

# **SYLLABUS FOR M.Sc. COMPUTER SCIENCE**

**(2022-2024)**



**DEPARTMENT OF COMPUTER SCIENCE**

**MAHARAJA SRIRAM CHANDRA BHANJA DEO UNIVERSITY**

**(ERSTWHILE NORTH ORISSA UNIVERSITY)**

**SRIRAM CHANDRA VIHAR, TAKATPUR, BARIPADA,  
ODISHA, PIN-757003**

## **Vision**

---

To be a Premier Centre of Learning in providing high quality education and research in Computer Science and to foster the Graduates into globally competent professionals with expertise in software development which contribute to the society by harnessing technological skills in finding solution to the challenges of the ever-changing world.

## **Mission**

---

- To achieve academic excellence by imparting in-depth computational knowledge to the students through effective programming and hands on experience on latest tools and technologies.
- To train the students with cutting edge technological advance and make them industry-ready and successful computer professionals.
- To pursue interdisciplinary research skill that will fulfil the needs of the entire scientific community.
- To Inculcate professional behavior, strong ethical values, innovative research capabilities and leadership abilities for the societal issues and challenges of the 21st century.
- To Impart high-quality problem-solving approaches and team building skills and promote lifelong learning with a sense of societal and ethical responsibilities.

# Programme Objectives

---

Upon successful completion of the programme M.Sc. Computer Sc, students will be able to:

- Creatively solve problems, communicate effectively, and successfully function in diverse and inclusive multi-disciplinary teams.
- Demonstrate proficiency in the analysis of complex problems and the synthesis of solutions to those problems.
- Apply principles and practices of computing grounded in mathematics and science to successfully complete software-related projects to meet customer business objectives and/or productively engage in research.
- Apply their knowledge and skills to succeed in their careers and/or obtain advanced degrees.

## **Program Outcomes (POs)**

---

**PO1.** Develops an understanding of scientific theory principles and perspectives in sciences by critical thinking.

**PO2.** Develops problem solving skills and can design and carry out innovative research projects.

**PO3.** Communicates effectively, comprehends knowledge, writes effective reports, designs documentation, and makes effective presentations.

**PO4.** Functions effectively as an individual, as a member and leader of diverse teams in multidisciplinary settings for Holistic development.

**PO5.** Applies ethical principles and is committed to professional ethics, responsibilities in the field of research, can design, analyze, interpret data and find solutions for complex problems by applying the right tools. This study provides an excellent bridge between undergraduate study and Ph.D. research.

**PO6.** Realizes and promotes environmental sustainability through various eco-friendly measures that encourage judicious use of resources.

**PO7.** Postgraduate studies boost the self-directed career progress and outline the career paths. It improves the ability to tackle complex and challenging assessment tasks and helps in lifelong learning to be globally competent.

**PO8.** Takes up responsibilities in production, quality testing, designing, and marketing which contribute to the growth of industry and thus increases employability.

# Program Specific Outcomes (PSOs)

---

Upon completion of these courses the student would

**PSO1.** Provides technology-oriented students with the knowledge and ability to develop creative solutions.

**PSO2.** Develop skills to learn new technology.

**PSO3.** Apply computer science theory and software development concepts to construct computing-based solutions.

**PSO4.** Design and develop computer programs/computer-based systems in the areas related to algorithms, networking, web design, cloud computing, Artificial Intelligence, Mobile applications.

**PSO5.** Engage in professional development in the fields of Information Technology and Computer Science.

**PSO6.** Know about computing principles and business practices employed as software solutions in industries.

# CONTENTS

<b>Semester</b>	<b>Pages</b>
<b>Overall Structure</b>	<b>7-8</b>
<b>First Semester</b>	<b>9-23</b>
<b>Second Semester</b>	<b>24-40</b>
<b>Third Semester</b>	<b>41-57</b>
<b>Fourth Semester</b>	<b>58-64</b>
<b>Scheme of Examinations</b>	<b>65</b>

# M.Sc. Computer Science

## FIRST YEAR

### Semester-1<sup>st</sup>

Paper Code	Subject	Title of the Paper	Credit	Max. Marks		
				Internal	Univ. Exam	Total
CS-401	Core	Advanced C Programming	4	20	80	100
CS-403	Core	Advanced Computer Organization & Architecture	4	20	80	100
CS-405	Core	Advanced Data Structure & Algorithm	4	20	80	100
CS-407	Core	Discrete Mathematics	4	20	80	100
CS-409	Core	Advanced Operating System	4	20	80	100
CS-411	Practical	Advanced C Programming Lab	4			100
CS-413	Practical	Advanced Data Structure & Algorithm Lab	4			100
<b>Total</b>			<b>28</b>	<b>Total Marks</b>		<b>700</b>

### Semester-2<sup>nd</sup>

Paper Code	Subject	Title of the Paper	Credit	Max. Marks		
				Internal	Univ. Exam	Total
CS-402	Core	Theory of Computation	4	20	80	100
CS-404	Core	Python Programming	4	20	80	100
CS-406	Core	Advanced Computer Networks	4	20	80	100
CS-408	Core	Soft Computing	4	20	80	100
CS-410	Elective-I	Artificial Intelligence or Information Security & Cryptography or Wireless Sensor Network or Mobile Computing	4	20	80	100
OE-CS-412	Open Elective	MATLAB Programming	5	20	80	100
CS-414	Practical	Soft Computing Lab	4			100
CS-416	Practical	Python Programming Lab	4			100
<b>Total</b>			<b>33</b>	<b>Total Marks</b>		<b>800</b>

## SECOND YEAR

### Semester-3<sup>rd</sup>

Paper Code	Subject	Title of the Paper	Credit	Max. Marks		
				Internal	Univ. Exam	Total
CS-501	Core	Internet of Things	4	20	80	100
CS-503	Core	Advanced JAVA Programming	4	20	80	100
CS-505	Core	Compiler Design	4	20	80	100
CS-507	Core	Advanced Database System and Implementation	4	20	80	100
CS-509	Elective-II	Machine Learning or Big Data Analytics or Data Mining or Introduction to Cyber Security	4	20	80	100
CS-511	Practical	Advanced JAVA Programming Lab	4		100	100
CS-513	Practical	Advanced Database Lab	4		100	100
<b>Total</b>			<b>28</b>		<b>Total Marks</b>	<b>700</b>

### Semester-4<sup>th</sup>

Paper Code	Subject	Title of the Paper	Credit	Max. Marks		
				Internal	Univ. Exam	Total
CS-502	Core	Advanced Software Engineering	4	20	80	100
CS-504	Elective-III	Introduction to Deep Learning or Pattern Recognition or Cloud Computing or Digital Image Processing	4	20	80	100
CS-506		Dissertation & Viva Voce	8			100
<b>Total</b>			<b>16</b>		<b>Total Marks</b>	<b>300</b>
<b>Total Credits</b>			<b>105</b>		<b>Overall Total Marks</b>	<b>2500</b>



## FIRST YEAR

### Semester-1<sup>st</sup>

Paper Code	Subject	Title of the Paper	Credit	Max. Marks		
				Internal	Univ. Exam	Total
CS-401	Core	Advanced C Programming	4	20	80	100
CS-403	Core	Advanced Computer Organization & Architecture	4	20	80	100
CS-405	Core	Advanced Data Structure & Algorithm	4	20	80	100
CS-407	Core	Discrete Mathematics	4	20	80	100
CS-409	Core	Advanced Operating System	4	20	80	100
CS-411	Practical	Advanced C Programming Lab	4			100
CS-413	Practical	Advanced Data Structure & Algorithm Lab	4			100
<b>Total</b>			<b>28</b>	<b>Total Mark</b>		<b>700</b>

# SEMESTER -I

CS-401

Advanced C Programming

Credit - 04

## Course Objective

- To help students equip with the required knowledge and skill of Structured programming approach needed for the development of robust applications.
- To enable students to get hands-on experience on file handling.
- To provide insight to students on memory handling ability through Array and Pointer.
- To understand the advance test cases for handling real world projects.

## Course Outcome

- Able to explain a problem-solving process using C Language.
- Able to develop modular programming using looping concepts and functions.
- Able to analyze and solve complex real-world projects by developing application programs.
- Able to develop animation contents using graphics, memory and file handling functions.

## UNIT-I

**C Language Basics:** Identifiers, Control Structures, Decision making, Branching, Looping

**Functions:** Modular programs, Predefined functions, User-defined functions, Formal vs Actual arguments, Function definition, Function prototype, Function call, Parameter passing, Recursion, Storage Classes – auto, extern, register, static, scope of variables.

## UNIT-II

**Array and String:** Declaration, Initialization, One-dimensional and Multi-dimensional array, Array using functions, String-handling functions

**Pointer:** Declaration of Pointer variables, accessing data through pointers, Pointer Arithmetic, Passing parameters by Reference, Array of pointers, memory allocation functions – malloc(), calloc(), realloc() and free().

## UNIT-III

**Structure and Union:** Declaration of structure, pointer to structure, pointer to function, arrays of structure, arrays within structure, unions

**Preprocessors Directives:** Type of preprocessor directives, use of Macro

## UNIT-IV

**File Handling:** Concepts of file management, File handling functions – fopen(), fclose(), fprintf(), fscanf(), fseek(), ftell(), rewind(), putc(), getc(), putw(), getw(), Error handling functions, Command line arguments.

**Graphics:** Graphics library, Graphics handling functions, Use of colors in graphics.

## **Textbooks**

1. Brian W. Kernighan and Dennis Ritchie, “The C Programming Language”, 2<sup>nd</sup> Edition, Pearson Publisher, 2015, ISBN: 978-9332549449.
2. Herbert Schildt, “C: The Complete Reference”, 4<sup>th</sup> Edition, McGraw Hill Education, 2017, ISBN: 978-0070411838.

## **Reference Books**

1. Ashok N. Kamthane, “Programming in C”, 3<sup>rd</sup> Edition, Pearson Publisher, 2015, ISBN: 978-9332543553.
2. E. Balagurusamy, “Programming in ANSI C”, 8<sup>th</sup> Edition, McGraw Hill Education, 2019, ISBN: 978-9351343202.
3. Y. Kanetkar, “Let Us C: Authentic Guide to C Programming”, 18<sup>th</sup> Edition, BPB Publisher, 2021, ISBN: 978-9391392994.
4. Byron Gottfried, “Schaum’s Outlines: Programming with C”, 4<sup>th</sup> Edition, McGraw Hill Education, 2018, ISBN: 978-9353160272.

## **Online Sources**

1. <https://nptel.ac.in/courses/106105171>
2. <https://nptel.ac.in/courses/106104128>
3. [https://onlinecourses.nptel.ac.in/noc22\\_cs101/preview](https://onlinecourses.nptel.ac.in/noc22_cs101/preview)
4. <https://www.programiz.com/c-programming>
5. <https://www.javatpoint.com/c-programming-language-tutorial>
6. <https://www.tutorialspoint.com/cprogramming/index.htm>
7. <https://www.geeksforgeeks.org/c-programming-language/>

**Course Objective**

- To enable students to get know about the Parallelism concepts in Programming
- To give the students an elaborate idea about the different memory systems and buses.
- To introduce the advanced processor architectures to the students.
- To make the students know about the importance of multiprocessor and multicomputer.
- To study about various data flow computer architectures

**Course Outcome**

- Apply the concepts of parallelism in hardware/software.
- Discuss about various memory hierarchy, memory organization and mapping techniques.
- Describe architectural features of advanced processors.
- Interpret performance of different pipelined processors.
- Identify the best computer architecture for building multi core processors and hybrid architectures

**UNIT-I**

**Processors and Memory Hierarchy:** Advanced Processor Technology, Design Space of Processors, Instruction-Set Architectures, CISC scalar Processors, RISC scalar Processors, Super Scalar and Vector Processors: Superscalar Processors.

**Pipeline and vector processing:** Parallel Processing, Pipelining, Arithmetic Pipeline, Instruction Pipeline, RISC Pipeline, Vector Processing, Array Processors.

**UNIT-II**

**Memory Systems and Buses:** Memory hierarchy-cache and shared memory concepts-Cache memory organization-cache addressing models, Aliasing problem in cache, cache memory mapping techniques-Shared memory organization-Interleaved memory organization, Lower order interleaving, Higher order interleaving. Backplane bus systems-Bus addressing, arbitration and transaction.

