverse filter and wiener filter over image and comment on them

Course Outcome:

1. Developed scientific and strategic approach to solve complex problems Computer in the domain of Computer Graphics and Digital Image Processing using C++ and MATLAB respectively.
2. Implemented various algorithms for scan conversion and filling of basic primitives objects and their comparative analysis and applied 2-D and 3-D geometric transformations, viewing and clipping on graphical objects.
3. Exposed students to MATLAB and Image Processing Toolbox.
4. Used various tools in MATLAB to implemented image transformation, image enhancement in spatial and frequency domain.
5. Developed the programs on various digital image processing techniques.

Semester-IV

Course Code: **CS-401** **Full Time Industrial Training** *Total Marks:* **300**

Course Objectives:

- To provide comprehensive learning platform to students where they can enhance their employ ability skills and become job ready along with real corporate exposure.
- To enhance students' knowledge in one technology.
- To increase self-confidence of students and helps in finding their own proficiency.
- To cultivate student's leadership ability and responsibility to perform or execute the given task.
- To provide learners hands on practice within a real job situation.

Twelve credits shall be awarded to the Industrial Training/Project course, which will commence in the IVth Semester and the final work and report will be completed at the end of IVth Semester of M. Sc. (Computer Science). The student is expected to work on software development project. The project work should have coding part. Student will have to submit the bound project report in university prescribed format at the end of the semester. Student will have to appear for Project Viva-voce and the marks and the credits will be allotted at the end of IVth semester of M. Sc. (Computer Science).

Course Outcomes:

- Capability to acquire and apply fundamental principles of Computers Science.
- Become master in one's specialized technology.
- Become updated with all the latest changes in technological world.
- Ability to communicate efficiently.
- Knack to be a multi-skilled Computer Science professional with good technical knowledge, management, leadership and entrepreneurship skills.
- Ability to identify, formulate and model problems and find engineering solution based on a systems approach.
- Capability and enthusiasm for self-improvement through continuous professional development and life-long learning