**UNIT-III**

**Multi-Processor and Multi Computers:** Multiprocessor system interconnects- Cross bar switch, Multiport Memory-Hot spot problem, Message passing mechanisms-Pipelined Processors-Linear pipeline, on linear pipeline Instruction pipeline design-Arithmetic pipeline design.

**UNIT-IV**

**Pipelining and Superscalar Techniques:** Linear Pipeline Processors: Asynchronous and Synchronous models, Clocking and Timing Control, Speedup, Efficiency and Throughput, Pipeline Schedule Optimization, Instruction Pipeline Design: Instruction Execution Phases, Mechanisms for Instruction Pipelining, Dynamic Instruction Scheduling, Branch Handling Techniques.

## **Textbooks**

1. Kai Hwang and Naresh Jotwani, “Advanced Computer Architecture: Parallelism, Scalability, Programmability”, 3<sup>rd</sup> Edition, McGraw Hill Publisher, 2016, ISBN: 978-9339220921.
2. Kai Hwang and F.A.Briggs, “Computer Architecture And Parallel Processor”, 1<sup>st</sup> Edition, McGraw Hill Publisher, 1984, ISBN: 978-0070315563.

## **Reference Books**

1. Morris M. Mano, “Computer System Architecture”, 3rd Edition, Pearson/Prentice Hall India, 2007, ISBN: 978-8131700709.
2. John L. Hennessy, David A. Patterson., “Computer Architecture - A Quantitative Approach”, 6th Edition, Morgan Kaufman Series, 2017, ISBN: 978-0128119051.
3. Vincent P. Heuring and Harry F. Jordan, “Computer Systems Design and Architecture”, 2nd Edition, Pearson Publisher, 2003, ISBN: 978-0130484406.
4. William Stallings, “Computer Organization and Architecture”, 11th Edition, Pearson Publisher, 2018, ISBN: 978-0134997193.
5. Dezso Sima, Terence Fountain, and Peter Karsuk, “Advanced Computer Architectures-A Design Space Approach”, 1<sup>st</sup> Edition, Pearson Publisher, 2002, ISBN: 978-8131702086.
6. John Paul Shen and Mikko H. Lipasti, “Modern Processor Design”, McGraw Hill Education Publisher, 2004, ISBN: 978-00707570641.

## **Online Sources**

1. <https://nptel.ac.in/courses/106103206>
2. <https://nptel.ac.in/courses/106102229>

**Course Objective**

- To develop skills to design and analyze linear and non-linear data structures.
- Develop algorithms for manipulating linked lists, stacks, queues, trees and graphs.
- Develop recursive algorithms as they apply to trees and graphs.
- To Strengthen the ability to identify and apply the suitable data structure for the given real-world problem.

**Course Outcome**

- Analyze the asymptotic performance of algorithms and methodologies to solve recurrence relations for algorithms
- Design and develop applications in linear and nonlinear list data structure
- Design and develop algorithms using standard paradigms like Greedy, Backtracking, Divide & conquer and dynamic programming
- Construct efficient algorithms for some common computer engineering design problems

**UNIT-I**

**Complexity Analysis:** Time and Space complexity of algorithms, asymptotic notation, Recurrence

**Lists:** Abstract data type, sequential and linked representations, doubly linked lists, circular lists, applications of lists in bin sort, radix sort.

**Stack & Queue:** Stack operations, Types of Queues, Queue Operations

**UNIT-II**

**Graph:** Directed and undirected graphs, Connectivity in graphs, Graph traversal – breadth first and depth first, Spanning trees- Kruskal& Prim's algorithms.

**Trees:** Binary trees and their properties, Binary Search Tree, AVL Tree, B-Tree, Threaded binary trees

**UNIT-III**

**Algorithms:** Greedy Technique, Backtracking, Divide & Conquer, Dynamic Programming, Single Source Shortest Path, All Pair Shortest Path

**String Matching:** Brute-Force Pattern Matching, The Boyer- Moore Algorithm, The Longest Common Subsequence Problem (LCS).

**UNIT-IV**

**Hashing:** Review of Hashing, Hash Function, Collision Resolution Techniques in Hashing, Separate Chaining, Open Addressing, Linear Probing, Quadratic Probing, Double Hashing, Rehashing, Extendible Hashing.

**Complexity Theory:** P, NP and NP-Completeness, Approximate Algorithm, Computational Geometry.

## **Textbooks**

1. S. Sahni, “Data Structures, Algorithms, and Applications in C++”, 2<sup>nd</sup> Edition, Silicon Press, 2005, ISBN: 978-8173715228.
2. Thomas Cormen, Charles Leiserson, Ronald Rivest and Clifford Stein, “Introduction to Algorithms”, 3<sup>rd</sup> Edition, MIT Press, 2009, ISBN: 978-0262033848.

## **Reference Books**

1. E. Horowitz, S. Sahni and S. Rajasekaran, “Fundamentals of Computer Algorithms”, 2<sup>nd</sup> Edition, 2008, Universities Press, India, ISBN: 978-8173716126.
2. A. M. Tenenbaum, Y. Langsam, and M. J. Augenstein, “Data Structures Using C and C++”, 2<sup>nd</sup> Edition, Prentice Hall, 1995, ISBN: 978-0130369970.

## **Online Sources**

1. <https://nptel.ac.in/courses/106106127>
2. <https://nptel.ac.in/courses/106106131>
3. [https://www.tutorialspoint.com/data\\_structures\\_algorithms/index.htm](https://www.tutorialspoint.com/data_structures_algorithms/index.htm)
4. <https://www.programiz.com/dsa>

**Course Objective**

- Introduce concepts of mathematical logic for analyzing propositions and proving theorems.
- Use sets for solving applied problems and use the properties of set operations algebraically.
- Work with relations and investigate their properties.
- Investigate functions as relations and their properties.
- Introduce basic concepts of graphs, digraphs, and trees.

**Course Outcome**

- Able to apply basic terminology, formal logic, Notation, and its application.
- Able to differentiate the theory of inference and Predicate Calculus.
- Able to specify and manipulate basic mathematical objects such as sets, functions and induction to recursion.
- Able to apply the basic concepts of Algebraic Structures & Groups.
- Able to implement the basic concepts of graphs, trees and related algorithms.

**UNIT-I**

**Mathematical Logic:** propositional logic, propositional equivalences, predicates & quantifiers, rule of inference, direct proofs, proof by contraposition, proof by contradiction.

**Boolean Algebra:** Boolean functions and its representation, logic gates, minimizations of circuits by using Boolean identities and K-map.

**UNIT-II**

**Basic Structures:** Sets representations, set operations, functions, sequences and summations, Division algorithm, modular arithmetic, solving congruences, applications of congruences.

**Recursion:** Proofs by mathematical induction, recursive definitions, structural induction, generalized induction, recursive algorithms.

**UNIT-III**

**Counting:** Basic counting principle, inclusion-exclusion for two-sets, pigeonhole principle, permutations and combinations, Binomial coefficient, and identities, generalized permutations and combinations.

**Recurrence Relations:** introduction, solving linear recurrence relations, generating functions, principle of inclusion-exclusion, applications of inclusion-exclusion.

**UNIT-IV**

**Relations:** relations and their properties, representing relations, closures of relations, equivalence relations, partial orderings.

**Graphs & Trees:** Graphs definitions, types of graphs, representing graphs, graph isomorphism, connectivity of graphs, Euler and Hamilton paths and circuits, Dijkstra's algorithm to find shortest path, graph coloring and its applications, Tree definitions-properties of trees, tree traversals – pre-order, in-order, post-order, prefix, infix, postfix notations.



## **Textbook**

1. Kenneth H. Rosen, “Discrete Mathematics and its Applications”, 8<sup>th</sup> Edition, McGraw Hill Publisher, 2019, ISBN: 978-1259676512.

## **Reference Books**

1. Ralph P. Grimaldi, “Discrete and Combinatorial Mathematics: An Applied Introduction”, 5<sup>th</sup> Edition, Pearson Publisher, 2019, ISBN: 978-93535433055.
2. Cliff L. Stein, Robert Drysdale, Kenneth Bogart, “Discrete Mathematics for Computer Scientists, 1<sup>st</sup> Edition, Pearson Publisher, 2010, ISBN: 978-0132122719.
3. J.P. Tremblay, R. Manohar, “Discrete Mathematical Structures with Applications to Computer Science”, Tata McGraw-Hill Education, 1997, ISBN: 978-0074631133.
4. Joe L. Mott, Abraham Kandel, Theodore P. Baker, “Discrete Mathematics for Computer Scientists and Mathematicians”, 2<sup>nd</sup> Edition, Brady Publisher, 1985, ISBN: 978-0835913911.

## **Online Sources**

1. <https://nptel.ac.in/courses/106108227>
2. <https://nptel.ac.in/courses/106106183>
3. <https://nptel.ac.in/courses/106103205>

**Course Objective**

- Define, explain, and apply introductory operating systems concepts: process management, inter-process communication, memory management, I/O systems, file systems.
- Design and implement a correct concurrent program requiring synchronization
- Gain experience in implementing and debugging operating system components, including the kernel module, system call, synchronization primitives, and the file system
- Understanding the importance of Distributed and Multiprocessor Operating system

**Course Outcome**

- Identify the design approaches of advanced operating systems
- Analyze the design issues of distributed operating systems.
- Evaluate design issues of multi-processor operating systems.
- Identify the requirements of Distributed File System and Distributed Shared Memory.
- Formulate the solutions to schedule the real time applications.

**UNIT-I**

**Operating-System Structures:** Operating-System Services, User Interface for Operating-System, System Calls, Types of System Calls, Operating-System Design and Implementation, Operating-System Structure, Operating- System Debugging.

**Process Management:** Process Concept, Process Scheduling, Operations on Processes, Inter process Communication, Examples of IPC Systems, Communication in Client–Server Systems.

**Threads:** Overview, Multithreading Models, Threading Issues.

**UNIT-II**

**Memory Management:** Main Memory, Swapping, Contiguous Memory Allocation, Segmentation, Paging, Structure of the Page Table. Virtual Memory: Demand Paging, Page Replacement, Allocation of Frames, Thrashing, Memory-Mapped Files.

**Mass-Storage Structure:** Overview, Disk Structure, Disk Scheduling, Disk Management, Swap-Space Management, RAID Structure, Stable-Storage Implementation.

**UNIT-III**

**File Systems:** File Concept, Access Methods, Directory and Disk Structure, Filesystem Mounting, Protection. File-System Structure and Implementation, Directory Implementation, Allocation Methods, Free-Space Management, Recovery, Network File System.

**Protection and Security:** Goals of Protection, Principles of Protection, Access Matrix, Access Control, Revocation of Access Rights, Program Threats, System and Network Threats, Cryptography as a Security Tool, User Authentication, Implementing Security Defences.

## **UNIT-IV**

**Architectures of Distributed Systems:** System Architecture Types, Distributed Operating Systems, Issues in Distributed Operating Systems, Deadlock Handling Strategies in Distributed Systems, Issues in Deadlock Detection and Resolution, Distributed Deadlock Detection Algorithms

**Multi-Processor Operating Systems:** Motivation for multiprocessor Systems, Basic Multiprocessor System Architectures, Structures of Multiprocessor Operating Systems, Operating Design Issues.

### **Textbooks**

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, “Operating System Concepts”, 10<sup>th</sup> Edition, Wiley Publisher, 2018, ISBN: 978-1119320913.
2. Andrew S. Tanenbaum and Maarten van Steen. “Distributed Systems: Principles and Paradigms”, 2nd Edition, Prentice Hall, 2007, ISBN: 978-8120334984.

### **Reference Books**

1. Mukesh Singhal, Niranjana G. Shivaratri, “Advanced Concepts in Operating Systems”, Tata McGraw-Hill Education, 2011, ISBN: 978-0070575721.
2. Thomas W. Doepfner, “Operating systems in Depth”, John Wiley & Sons, 2011, ISBN: 978-1118136393.
3. William Stallings, “Operating Systems: Internals and Design Principles”, 9<sup>th</sup> Edition, Pearson Publisher, 2018, ISBN: 978-9352866717.
4. Dhananjay M. Dhamdhere, “Operating Systems: A Concept Based Approach”, 3<sup>rd</sup> Edition, McGraw Hill Education, 2017, ISBN: 978-1259005589.

### **Online Sources**

1. <https://nptel.ac.in/courses/106102132>
2. <https://www.geeksforgeeks.org/operating-systems/>

**Course Objective**

- Enable students to get in-depth practical knowledge of C Language
- Developing problem-solving ability by analyzing a problem statement
- Design and develop applications using files, structures, preprocessor directives and memory allocation functions.
- Design animation contents using C Language.

**Course Outcome**

- Enable students to apply code reusability with functions and pointers.
- Design and simulate animation contents using Graphics and preprocessor directives.
- Develop application software using dynamic memory allocation and files handling functions.

Sl. No.	List of Experiments
1	Write a program using Function <ol style="list-style-type: none"> <li>To design a simple calculator for arithmetic operations.</li> <li>To use a recursive function to generate Fibonacci series.</li> <li>To find if a given number is prime or not.</li> <li>To check whether an entered number is Perfect number or not.</li> </ol>
2	Write a program using Array and String <ol style="list-style-type: none"> <li>To return an entered String in UPPERCASE.</li> <li>To return the ASCII value of each characters of an inputted string.</li> <li>To create a function that accepts an array of integer values and find the number which divides all other numbers.</li> </ol>
3	Write a program to demonstrate the use of Structures <ol style="list-style-type: none"> <li>To store a date in mm-dd-yyyy format and display it in words.</li> <li>To create a list of books using array of structure that will include title, author, publisher, publishing year, number of pages and price.</li> <li>To store two different times including seconds, minutes and hours and find the difference between them using a function.</li> </ol>
4	Write a program to demonstrate the use of Pointers <ol style="list-style-type: none"> <li>To accept a string using character pointer and display it.</li> <li>To compute the square and cube of an inputted number using pointer of a variable containing the entered number.</li> <li>To create separate functions to add, multiply, subtract two numbers and call the functions using a function pointer.</li> </ol>
5	Write a program to demonstrate the use of Files <ol style="list-style-type: none"> <li>To create, open, read, write, and display the file contents.</li> <li>To find the size of file.</li> <li>To count number of lines, words, characters in a file.</li> <li>To combine the contents of two different files in a third file.</li> <li>To write the contents of a file in reverse order in another file.</li> <li>To display number 1 to 100 and redirect the output to a text file.</li> </ol>
6	Write a program to demonstrate the use of Graphics <ol style="list-style-type: none"> <li>To draw basic graphics construction like, line, rectangle, ellipse, circle, and arc.</li> <li>To create a screensaver using different circles filled with different colors and patterns.</li> <li>To create a countdown timer.</li> <li>To draw a moving car.</li> <li>To implement bouncing ball using sine wave form.</li> <li>To draw a smiling face animation.</li> </ol>

## **Textbook**

1. Herbert Schildt, "C: The Complete Reference", 4<sup>th</sup> Edition, McGraw Hill Education, 2017, ISBN: 978-0070411838.

## **Reference Books**

1. Ashok N. Kamthane, "Programming in C", 3<sup>rd</sup> Edition, Pearson Publisher, 2015, ISBN: 978-9332543553.
2. Y. Kanetkar, "Let Us C: Authentic Guide to C Programming", 18<sup>th</sup> Edition, BPB Publisher, 2021, ISBN: 978-9391392994.

## **Online Sources**

1. <https://www.programiz.com/c-programming>
2. <https://www.javatpoint.com/c-programming-language-tutorial>
3. <https://www.tutorialspoint.com/cprogramming/index.htm>
4. <https://www.geeksforgeeks.org/c-programming-language/>

**Course Objective**

- Understand and remember algorithms and its analysis procedure.
- Introduce the concept of data structures through ADT including List, Stack, Queues.
- To design and implement various data structure algorithms.
- To introduce various techniques for representation of the data in the real world.
- To develop application using data structure algorithms.
- Compute the complexity of various algorithms.

**Course Outcome**

- Selects appropriate data structures as applied to specified problem definition.
- Implement operations like searching, insertion, and deletion, traversing mechanism etc. on various data structures.
- Students will be able to implement Linear data structures.
- Implement appropriate sorting/searching technique for given problem.
- Design advance data structure using Non-Linear data structure.
- Determine and analyze the complexity of given Algorithms.

**Software Requirement: C**

Sl. No.	List of Experiments
1	Write a program to perform the following operations on singly linked list. i) Creation ii) Insertion iii) Deletion iv) Traversal.
2	Write a program that implements stack (its operations) using i) Arrays ii) linked list
3	Write a programs that implements Queue (its operations) using i) Arrays ii) linked list
4	Write a program that implements the Quick Sort method to sort a given list of integers in ascending order.
5	Write a program that implement the Merge Sort method to sort a given list of integers in ascending order.
6	Write a program to perform the following: i) Creating a Binary Tree of integers ii) Traversing the above binary tree in pre-order, in-order and post-order.
7	Write a program to perform the following: i) Creating a AVL Tree of integers.
8	Write a program that implement Kruskal's algorithm using a disjoint set data structure.
9	Write a program to find the minimal spanning tree of a graph using the Prim's algorithm.
10	Write a program to simulate various graph traversing algorithms.

## **Textbook**

1. Mark Allen Weiss, “Data Structures and Algorithm Analysis in C++”, 4<sup>th</sup> Edition, Pearson Education, 2014, ISBN: 978-0132847377.

## **Reference Books**

1. Sartaj Sahni, “Data Structures Algorithms and Applications in C++”, 2<sup>nd</sup> Edition, Silicon Press, 2005, ISBN: 978-0929306322.
2. Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekharan, “Fundamentals of Computer Algorithms”, 2<sup>nd</sup> Edition, Silicon Press, 2008, ISBN: 978-0929306414.
3. Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman, “The Design and Analysis of Computer Algorithms”, 1<sup>st</sup> Edition, Pearson Education, 2002, ISBN: 978-8131702055.
4. Adam Drozdek, “Data Structures and Algorithms in JAVA”, 3<sup>rd</sup> Edition, Cengage Learning, 2008, ISBN: 978-9814239233.
5. Ellis Horowitz, Sartaj Sahni, Dinesh Mehta, “Fundamentals of Data Structures in C++”, 2<sup>nd</sup> Edition, Universities Press, 2008, ISBN: 978-8173716065.

## **Online Sources**

1. <https://www.geeksforgeeks.org/advanced-data-structures/>
2. <https://www.javatpoint.com/data-structure-tutorial>

## Semester-2<sup>nd</sup>

Paper Code	Subject	Title of the Paper	Credit	Max. Marks		
				Internal	Univ. Exam	Total
CS-402	Core	Theory of Computation	4	20	80	100
CS-404	Core	Python Programming	4	20	80	100
CS-406	Core	Advanced Computer Networks	4	20	80	100
CS-408	Core	Soft Computing	4	20	80	100
CS-410	Elective-I	Artificial Intelligence or Information Security & Cryptography or Wireless Sensor Network or Mobile Computing	4	20	80	100
OE-CS-412	Open Elective	MATLAB Programming	5	20	80	100
CS-414	Practical	Soft Computing Lab	4			100
CS-416	Practical	Python Programming Lab	4			100
<b>Total</b>			<b>33</b>	<b>Total Marks</b>		<b>800</b>



**Course Objective**

- Understand basic properties of deterministic and nondeterministic finite automata
- Understanding the Context free languages and grammars, and Normalizing CFG.
- Understand the concept of Pushdown automata and its application.
- Understand basic properties of Turing machines and computing with Turing machines.
- Know the concepts of tractability and decidability, the concepts of NP-completeness and NP-hard problem.
- Understand the challenges for Theoretical Computer Science and its contribution to other sciences.

**Course Outcome**

- Acquire a fundamental knowledge of the core concepts in automata theory and formal languages.
- An ability to design grammars and automata (recognizers) for different language classes.
- An ability to identify formal language classes and prove language membership properties.
- An ability to prove and disprove theorems establishing key properties of formal languages and automata.
- Acquire a fundamental knowledge of core concepts relating to the theory of computation and computational models including (but not limited to) decidability and intractability.
- Design and analyze the Turing machine for formal languages.

**UNIT – I**

**Fundamentals:** Alphabets, Strings, Languages, Problems, Graphs, Trees, Finite State Systems, Definitions, Finite Automaton Model, Acceptance of Strings, And Languages.

**Types of FA:** Deterministic Finite Automaton and Nondeterministic Finite Automaton, Transition Diagrams, Transition Tables, Proliferation Trees and Language Recognizers, Equivalence Of DFA And NFA. Finite Automata With  $\epsilon$ -Moves, Significance, Acceptance Of Languages,  $\epsilon$ -Closure, Equivalence Of NFA With And Without  $\epsilon$ -Moves, Minimization Of Finite Automata, Finite Automata With Output– Moore And Mealy Machines.

**UNIT – II**

**Regular Languages:** Regular Sets, Regular Expressions, Identity Rules, Constructing Finite Automata for A Given Regular Expressions, Conversion Of Finite Automata To Regular Expressions. Pumping Lemma of Regular Sets And Its Applications, Closure Properties Of Regular Sets.

**Grammar Formalism:** Regular Grammars–Right Linear and Left Linear Grammars, Equivalence Between Regular Linear Grammar And Finite Automata, Inter Conversion, Context Free Grammar, Derivation Trees, Sentential Forms, Right Most And Leftmost Derivation Of Strings, Ambiguity.

**UNIT – III**

**Context Free Grammars:** Simplification of Context Free Grammars, Chomsky Normal Form, Greibach Normal Form, Pumping Lemma For Context Free Languages And Its Applications, Closure

of properties of CFL (proofs omitted), Membership Algorithm (CYK Algorithm) for Context Free Grammars.

**Push Down Automata:** PDA Definition, Model, Acceptance Of CFL, Acceptance By Final State And Acceptance By Empty State And Its Equivalence. Equivalence Of PDA's And CFL's, Inter-Conversion. (Proofs Not Required).

## **UNIT – IV**

**Turing Machine:** TM Definition, Model, Design Of TM, Computable Functions, Unrestricted Grammars, Recursively Enumerable Languages. Church's Hypothesis, Counter Machine, Types Of Turing Machines (Proofs Omitted). Linear Bounded Automata And Context Sensitive Language.

**Computability Theory:** Chomsky Hierarchy Of Languages, Definitions Of P And NP Problems, NP Complete And NP Hard Problems.

### **Textbook**

1. J. E. Hopcroft, Rajeev Motwani and J. D. Ullman, "Introduction to Automata Theory, Languages, and Computation", 3<sup>rd</sup> Edition, Pearson Publisher, 2008, ISBN: 978-8131720479.

### **Reference Books**

1. John C. Martin, "Introduction to Languages and the Theory of Computation", 4<sup>th</sup> Edition, McGraw Hill Education, 2010, ISBN: 978-0073191461.
2. K. L. P. Mishra and N. Chandrasekaran, "Theory of Computer Science: Automata, languages and Computation", 3<sup>rd</sup> Edition, 2006, Prentice Hall of India, ISBN: 978-8120329683.
3. Peter Linz, "An Introduction to Formal Languages and Automata", 6<sup>th</sup> Edition, Jones & Bartlett Publisher, 2016, ISBN: 978-9384323219.
4. Zvi Kohav, Niraj K. Jha, "Switching and Finite Automata Theory", 3<sup>rd</sup> Edition, Cambridge University Press, 2009, ISBN: 978-0521857482.

### **Online Sources**

1. <https://nptel.ac.in/courses/106103070>
2. <https://nptel.ac.in/courses/106105196>
3. <https://nptel.ac.in/courses/106106049>

**Course Objective**

- Learn Syntax and Semantics and create Functions in Python
- Handle Strings and Files in Python
- Understand Lists, Dictionaries and Regular expressions in Python
- Understand use of functions and file handling in Python
- Implement Object Oriented Programming concepts in Python

**Course Outcome**

- Identifies Python syntax and semantics and analyzes the use of flow control in python
- Apply the concepts of functions, File Systems, and Exceptions for developing python programs
- Create, run, and manipulate Python Programs using core data structures like Lists, Dictionaries
- Implement file handling functions and user defined functions in Python
- Interpret the concepts of Object-Oriented Programming as used in Python

**UNIT-I**

**Introduction to Python:** Python Variables, Python Basic Operators, Understanding Python Blocks. Python Data Types, Declaring and Using Numeric Data Types: Int, Float Etc.

**Python Program Flow Control Conditional blocks:** If, Else and Else If, Looping In Python, Loop Manipulation Using Pass, Continue, Break And Else. Programming Using Python Conditional and Loop Blocks.

**UNIT-II**

**Functions:** Introduction, Defining And Calling A Void Function, Designing A Program To Use Functions, Local Variables, Passing Arguments To Functions, Global Variables And Global Constants, Value-Returning Functions-Generating Random Numbers, Writing Our Own Value-Returning Functions, The Math Module, Storing Functions In Modules.

**File and Exceptions:** Introduction To File Input And Output, Using Loops To Process Files, Processing Records, Exceptions.

**UNIT-III**

**Lists and Tuples:** Sequences, Introduction to Lists, List slicing, Finding Items in Lists with the in Operator, List, Methods and Useful Built-in Functions, Copying Lists, Processing Lists, Two-Dimensional Lists, Tuples.

**Strings:** Basic String Operations, String Slicing, Testing, Searching, and Manipulating Strings.

**Recursion:** Introduction, Problem Solving with Recursion, Examples of Recursive Algorithms.

## **UNIT-IV**

**Object-Oriented Programming:** Procedural and Object-Oriented Programming, Classes, Working with Instances, Techniques for Designing Classes, Inheritance, Polymorphism.

**Python packages:** Simple programs using the built-in functions of packages matplotlib, Numpy, pandas etc.

### **Textbook**

1. Wesley J. Chun, “Core Python Applications Programming”, 3<sup>rd</sup> Edition, Pearson Education, 2012, ISBN: 978-0132678209.

### **Reference Books**

1. Charles Dierbach, “Introduction to Computer Science using Python”, 1<sup>st</sup> Edition, Wiley Publisher, 2015, ISBN: 978-8126556014.
2. Jake VanderPlas, “Python Data Science Handbook”, 1<sup>st</sup> Edition, O’Reilly Media Inc., 2016, ISBN: 978-1491912058.
3. Jeeva Jose and P. Sojan Lal, “Introduction to Computing and Problem Solving with PYTHON”, 1<sup>st</sup> Edition, Khanna Publishers, New Delhi, 2016, ISBN: 978-9382609810.
4. Allen B. Downey, Jeffrey Elkner and Chris Meyers, “How to think like a Computer Scientist: Learning with Python”, 1<sup>st</sup> Edition, Green Tea Press, 2002, ISBN : 978-0971677500.
5. R. Nageswara Rao, “Core Python Programming”, 3<sup>rd</sup> Edition, Dreamtech Press, 2021, ISBN: 978-9390457151.

### **Online Sources**

1. <https://nptel.ac.in/courses/106106145>
2. <https://docs.python.org/3/tutorial/>

**Course Objective**

- To learn a few of the advanced technologies, including routing protocols, wireless & mobile networks etc.
- Understand the architecture of the Internet protocols as a layered model.
- To understand the fundamentals of data transmission, encoding and multiplexing.
- To understand how the various components of wide area networks and local area networks work together.

**Course Outcome**

- Able to Configure IPv6 Network
- Familiarization of routing protocol in the given network situation.
- Able to apply TCP/IP protocol stack in the design and implementation of network systems
- Configure various Application Layer Protocols

**UNIT-I**

**Basic Networking Concepts Revisited:** Introduction To Networks, Layering And Link Layer, Network Layer, Routing, End-To-End Layer, Congestion Control, Network Layer And Protocols: Ipv4, Ipv6, Transition From Ipv4 To Ipv6, Mobile IP, VPN

**Bridging and Switching:** Concept Of Bridging, Bridging Algorithms; Ethernet Switches; Virtual LANs.

**UNIT-II**

**Types of Routing:** Unicast, Multicast, Broadcast

**Routing Protocols:** Intra-Domain, Inter-Domain Routing, Routing Protocols (Distance Vector Routing, Bellman-Ford Algorithm, Link State Routing, Path Vector Routing)

**Unicast Routing Protocols:** Internet Structure, Routing Information Protocol (RIP), Open Shortest Path First (OSPF), Border Gateway Protocol (BGPv4)

**Multicast Routing:** Multicast Addresses; Multicast Routing Protocols (DVMRP, MOSPF, PIM), Congestion Control, Address Learning Bridges, Spanning Tree, Source Routing, Bridges, Routers, Gateway.

**UNIT-III**

**Transport Layer:** Design Issues, Connection Management, Transmission Control Protocol (TCP), Flow and Congestion Control, TCP Variants, TCP Modeling, Active Queue Management

**User Datagram Protocol (UDP):** User Datagram, UDP Services, UDP Applications

**UNIT-IV**

**Application Layer Protocols:** Concept of DNS and its Operation, Static and Dynamic Allocation (DHCP), Remote Login – TELNET and SSH

**Introduction on Emerging applications:** VoIP, SIP, video over P2P, Introduction to Network Security, Cloud, IOT & BYOD

## **Textbook**

1. J. F. Kurose, K. W. Ross, “Computer Networking. A Top-Down Approach”, 8<sup>th</sup> Edition, Pearson Education, 2021, ISBN: 978-1292405469.

## **Reference Books**

1. Andrew S. Tanenbaum and David J. Wetherall, “Computer Networks”, 5th Edition, Pearson Education, 2014, ISBN: 978-9332518742.
2. Forouzan Behrouz A., “Data Communication and Networking 5E”, McGraw Hill Education (India), New Delhi, 2005, ISBN:978-1-25-906475-3
3. William Stallings, "Data and Computer Communications", 10<sup>th</sup> Edition, Pearson Education, 2014, ISBN: 978-1292014388.
4. W. Richard Stevens and Kevin R. Fall, "TCP/IP Illustrated - Volume I, The Protocols", 2<sup>nd</sup> Edition, Pearson Education, 2014, ISBN: 978-9332535954.
5. Douglas E. Comer, “Internetworking with TCP/IP: Principles, Protocol and Architecture - Volume I”, 6<sup>th</sup> Edition, Pearson Education, 2015, ISBN: 978-9332550100.

## **Online Resources**

1. <https://nptel.ac.in/courses/106105183>
2. TCP/IP Illustrated, Volume 1 The Protocols W. Richard Stevens
3. <http://study-ccna.com/>
4. <http://www.packettracemetwork.com/>
5. <https://www.tutorialspoint.com/listtutorials/networking/1>
6. [www.txv6tf.org/wp-content/uploads/2010/08/Muhummad-Tutorial-ipv6-basics.pdf](http://www.txv6tf.org/wp-content/uploads/2010/08/Muhummad-Tutorial-ipv6-basics.pdf)
7. <http://cnp3book.info.ucl.ac.be/2nd/html/protocols/bgp.html>
8. <http://www.ciscopress.com/articles/article.asp?p=2180210andseqNum=5>

**Course Objective**

- To provide an understanding of the soft computing field.
- To familiarize with neural networks and learning methods for neural networks.
- To understand the underlying principles of Fuzzy set theory and Fuzzy Inference.
- To introduce basics of genetic algorithms and their applications in optimization and planning.
- To develop skills thorough understanding of the theoretical and practical aspects of Soft Computing.

**Course Outcomes**

- Identify and describe soft computing techniques and their roles in building intelligent machines.
- Recognize the feasibility of applying neural network methodology for a particular problem.
- Apply fuzzy logic and reasoning to handle uncertainty and solve engineering problems.
- Apply genetic algorithms to combinatorial optimization and real-life problems.
- Apply and evaluate the Swarm based algorithms to solve real life optimization problems.

**UNIT-I**

**Basics of Soft Computing:** Overview of Soft Computing, Difference Between Soft and Hard Computing, Characteristics of Soft Computing, Requirement of Soft Computing, Applications of Soft Computing, Components of Soft Computing.

**UNIT-II**

**Neural Network:** What Is Neural Network?, How Brain Works?, Neuron as a Simple Computing Element, Basic Building Block Of An Artificial Neuron, The Perceptron, Learning Rules And Various Activation Functions, McCulloch & Pitts Model, Single Layer and Multilayer Perceptrons, The Hopfield Network, RBF Neural Network, Back Propagation Networks, Architecture of Backpropagation (BP) Networks, Backpropagation Learning, Accelerated Learning in Multilayer Perceptron.

**UNIT-III**

**Fuzzy Logic:** Fuzzy Set theory, Fuzzy versus Crisp set, Fuzzy Relation, Membership Functions, Fuzzy Rules & Fuzzy Reasoning, Fuzzification, Minmax Composition, Defuzzification Method, Fuzzy Logic, Fuzzy Rule based systems, Predicate logic, Fuzzy Decision Making, Fuzzy Control Systems, Fuzzy Classification.

**UNIT-IV**

**Genetic Algorithms:** Fundamentals of Genetic Algorithms: Encoding, Fitness Functions, Reproduction. Genetic Modeling: Encoding, Crossover, Selection, Mutation, Convergence of GA, Applications to Real Life Problems, Issues Related to GA.

**Swarm Algorithms:** Fundamentals of Swarm Algorithms, Reflection of Various Animal Behavior in Swarm Algorithms, Particle Swarm Optimization, Applications of PSO.

## **Textbooks**

1. James A. Freeman and David M. Skapura, “Neural Networks: Algorithms, Applications, and Programming Techniques”, 1<sup>st</sup> Edition, Pearson Education, 2012, ISBN: 978-8131708088.
2. Jyh-Shing Roger Jang, Chuen-Tsai Sun, Eiji Mizutani, “Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning Machine Intelligence”, Prentice Hall of India, 2008, ISBN: 978-8120322431.
3. N. P. Padhy and S. P. Simon, “Soft Computing with MATLAB Programming”, U.K. Edition, Oxford University Press, 2015, ISBN: 978-0199455423.
4. S. N. Sivanandam & S. N. Deepa, “Principles Of Soft Computing”, Wiley India, 2007, ISBN: 978-8126510757.

## **Reference Books**

1. S. Rajasekaran, G. A. Vijayalakshmi Pai, “Neural Networks, Fuzzy Logic, And Genetic Algorithm: Synthesis And Applications”, Eastern Economy Edition, Prentice Hall of India, 2003, ISBN: 978-8120321861.
2. Simon Haykins, “Neural Networks: A Comprehensive Foundation”, 2<sup>nd</sup> Edition, Pearson Education, 1997, ISBN: 978-0138958633.
3. David E. Goldberg, “Genetic Algorithm in Search Optimization and Machine Learning” 13<sup>th</sup> Edition, Addison Wesley, 1989, ISBN: 978-0201175673.
4. Vojislav Kecman, “Learning and Soft Computing: Support Vector Machines, Neural Networks and Fuzzy Logic Models”, MIT Press, 2001, ISBN: 978-0262527903.
5. F. O. Karry and C. De Silva, “Soft Computing and Intelligent Systems Design – Theory, Tools and Applications”, 1<sup>st</sup> Edition, Pearson Education, 2009, ISBN: 978-8131723241.
6. George J. Klir and Bo Yuan, “Fuzzy Sets and Fuzzy Logic: Theory and Applications”, 1<sup>st</sup> Edition, Pearson Publisher, 1995, ISBN: 978-0131011717.

## **Online Sources**

1. <http://nptel.ac.in/courses/117105084/>
2. <http://nptel.ac.in/courses/108104049/>



**Course Objective**

- Understanding of introductory concepts and problem-solving techniques in Artificial Intelligence.
- Getting insight to Intelligent agents, Reasoning & Logic propositional Logic.
- Understand concept of planning in Artificial intelligence.
- Getting concepts on Learning in Artificial Neural Networks and various neural network models
- Get idea of knowledge representation and Natural language processing in Artificial intelligence.

**Course Outcome**

- Able to assess critically the techniques presented and to apply them to real world problems.
- Familiarity with the major concepts of intelligent agents, Reasoning & Logic propositional Logic.
- Able to implement the concept of planning in Artificial intelligence.
- Able to implement the Learning intelligent artificial agents.
- Aware of expert system and pattern recognition in Artificial intelligence.

**UNIT – I**

**Introduction to Artificial Intelligence:** Introduction, AI Techniques, Problem Solving - Problem-Solving Process, Formulating Problem, Problem Types and Characteristics, Problem Analysis and Representation, Problem Space And Search, Real-World Problems, Problem Reduction Methods.

**Search Methods:** General Search Algorithm, Uniformed Search Methods – BFS, DFS, Informed Search – Best First Search, Hill Climbing, A\* Search, Iterative Deepening, Min Max Search, Alpha-Beta Pruning, Constraint Satisfaction Search.

**UNIT – II**

**Logic:** Propositional Logic, Predicate Logic, Unification, Inference in FOL, Representing Knowledge Using Rules, Resolution.

**Knowledge and Reasoning:** Knowledge Representation, Knowledge-Based Agents, The Wumpus World, Semantic Networks, Frame Systems, Conceptual Dependency, Inference, Types of Reasoning.

**UNIT – III**

**Uncertain Knowledge and Reasoning:** Uncertainty and Methods, Bayesian Probability and Belief Network, Probabilistic Reasoning, Forward and Backward Reasoning, Perception, Making Simple Decisions, Making Complex Decisions.

**Planning:** Simple Planning Agent, Planning Languages, Types of Planning, Blocks World, Goal Stack Planning, Means-Ends Analysis, Planning as a State-Space Search.

**UNIT – IV**

**Expert Systems:** Introduction, Rule Based Architecture of Expert System, Applications of Expert System, Knowledge Acquisition, Forward and Backward Chaining, Uncertainty Management in Expert Systems.

**Pattern Recognition:** Introduction, Machine Perception and Pattern Recognition, Feature Extraction, Classification, Object Recognition, Speech Recognition, Pattern Mining.

### **Textbooks**

1. Kevin Knight, Elaine Rich, B Nair, “Artificial Intelligence”, 3<sup>rd</sup> Edition, Tata McGraw Hill Publication, 2019, ISBN: 978-0070087705.
2. Stuart Russell and Peter Norvig, “Artificial Intelligence: A Modern Approach”, Global Edition, Pearson Education, 2021, ISBN: 978-1292401171.

### **Reference Books**

1. Nils J Nilsson, “Artificial Intelligence: A New Synthesis”, Morgan Kaufmann Publisher, 2004, ISBN: 978-0120883653.
2. Eugene Charniak, and Drew McDermott, “Introduction to Artificial Intelligence”, 2<sup>nd</sup> Edition, Pearson Education, 1985, ISBN: 978-8131703069.
3. R. O. Duda, P. E. Hart, and D. G. Stork, “Pattern Classification”, 2<sup>nd</sup> Edition, Wiley Publisher, 2007, ISBN: 978-8126511167.

### **Online Source**

1. <https://www.javatpoint.com/artificial-intelligence-tutorial>

**Course Description**

This course is to be offered by Computer Science Department to other departments like Physics, Chemistry, Mathematics, etc. This Open Elective course provides a gentle introduction to the MATLAB computing environment and is intended for beginning users and those looking for a review. It is designed to give students a basic understanding of MATLAB, including popular toolboxes. The course consists of interactive lectures and sample MATLAB problems given as assignments and discussed in class. No prior programming experience or knowledge of MATLAB is assumed. Concepts covered include basic use, graphical representations, and tips for designing and implementing MATLAB code.

**Course Objective**

- Understanding the MATLAB environment
- Being able to do simple calculations using MATLAB
- Being able to carry out simple numerical computations and analyses using MATLAB

**Course Outcome**

- Understand the main features of the MATLAB development environment
- Use the MATLAB GUI effectively
- Design simple algorithms to solve problems
- Write simple programs in MATLAB to solve scientific and mathematical problems

**UNIT-I**

**MATLAB basics:** The MATLAB Environment - Basic Computer Programming - Variables and Constants, Operators, And Simple Calculations - Formulas and Functions - MATLAB Toolboxes, Data Import & Export.

**UNIT-II**

**Procedures and Functions:** Arguments and Return Values, M-Files, Formatted Console Input-Output, String Handling.

**Control Statements:** Conditional Statements: If, Else, Elseif, Repetition Statements - While, For.

**UNIT-III**

**Matrices and vectors:** Matrix and Linear Algebra Review - Vectors and Matrices In MATLAB - Matrix Operations and Functions in MATLAB, Algebra, Calculus, Differential, Integration, Polynomials.

## **UNIT-IV**

**MATLAB Plots:** Basic Plotting, Plotting, Multiple Plot, 2-D Plot, 3-D Plot, Subplot, Handle Graphics, Animation.

### **Textbook**

1. Brian R. Hunt, Ronald L. Lipsman, Jonathan M. Rosenberg, “A Guide to MATLAB - for Beginners and Experienced Users”, 3<sup>rd</sup> Edition , Cambridge University Press, 2014, ISBN: 978-1107662223.

### **Reference Books**

1. Stephen J. Chapman, “Essentials of MATLAB Programming”, 3<sup>rd</sup> Edition, Cengage Learning, 2009, ISBN: 978-1305970656.
2. David McMahon, “MATLAB Demystified”, 1<sup>st</sup> Edition, McGraw-Hill Companies, 2007, ISBN: 978-0071485517.
3. Holly Moore, “MATLAB® for Engineers”, 6<sup>th</sup> Edition, Pearson Education, Inc., 2020, ISBN-013: 978-0357030394.
4. David M. Smith, “Engineering computation with MATLAB”, 3<sup>rd</sup> Edition, Pearson Education, Inc., 2013, ISBN: 978-0132568708.
5. Rudra Pratap, “Getting Started with MATLAB: A Quick Introduction for Scientists & Engineers”, 7<sup>th</sup> Edition, Oxford University Press, 2017, ISBN: 978-0190602062.

### **Online Source**

1. <https://www.mathworks.com/matlabcentral/>

**Course Objective**

- This course introduces soft computing techniques that are different from conventional AI techniques.
- This course also provides necessary mathematical background for understanding and implementing soft computing Techniques, such as neural networks, fuzzy systems, and genetic algorithms.

**Course Outcome**

- Familiarity with the MATLAB/ Python for developing various membership functions
- Develop various applications using soft computing techniques like Genetic Algorithms, Fuzzy Logic, Neural Networks, and their combination.
- Implement algorithms based on soft computing.
- Apply soft computing techniques to solve engineering or real-life problems.

**Software Requirement:** MATLAB/ Python

Sl. No.	List of Experiments
1	Write a program to plot various membership functions
2	Use Fuzzy toolbox to model tip value that is given after a dinner which can be-not good, satisfying, good and delightful and service which is poor, average, or good and the tip value will range from Rs. 10 to 100.
3	Write a program to Implement FIS Editor
4	Generate AND, NOT function using McCulloch-Pitts neural net.
5	Write a program to implement De-Morgan's Law.
6	Write a program for Perceptron net for an AND function with bipolar inputs and targets
7	Write a program for Hebb Net to classify two-dimensional input patterns in bipolar with their given targets
8	Write a program of Perceptron Training Algorithm
9	Plot the correlation plot on dataset and visualize giving an overview of relationships among data on iris data. Analysis of covariance: variance (ANOVA) if data have categorical variables on iris data.
10	Write a program to implement Back Propagation Algorithm.

**Textbooks**

1. Timothy J. Ross, "Fuzzy Logic With Engineering Applications", 3<sup>rd</sup> Edition, Wiley Publisher, 2011, ISBN: 978-8126531264.
2. S. Rajasekaran, G. A. Vijayalakshmi Pai, "Neural Networks, Fuzzy Logic, And Genetic Algorithm: Synthesis And Applications", Eastern Economy Edition, Prentice Hall of India, 2003, ISBN: 978-8120321861.

3. S. N. Sivanandam & S. N. Deepa, “Principles Of Soft Computing”, Wiley India, 2007, ISBN: 978-8126510757.

### **Reference Books**

1. David E. Goldberg, “Genetic Algorithm in Search Optimization and Machine Learning” 13<sup>th</sup> Edition, Addison Wesley, 1989, ISBN: 978-0201175673.
2. Jyh-Shing Roger Jang, Chuen-Tsai Sun, Eiji Mizutani, “Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning Machine Intelligence”, Prentice Hall of India, 2008, ISBN: 978-8120322431.
3. Satish Kumar, “Neural Networks: A Classroom Approach”, 2<sup>nd</sup> Edition, McGraw Hill Education, ISBN: 978-1259006166.
4. Samir Roy & Udit Chakraborty, “Introduction to Soft Computing: Neuro-Fuzzy and Genetic Algorithms”, 1<sup>st</sup> Edition, Pearson Education, 2013, ISBN: 978-8131792469.

### **Online Sources**

1. <https://www.javatpoint.com/what-is-soft-computing>
2. <https://www.javatpoint.com/fuzzy-logic>
3. <https://www.javatpoint.com/artificial-neural-network-genetic-algorithm>

**Course Objective**

- To be able to introduce core programming basics and program design with functions using Python programming language.
- To understand a range of Object-Oriented Programming, as well as in-depth data and information processing techniques.
- To understand the high-performance programs designed to strengthen the practical expertise.

**Course Outcome**

- Solve the basic mathematical problem using python programming
- Able to develop programs using basic concepts scripting and the contributions of scripting language
- Use basic data types of control structures and utility functions from standard library for faster programming
- Able to develop programs to store, retrieve and manipulate data with disk file

Sl. No.	List of Experiments
1	Write Python Program to do the following task. <ol style="list-style-type: none"> <li>Do arithmetical operations</li> <li>Find the area of a triangle</li> <li>To swap two variables</li> <li>To generate a random number</li> <li>To convert kilometers to miles</li> </ol>
2	Write the following Programs to understand the function in Python. <ol style="list-style-type: none"> <li>Decimal to Binary, Octal and Hexadecimal</li> <li>To Make a Simple Calculator</li> <li>To Display Calendar</li> <li>To Find Factorial of Number Using Recursion</li> </ol>
3	Write the following Programs to perform Loops and conditions in Python. <ol style="list-style-type: none"> <li>To Check Prime Number</li> <li>To Find the Factorial of a Number</li> <li>To Print the Fibonacci sequence</li> <li>To Find the Sum of Natural Numbers</li> </ol>
4	Write the following Programs to understand the Array in Python. <ol style="list-style-type: none"> <li>To copy all elements of one array into another array</li> <li>To find the frequency of each element in the array</li> <li>To print the elements of an array</li> <li>To print the largest/smallest element in an array</li> <li>To sort the elements of an array in ascending/descending order</li> </ol>
5	Write the following Programs to understand the Matrix in Python. <ol style="list-style-type: none"> <li>To Add Two Matrices</li> <li>To Multiply Two Matrices</li> <li>To Transpose a Matrix</li> </ol>
6	Write the following Programs to perform the String operations in Python. <ol style="list-style-type: none"> <li>To reverse a string</li> <li>To convert int to string</li> <li>To concatenate two strings</li> <li>To generate a random string</li> </ol>

7	Write the following Programs to perform List operations in Python. a. To append element in the list b. To compare two lists c. To remove an element from a list d. To add two lists
8	Write a Python program that inputs a text file. The program should print all the unique words in the file in alphabetical order.
9	Understanding various Python packages: Numpy, Matplotlib, Pandas

### **Textbook**

1. Al Sweigart, “Automate The Boring Stuff With Python: Practical Programming For Total Beginners”, 2<sup>nd</sup> Edition, No Starch Press, 2019, ISBN: 978-1593279929.

### **Reference Books**

1. Eric Matthes, “Python Crash Course: A Hands-On, Project-Based Introduction To Programming”, 2<sup>nd</sup> Edition, No Starch Press, 2019, ISBN: 978-1593279288.
2. Paul Barry, “Head First Python: A Brain-Friendly Guide”, 2<sup>nd</sup> Edition, O’Reilly Publisher, 2016, ISBN: 978-1491919538.
3. Mark Lutz, “Learning Python: Powerful Object-Oriented Programming”, 5<sup>th</sup> Edition, O’Reilly Publisher, 2013, ISBN: 978-1449355739.

### **Online Source**

1. <https://www.javatpoint.com/python-tutorial>



## Second Year

### Semester – 3<sup>rd</sup>

Paper Code	Subject	Title of the Paper	Credit	Max. Marks		
				Internal	Univ. Exam	Total
CS-501	Core	Internet of Things	4	20	80	100
CS-503	Core	Advanced JAVA Programming	4	20	80	100
CS-505	Core	Compiler Design	4	20	80	100
CS-507	Core	Advanced Database System and Implementation	4	20	80	100
CS-509	Elective-II	Machine Learning or Big Data Analytics or Data Mining or Introduction to Cyber Security	4	20	80	100
CS-511	Practical	Advanced JAVA Programming Lab	4		100	100
CS-513	Practical	Advanced Database Lab	4		100	100
<b>Total</b>			<b>28</b>		<b>Total Mark</b>	<b>700</b>

**Course Objective**

- To understand about the fundamentals of Internet of Things and its building blocks along with their characteristics
- To understand the recent application domains of IoT in everyday life
- To understand the protocols and standards designed for IoT and the current research on it.
- To understand the other associated technologies like cloud and fog computing in the domain of IoT

**Course Outcome**

- The students will be thorough about the technology behind the IoT and associated technologies
- The students will be able to use the IoT technologies in practical domains of society.
- The students will be able to assess state of the art methodologies in IoT application domains.
- Design IoT device to work with cloud computing infrastructure
- To get familiar with the data analytics with IoT.

**UNIT – I**

**Introduction to Internet of Things:** Introduction, Physical Design of IoT, Logical Design of IoT, IoT Enabling Technologies, IoT Levels & Deployment Templates, Domain Specific IoTs: Home Automation, Cities, Environment, Energy, Retail, Agriculture, Health & Lifestyle.

**UNIT – II**

**IoT and M2M:** Introduction to M2M, Difference between IoT and M2M, SDN and NFV for IoT.

**IoT System Management with NETCONF-YANG:** Need for IoT Systems Management, SNMP, Network Operator requirements, NETCONF, YANG, IoT Systems Management with NETCONF-YANG.

**IoT Platforms Design Methodology:** Introduction, IoT Design Methodology, Case Study on IoT system for weather Monitoring. Motivation for Using Python. Python Packages for IoT.

**UNIT – III**

**IoT Physical Devices & Endpoints:** What is an IoT Device, Exemplary Device: Raspberry Pi, About the Board, Linux on Raspberry Pi, Introduction to Arduino, Arduino Board Structure, Arduino Family.

**IoT Physical Servers & Cloud Offerings:** Introduction to Cloud Storage Models & Communication APIs, WAMP AutoBahn for IoT, Xively Cloud for IoT, Python Web Application Framework-Django, Amazon Web Services for IoT

**UNIT – IV**

**Case Studies of IoT Design:** Home Automation, Cities, Environment, Agriculture, Productivity Applications, Health Monitoring Case study.

**Introduction to Data Analytics for IoT:** Apache Hadoop, Oozie, Spark, Storm.

## **Textbook**

1. Arshdeep Bahga and Vijay Madiseti, “Internet of Things: A Hands-on Approach”, 1<sup>st</sup> Edition, Universities Press, 2015, ISBN: 978-8173719547.

## **Reference Books**

1. Graham Meikle and Mercedes Bunz, “The Internet of Things: Digital Media and Society Series”, 1<sup>st</sup> Edition, Polity Press, 2017, ISBN: 978-1509517466.
2. Rajkumar Buyya and Amir Vahid Dastjerdi, “Internet of Things: Principles and Paradigms”, 1<sup>st</sup> Edition, Morgan Kaufman Publisher, 2016, ISBN: 978-0128053959.
3. Adrian McEwen and Hakim Cassimally, “Designing the Internet of Things”, 1<sup>st</sup> Edition, Wiley Publisher, 2013, ISBN: 978-118430620.
4. Olivier Hersent, David Boswarthick and Omar Elloumi, “The Internet of Things: Key Applications and Protocols”, Student Edition, Wiley Publisher, 2015, ISBN: 978-8126557653.
5. Ovidiu Vermesan and Peter Friess, “Internet of Things Converging Technologies for Smart Environments, and Integrated Ecosystem”, River Publisher, 2013, ISBN: 978-8792982735.

## **Online Sources**

1. <https://geekflare.com/internet-of-things-iot-learning-resources/>
2. <https://www.javatpoint.com/iot-internet-of-things>

**Course Objective**

- To help students equip with the required knowledge and skill of Object-oriented programming approach needed for the development of robust web applications.
- To enable students to get hands-on experience on GUI technologies viz. AWT and Swings.
- To provide insight to students on Database Interaction, server-side components, and servlets.
- To understand the advance Java UNIT test cases for handling real world projects.

**Course Outcome**

- Able to apply AWT and Swings concept for developing GUI Framework
- Able to develop client server programming using networking concepts.
- Able to design and develop programs using Database and Servlets concepts.
- Able to develop programs for designing and testing a Maven project.

**UNIT-I**

**Abstract Window Toolkit (AWT):** Component, Container, Window, Frame, Panel, AWT Controls and Layout Managers, Event Classes and Listener Interfaces, Adapter Classes

**Swings:** Basics of Swing, JButton, JRadioButton, JTextArea, JComboBox, JTable, JColorChooser, JProgressBar, JSlider, Graphics in swing: Displaying Image, Edit Menu for Notepad, Open Dialog Box, Creating Notepad

**UNIT-II**

**Database:** Introduction to JDBC and ODBC, JDBC Architecture, Types of JDBC Drivers, Driver Interfaces and Driver Manager classes, Stored Procedures and Transaction

**Networking:** Basics of Networking, INET Address, TCP/IP Sockets, Data Grams, Simple Client Server socket programming. Remote method invocation (RMI)

**UNIT-III**

**Servlet:** Basics of Web, Servlet API, Servlet Interface, GenericServlet, HttpServlet, Servlet Life Cycle, Working with Apache Tomcat Server, Servlet in Eclipse, Servlet Request methods, Servlet Registration with DB, Servlet Request Dispatcher and send Redirect, Servlet Config, Servlet Context, Session Tracking

**JSP:** Lifecycle of JSP, JSP API, JSP in Eclipse, Scripting Elements: Scriptlet Tag, Expression Tag, Declaration Tag, Implicit Objects, MVC in JSP, Custom Tags

**UNIT-IV**

**JAVA Mail API:** Sending and receiving Emails, Sending HTML content

**JUNIT:** Introduction, Types of Testing, Annotations used in JUNIT, Assert Class, Test Cases

**Maven:** Introduction to Maven, Ant vs Maven, Maven Repository, Maven Example, Maven Web App Example, Understanding POM.XML, Maven using Eclipse

## **Textbooks**

1. Herbert Schildt, “JAVA: The Complete Reference, Twelfth Edition”, McGraw Hill Education, New Delhi, 2021, ISBN: 978-1260463415.
2. Steven Holzner, “JAVA 2 Programming (JDK 5 Edition)”, Wiley Publisher, New Delhi, 2005, ISBN: 978-8177226553.

## **Reference Books**

1. John Hunt and Alex McManus, “Key JAVA: Advanced Tips and Techniques”, Practitioner Series, Springer Verlag, 2013, ISBN: 978-1447106074.
2. Santosh Kumar K., Kogent Solutions Inc. “JDBC, Servlets and JSP Black Book”, Dreamtech Press, New Delhi, 2008, ISBN: 978-8177228373.
3. Lasse Koskela, “Effective UNIT Testing: A Guide for Java Developers”, Manning publisher, 2013, ISBN: 978-1638353881.
4. Brian Goetz, Tim Peierls, Joshua Bloch, Joseph Bowbeer, David Holmes, and Doug Lea, “Java Concurrency in Practice”, 10<sup>th</sup> Edition, Addison-Wesley Publisher, 2006, ISBN: 978-0321349606

## **Online Sources**

1. <https://www.javatpoint.com/java-awt>
2. <https://nptel.ac.in/courses/106105191>
3. <https://www.javatpoint.com/servlet-tutorial>
4. <https://www3.ntu.edu.sg/home/ehchua/programming/java/JavaServerPages.html>
5. <https://www.mygreatlearning.com/blog/java-server-pages-jsptutorial/>
6. <https://www.tutorialspoint.com/awt/index.htm>
7. <https://docs.oracle.com/javase/tutorial/uiswing/index.html>
8. <https://www.javatpoint.com/java-mail-api-tutorial>
9. <https://www.javatpoint.com/jUNIT-tutorial>
10. <https://maven.apache.org/guides/getting-started/maven-in-five-minutes.html>

**Course Objective**

- Provide an understanding of the fundamental principles in compiler design.
- Provide the skills needed for building compilers for various situations that one may encounter in a career in Computer Science.
- Learn the process of translating a modern high-level language to executable code required for compiler construction.
- To apply the optimization techniques to have a better code for code generation.

**Course Outcome**

- Familiar with fundamentals of compiler and identify the relationships among different phases of the compiler.
- Interpret the application of finite state machines, recursive descent, production rules, parsing, and language semantics.
- Analyze & implement required module, which may include front-end, back-end, and a small set of middle-end optimizations.
- Apply modern tools and technologies for designing new compiler.

**UNIT – I**

**Introduction:** Language Processors, Phases of A Compiler, A Model for A Compiler Front End, Syntax-Directed Translation, Parsing, A Translator for Simple Expressions, Lexical Analysis: Role of Lexical Analyzer, Input Buffering, Specification Of Tokens, Lexical Analyzer Generator, Data Structures In Compilation.

**Top-Down Parsing:** Introduction, Context Free Grammars, Writing A Grammar, Recursive-Descent Parsing, LL (1) Grammars, Predictive Parsing, Preprocessing Steps Required for Predictive Parsing.

**UNIT – II**

**Bottom-Up Parsing:** Shift Reduce Parsing, SLR Parsing, CLR Parsing and LALR Parsing, Error Recovery In Parsing, Handling Ambiguous Grammar, Parser Generator – YACC, Semantic Analysis: Syntax-Directed Definitions, Evaluation Order For SDD's, Application Of SDT.

**UNIT – III**

**Intermediate-Code Generation:** Syntax Trees, Three-Address Code, Types and Declarations, Translation of Expressions, Type Checking. Runtime Environment: Storage Organization, Stack Allocation of Space, Heap Management, Storage Allocation for Arrays, Strings and Records, Introduction to Garbage Collection and Trace-Based Collection

**UNIT – IV**

**Code Generation:** Issues in The Design of Code Generator, Target Language, Addresses in The Target Code, Basic Blocks and Flow Graphs, Optimization of Basic Blocks, Peephole Optimization, Register Allocation and Assignment.

**Code Optimization:** Principal Sources of Optimization, Data Flow Analysis, Constant Propagation, Partial Redundancy Elimination, Loops In Flow Graphs.

### **Textbook**

1. Alfred V. Aho, Monica S. Lam, Ravi Sethi, J. D. Ullman, “Compilers: Principles, Techniques, & Tools”, 2<sup>nd</sup> Edition, Pearson Education, 2013, ISBN: 978-9332518667.

### **Reference Books**

1. Dick Grune, Henry E. Bal, Cerial J. H. Jacobs, Koen G. Langendoen, “Modern Compiler Design”, John Wiley & Sons Publisher, 2000, ISBN: 978-0471976974.
2. Kenneth C. Louden, “Compiler Construction: Principles and Practice”, 2003, S. Chand Publisher, ISBN: 978-9812436948.
3. Thomas W. Parsons, “Introduction to Compiler Construction”, 1992, W.H. Freeman Publisher, ISBN: 978-0716782612.
4. Andrew W. Appel, “Modern Compiler Implementation in C”, Revised Edition, 2004, Cambridge University Press, 2004, ISBN: 978-0521607650.
5. John R. Levine, Tony Mason, and Doug Brown, “LEX & YACC”, 2<sup>nd</sup> Edition, Shroff Publisher, 1992, ISBN: 978-8173660627.
6. Keith D. Cooper and Linda Torczon, “Engineering a Compiler”, Morgan Kaufmann Publisher, 2003, ISBN: 978-1558606982.

### **Online Source**

1. <https://www.javatpoint.com/compiler-tutorial>

### **Course Objective**

- To understand the basic concepts and terminology related to DBMS and Relational Database Design.
- To understand the limitations of relational database over object-oriented database.
- To the design and implement Distributed Databases.
- To understand advanced DBMS techniques to construct tables and write effective queries, forms, and reports.
- To understand few emerging database models for dealing with real world problems.

### **Course Outcome**

- Demonstrate the basic elements of a relational database management system.
- Apply normalization for the development of application software.
- Design object-oriented database with advanced structural queries of SQL.
- Analyze key notions of transaction management and recovery techniques.
- Assess and apply database functions of distributed database.
- Familiar with advanced querying and decision support system in other advanced databases

### **UNIT-I**

**Relational Database Design:** Features of Good Database Design, Enhanced ER Tools, Subclasses, Super Class, and Inheritance, Specialization and Generalization, Constraints and Characteristics of Specialization and Generalization, Converting EER Diagram to Tables.

**Normalization:** Functional Dependency Theory and Normalization, Multi Value Dependency and 4NF, Join Dependency and 5NF, Inclusion Dependencies and Template Dependency, PJNF/DKNF, Modeling Temporal Data

### **UNIT-II**

**Object Oriented Database:** Limitations of Relational Database, The Need of Object-Oriented Databases, Complex Data Types, Structured Types and Inheritance in SQL, Data Types (Arrays, Multi-Set Etc) and Structure in Object Oriented Databases Using SQL, Object-Identity and Reference Types in SQL, ODL And OQL, Persistent Programming Languages, Object-Oriented Versus Object-Relational Database.

**Transaction Management and Recovery:** Advanced feature of Transactions, Enhanced Lock Based and time stamp-based Protocols, Multiple Granularity, Recovery and Atomicity, Advanced Recovery Techniques, Remote Backup Systems.

### **UNIT-III**

**Distributed Databases:** Centralised Versus Non-Centralized Databases, Homogeneous And Heterogeneous DDBMS And Their Comparison, Functions And Architecture, Distributed Database Design, Query Processing In DDBMS, Distributed Concurrency Management, Deadlock Management.



**Distributed Commit Protocols:** 2 PC and 3 PC, Concepts of replication servers.

## **UNIT-IV**

**Emerging Database Models:** Multimedia database, Geography databases, Gnome databases, Knowledge databases, deductive databases and semantic databases, Spatial database, Mobile databases, Web databases (JDBC, ODBC), Personal databases, Data grids, Digital libraries.

### **Textbook**

1. Avi Silberschatz, Henry F. Korth and S. Sudarshan, "Database System Concepts", 7<sup>th</sup> Edition, McGraw Hill Publisher, 2019, ISBN: 978-0078022159.

### **Reference Book**

1. R. Elmasri and S. B. Navathe, "Fundamentals of Database Systems", 7<sup>th</sup> Edition, Pearson Publisher, 2017, ISBN: 978-9332582705.
2. Thomas M. Connolly and Carolyn E. Begg, "Database Systems: A Practical Approach to Design, Implementation and Management", 6<sup>th</sup> Edition, Pearson Publisher, 2019, ISBN: 978-9353438913.
3. Gary W. Hansen and James V. Hansen, "Database Management and Design", 2<sup>nd</sup> Edition, Pearson Publisher, 1995, ISBN: 978-0133088007.
4. Jeffrey A Hoffer, Mary B Prescott, and Fred R. McFadden, "Modern Database Management", 8<sup>th</sup> Edition, Pearson Prentice Hall, 2006, ISBN: 978-0131566729.
5. C. J. Date, "An Introduction to Database System", 7<sup>th</sup> Edition, Pearson Publisher, 1999, ISBN: 978-0201385908.
6. David Kroenke, David Auer, Robert Yoder and Scott Vandenberg, "Database Processing: Fundamentals, Design, and Implementation", 15<sup>th</sup> Edition, Pearson Publisher, 2018, ISBN: 978-0134802749.

### **Online Source**

1. <https://www.javatpoint.com/dbms-tutorial>

**Course Objective**

- Understand basics Machine Learning algorithms and their applications.
- Understand modern notions in Supervised Machine Learning and its concepts.
- Understand and use of Unsupervised machine learning techniques.
- Understand and implement ensemble learning concepts.
- Understand and use Artificial neural network and Reinforcement learning concepts

**Course Outcome**

- Student will be familiar with the basics of Machine learning including Supervised Learning, Unsupervised Learning, Ensemble Learning, Reinforcement Learning concepts.
- Students will be able to analyze and implement Supervised Learning techniques in real life problem.
- Student will be able to analyze and implement Unsupervised machine learning techniques to solve various problem.
- Students will be able to apply advanced machine learning concepts called ensemble learning.
- Students will be familiar with the Artificial Neural network computing and their application in data science in software industry.

**UNIT – I**

**Introduction:** Machine Learning, Supervised Learning, Unsupervised Learning, Ensemble Learning, Reinforcement Learning

**Supervised Learning:** Decision Tree Induction, Naïve Bayes Classification, Rule based Classification, K-Nearest Neighbor, Performance evaluation metrics of Classifiers

**UNIT – II**

**Unsupervised Learning:** Clustering, Partitioned Clustering (K-Means), Hierarchical Clustering, BIRCH, CURE, Density based Clustering (DBSCAN). Performance evaluation metrics of Clustering and Cluster Indices

**UNIT – III**

**Regression:** Linear Regression, Multivariate Regression

**Artificial Neural Networks:** Feedforward & Feedback, Multi-Layer Feed Forward Networks, Delta Learning Rule for Multi-Perceptron Layer, Generalized Delta Learning Rule, Error Back-Propagation Training Networks, Introduction to Deep Learning

**UNIT – IV**

**Advance ML Approaches:** Ensemble Learning, Bagging, Boosting, Gradient Boosting (Random Forest, Adaptive Boosting)

## **Textbooks**

1. T. Hastie, R. Tibshirani and J. Friedman, “The Elements of Statistical Learning: Data Mining, Inference and Prediction”, 2<sup>nd</sup> Edition, Springer Publisher, 2017, ISBN: 978-0387848570.
2. Christopher M. Bishop, “Pattern Recognition and Machine Learning”, 1<sup>st</sup> Edition, Springer Publisher, 2006, ISBN: 978-0387310732.
3. Ethem Alpaydin, “Introduction to Machine Learning”, 2<sup>nd</sup> Edition, MIT Press, ISBN: 978-02620112430.

## **Reference Books**

1. Stephen Marsland, “Machine Learning: An Algorithmic Perspective”, 2<sup>nd</sup> Edition, CRC Press, 2014, ISBN: 978-1466583283.
2. Christopher M. Bishop, “Pattern Recognition and Machine Learning”, 1<sup>st</sup> Edition, Springer, ISBN: 978-1493938438.
3. Tom Mitchell, “Machine Learning”, Indian Edition, McGraw Hill Education, 2017, ISBN: 978-1259096952.

## **Online Source**

1. <https://www.javatpoint.com/machine-learning>

**Course Objective**

- Be familiar with mathematical foundations of data mining tools.
- Understand and implement classical models and algorithms in data warehouses and data mining
- Characterize the kinds of patterns that can be discovered by association rule mining, classification, and clustering.
- Master data mining techniques in various applications like social, scientific and environmental context.
- Develop skill in selecting the appropriate data mining algorithm for solving practical problems.

**Course Outcome**

- Familiar with the functionality of the various data mining and data warehousing component
- Analyzes the strengths and limitations of various data mining and data warehousing models
- Interpret the analyzing techniques of various data
- Differentiate different methodologies used in data mining and data ware housing

**UNIT-I**

**Introduction to Data Mining:** Why Data Mining? What Is Data Mining? What Kinds of Data Can Be Mined? What Kinds of Patterns Can Be Mined? Which Technologies Are Used? Major Issues in Data Mining. Getting To Know Your Data - Data Objects and Attribute Types, Basic Statistical Description of Data, Data Visualization, Measuring Data Similarity And Dissimilarity.

**Data Processing:** An Overview, Data Cleaning, Data Normalization, Data Integration, Data Reduction, Data Transformation and Data Discretization.

**UNIT-II**

**Data Warehousing and Online Analytical Processing (OLAP):** Basic Concepts of Data Warehouse, Data Warehouse Modelling–Data Cube And OLAP, Data Warehouse Design and Usage, Data Warehouse Implementation, Data Generalization By Attribute–Oriented.

**Data Cube Technology:** Data Cube Computation Preliminary Concepts, Data Cube Computation Methods, Processing Advanced Kinds of Queries By Exploring Cube Technology, Multidimensional Data Analysis In Cube Space. Mining Frequent Patterns, Associations, And Correlations – Basic Concepts, Frequent Itemset Mining Methods, Mining Various Kinds of Association Rules, From Association Mining to Correlation Analysis, Constraint-Based Association Mining.

**UNIT-III**

**Classification:** Basic Concepts, Decision Tree Induction, Bayes Classification Methods, Rule-Based Classification, Classification by Backpropagation, Support Vector Machines, Associative Classification, Lazy Learners, Other Classification Methods.

**Cluster Analysis:** Basic Concepts of Cluster Analysis, Partitioning Methods, Hierarchical Methods, Density-Based Methods, Evaluation of Clustering.

## **UNIT-IV**

**Outlier Detection:** Outliers and Outlier Analysis, Outlier Detection Methods, Statistical Approaches, Proximity-Based Approaches, Clustering-Based Approaches, Classification-Based Approaches.

**Data Mining Trends and Research Frontiers:** Mining Complex Data Types, Other Methodologies of Data Mining, Data Mining Applications, Data Mining And Society, Data Mining Trends.

### **Textbook**

1. Jiawei Han, Micheline Kamber, Jian Pei, “Data Mining – Concepts and Techniques”, 3<sup>rd</sup> Edition, Morgan Kaufmann Publisher, 2011, ISBN: 978-9380931913.

### **Reference Books**

1. Arun K Pujari, “Data Mining Techniques”, 4<sup>th</sup> Edition, The Orient Blackswan Publisher, 2016, ISBN: 978-9386235053.
2. Paweł Cichosz, “Data Mining Algorithms: Explained Using R”, Wiley Publisher, 2015, ISBN: 978-1118332580.
3. Nong Ye, “Data Mining: Theories, Algorithms, and Examples”, 1<sup>st</sup> Edition, 2014, CRC Press, ISBN: 978-0429067761.
4. Pang-Ning Tan, M. Steinbach and V. Kumar, “Introduction to Data Mining”, 1<sup>st</sup> Edition, Pearson Publisher, 2016, ISBN: 978-9332571402.
5. S. Sumathi and S. N. Sivanandam, “Introduction to Data Mining and its Applications”, Springer, 2006, ISBN: 978-3662500804.

### **Online Source**

1. <https://www.javatpoint.com/data-mining>
2. [https://www.tutorialspoint.com/data\\_mining/index.htm](https://www.tutorialspoint.com/data_mining/index.htm)
3. <https://www.tutorialride.com/data-mining/data-mining-tutorial.htm>

**Course Objective**

- Using Graphics, Animations and Multithreading for designing Simulation and Game based applications.
- Design and develop GUI applications using Abstract Windowing Toolkit (AWT), Swing and Event Handling.
- Design and develop Web applications
- Designing Enterprise based applications by encapsulating an application's business logic.
- Designing applications using pre-built frameworks.

**Course Outcome**

- Design and simulate Internet Programming using Java Applets, Graphics and Multithreading
- 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) & Swings.
- Apply event handling using AWT and Swing components.
- Develop database applications using Java Data Base Connectivity (JDBC)
- Create dynamic web pages, using Servlets and JSP.
- Develop a reusable software component, using Java Bean.

Sl. No.	List of Experiments
1	Write a program using AWT <ol style="list-style-type: none"> <li>To demonstrate the use of AWT components like Label, Textfield, TextArea, Button, CheckBox, RadioButton.</li> <li>To design a simple calculator with the use of GridLayout.</li> <li>To create a menu-bar containing items such as File, Edit, View and also create a submenu under the File menu: New and Open.</li> </ol>
2	Write a program using Swing <ol style="list-style-type: none"> <li>To display a ScrollPane, JComboBox in an JApplet with the items – English and Hindi.</li> <li>To create a JTable and JTree.</li> <li>To launch a JProgressBar.</li> </ol>
3	Write a program to demonstrate the use of <ol style="list-style-type: none"> <li>URL and URLConnection class and its methods.</li> <li>To implement chat server using ServerSocket and Socket Class.</li> <li>DataGramSocket and Datagram Packet.</li> </ol>
4	Write a program to demonstrate the use of create, insert, delete, update a record and ResultSet from database using JDBC.
5	Write a program to demonstrate <ol style="list-style-type: none"> <li>The use of HttpServlet as a Parameterized Servlet.</li> <li>To send username and password using HTML forms and authenticate the user.</li> <li>To create Session using HttpSession class.</li> <li>To implement Session tracking using Cookies.</li> </ol>
6	Write a program to send and receive e-mails using JAVA Mail API.

7	Write Test Cases using JUNIT a. To print the maximum number from an array b. To return cube of a given number c. To reverse a word
8	Write a program to build and test a Maven project.

### **Textbook**

1. Herbert Schildt, “JAVA: The Complete Reference”, 12<sup>th</sup> Edition, McGraw Hill Education, New Delhi, 2021, ISBN: 978-1260463415.

### **Reference Books**

1. Santosh Kumar K., Kogent Solutions Inc. “JDBC, Servlets and JSP Black Book”, Dreamtech Press, New Delhi, 2008, ISBN: 978-8177228373.
2. Simon Harrer, Jörg Lenhard, Linus Dietz, “Java By Comparison: Become a Java Craftsman in 70 Examples”, 1<sup>st</sup> Edition, Pragmatic Bookshelf Publisher, 2018, ISBN: 978-1680502879.
3. Lasse Koskela, “Effective UNIT Testing: A Guide for Java Developers”, Manning Publisher, 2013, ISBN: 978-1638353881.

### **Online Sources**

1. <https://docs.oracle.com/javase/tutorial/uiswing/index.html>
2. <https://www.javatpoint.com/java-mail-api-tutorial>
3. <https://www.javatpoint.com/jUNIT-tutorial>
4. <https://maven.apache.org/guides/getting-started/maven-in-five-minutes.html>

**Course Objective**

- The objective of this lab course is to understand the practical applicability of database management system concepts.
- Working on existing database systems, designing of database, creating relational database, analysis of table design using Oracle.
- The lab course also provides practical knowledge to understand advanced database concepts such as Datamining and Big Data Analysis.

**Course Outcome**

- Students get practical knowledge on designing and creating relational database systems.
- Executes various advanced queries using relational constraints, joins, views, and clauses of Oracle.
- Use of Oracle software to design and build related database systems.
- Students will be able to design and implement database applications on their own choice.

<b>Sl. No.</b>	<b>List of Experiments</b>
1	Working with Tables a. Create Table b. Alter Table c. Drop Table
2	Working on VIEW table a. Create VIEW b. Update VIEW c. Drop VIEW
3	Working with Oracle Queries a. Select b. Insert c. Update d. Delete e. Truncate
4	Working with Oracle Clauses a. DISTINCT, b. FROM, c. ORDER BY, d. GROUP BY, e. HAVING
5	Working with Oracle Operators a. Union, b. Intersect, c. Minus
6	Working with Oracle JOINS a. Inner Join, b. Outer Join, c. Equi Join, d. Self Join
7	Working with Oracle advance commands a. Procedure, b. Function, c. Trigger
8	Working with Oracle ALIASES, AND, OR, NOT, IS NULL



## **Textbook**

1. Kim Berg Hansen, “Practical Oracle SQL: Mastering the Full Power of Oracle Database”, Springer India, 2020, ISBN: 978-1484276075.

## **Reference Books**

1. Steven Feuerstein and Bill Pribyl, “Oracle PL/SQL Programming: Covers Versions Through Oracle Database 12c”, 6<sup>th</sup> Edition, O’Reilly Publisher, 2014, ISBN: 978-1449324452.
2. P.S. Deshpande, “SQL & PL/SQL for Oracle 11g Black Book”, Dreamtech press, 2011, ISBN: 978-8177229400.

## **Online Source**

1. <https://www.javatpoint.com/oracle-tutorial>

## Semester-4<sup>th</sup>

Paper No	Subject	Title of the Paper	Credit	Max. Marks		
				Internal	Univ. Exam	Total
CS-502	Core	Advanced Software Engineering	4	20	80	100
CS-504	Elective-III	Introduction to Deep Learning or Pattern Recognition or Cloud Computing or Digital Image Processing	4	20	80	100
CS-506		Dissertation & Viva Voce	8			100
<b>Total</b>			<b>16</b>	<b>Total Marks</b>		<b>300</b>

**Course Objective**

- Capable to acquire the generic software development skill through various stages of software life cycle.
- To apply principles of software development and evolution.
- To specify, abstract, verify and validate solutions to large-size problems.
- To plan, develop and manage large software and learn emerging trends in software engineering.
- To ensure the quality of software through software development with various protocol-based environment.

**Course Outcome**

- Students should be able to identify the need for engineering approaches to software development and various processes of requirements analysis for software engineering problems.
- Analyse various software engineering models and apply methods for design and development of software projects.
- Work with various techniques, metrics, and strategies for Testing software projects.
- Identify and apply the principles, processes, and main knowledge areas for Software Project Management
- Proficiently apply standards, CASE tools and techniques for engineering software projects.

**UNIT-I**

**Introduction:** FAQs about Software Engineering; Professional and Ethical Responsibility.

**Software Process:** Models; Process Iteration, Specification, Software Design, and Implementation; Verification & Validation; Software Evolution; Automated Process Support.

**Software Design:** Object-Oriented Design Process, Design Evolution; Real Time Software Design: Systems Design, Real-Time Executives, Monitoring and Control Systems, Design with Reuse: Component-Based Development, Application Families, Design Patterns; User Interface Design: Principles, User Interaction, Information Presentation, User Support, Interface Evaluation.

**UNIT-II**

**Project Management:** Management Activities, Software Project Management and Requirements, Project Planning, Project Scheduling, Risk Management.

**Software Requirements:** Functional and Non-Functional Requirements, User Requirements, System Requirements, Requirements Document, Requirements Engineering Process: Feasibility Studies, Requirements Elicitation and Analysis, Requirements Validation, Requirements Management.

**Verification and Validation (V & V):** Static and Dynamic V & V, V & V Goals, V & V vs. Debugging, Software Inspections / Reviews, Clean-Room Software Development.

**Software Testing:** Defect Testing, Integration Testing, Interface Testing, Object-Oriented Testing, Testing Work Benches.

### **UNIT-III**

**Software Cost Estimation:** Productivity, Estimation Techniques, Algorithmic Cost Modelling, Project Duration and Staffing. **Quality Management:** Quality Assurance and Standards, Quality Planning, Quality Control, Software Measurement and Metrics; **Process Improvement:** Process and Product Quality, Process Analysis and Modelling, Process Measurement, the SEI Process Maturity Model, and Process Classification

**Reengineering:** Business Process Reengineering, Software Reengineering, Reverse Reengineering, Restructuring, Forward Reengineering, Economics of Reengineering.

### **UNIT-IV**

**Clean Room Software Engineering:** Approach, Functional Specification, Design and Testing. **Component-Based Software Engineering:** CBSE Process, Domain Engineering, Component-Based Development, Classifying and Retrieving Components, And Economics Of CBSE. **Client/Server Software Engineering:** Structure of Client/Server Systems, Software Engineering for Client/Server Systems, Analysis Modelling Issues, Design And Testing Issues.

**Computer-Aided Software Engineering:** Building Blocks and Taxonomy For CASE, Integrated CASE Environments, Integration Architecture, CASE Repository, Case Study of Tools Like TCS Robot.

### **Textbook**

1. Roger S. Pressman and Bruce R. Maxim, “Software Engineering: A Practitioner’s Approach”, 8<sup>th</sup> Edition, McGraw-Hill Education, 2014, ISBN: 978-0078022128.

### **Reference Books**

1. Jonathon Bowen, “Formal Specification and Documentation using Z: A Case Study Approach”, Cengage Learning, 1995, ISBN: 978-1850322306.
2. Robert Oshana and Mark Kraeling, “Software Engineering for Embedded Systems: Methods, Practical Techniques, and Applications”, 2<sup>nd</sup> Edition, Newnes Publisher, 2019, ISBN: 978-0128094488.
3. J. F. Peters and W. Pedrycz, “Software Engineering: An Engineering Approach”, John Wiley and Sons Publisher, 1999.
4. Ali Behforooz, and Frederick J.Hudson, “Software Engineering Fundamentals”, 1<sup>st</sup> Edition, Oxford University Press, 2012, ISBN: 978-0198090502.
5. Carlo Ghezzi, Mehdi Jazayeri and Dino Mandrioli “Fundamentals of Software Engineering” 2<sup>nd</sup> Edition, Pearson Publisher, 2002, ISBN: 978-0133056990.

### **Online Source**

1. <https://www.javatpoint.com/software-engineering-tutorial>

**Course Objective**

- Understands the technical foundations of cloud systems architectures.
- Analyze the problems and solutions to cloud application problems.
- Apply principles of best practice in cloud application design and management.
- Identify and define technical challenges for cloud applications and assess their importance.

**Course Outcome**

- Familiar with the core concepts of the cloud computing paradigm
- Apply the fundamental concepts in datacenters to understand the tradeoffs in power, efficiency, and cost.
- Identify resource management fundamentals
- Assess the importance of virtualization in distributed computing
- Analyze various cloud programming models and apply them to solve problems on the cloud.

**UNIT-I**

**Era of Cloud Computing (CC):** Introduction, CC vs. Peer-To-Peer Architecture, CC vs. Client-Server Architecture, Components of CC, Impact of CC on Businesses.

**Introduction to Virtualization:** Introduction, Virtualization Benefits, Implementation Levels of Virtualization, Open-Source Virtualization Technology, Virtualization of CPU, Memory, I/O Devices, Hardware Support for Virtualization, Virtualization in Multicore Processors.

**Cloud Computing Services:** IaaS, PaaS, Leveraging PaaS for Productivity, Concerns with PaaS, Languages and PaaS, SaaS, DBaaS.

**Cloud Types and Models:** Private Cloud, Public Cloud, Hybrid Cloud.

**UNIT-II**

**Open-Source Cloud Implementation and Administration:** Eucalyptus & OpenStack Cloud Architectures, Cloud Standards, Cloud Ratings.

**Data Security in the Cloud:** Challenges with Cloud Data and Data Security, Data Confidentiality and Encryption, Data Availability, Data Integrity, CSRS. Cloud Application Requirements, SOA for Cloud Applications.

**UNIT-III**

**Adoption and Use of Cloud by Small and Medium Businesses:** Place of Adoption, Benefits, Adoption Phases, Vendor Roles and Responsibilities, Selection Phases, Provider Liability, Provider Capabilities, Success Factors for CC Adoption Process of Public Clouds by Enterprises.

**SLA with Cloud Service Providers:** Concept, Aspects and Requirements of SLA, Credit Calculation.

**Risks, Consequences, and Costs for Cloud Computing:** Introduction, Risk Assessment and Management, Risk of Vendor Lock-In, Loss of Control, Not Meeting Regulatory Compliances,

Malware and Internet Attacks, Management of Cloud Resources, Cloud Costs, Cost Allocations, Chargeback Models and Methodology.

#### **UNIT-IV**

**AAA Administration for Cloud:** AAA model, Industry Implementation for AAA, Authentication Management in The Cloud.

**Security as a Service:** Benefits of Security as a Service, Concerns with Security as a Service, Security Service Providers, IdMaaS, attributes of IdMaaS providers.

**Application Security in the Cloud:** Cloud Application SDLC, Cloud Service Reports by Providers, Application Security in IaaS, PaaS, and SaaS Environments.

#### **Textbook**

1. K. Jayaswal, J. Kallakurchi, D. J. Houde, and D. Shah, “Cloud Computing: Black Book”, Dreamtech Press, 2014, ISBN: 978-9351194187.

#### **Reference Books**

1. R. Buyya, J. Broberg and A. Goscinski, “Cloud Computing: Principles and Paradigms”, 1<sup>st</sup> Edition, Wiley Publisher, 2013, ISBN: 978-8126541256.
2. A. Bahga and V. Madiseti, “Cloud Computing: A Hands-on Approach”, 1<sup>st</sup> Edition, The Orient Blackswan Publisher, 2014, ISBN: 978-8173719233.
3. David E.Y. Sarna, “Implementing and Developing Cloud Computing Applications”, 1<sup>st</sup> Edition, Auerbach Publisher, 2018, ISBN: 978-1138382015.
4. Kai Hwang, G. C. Fox and J. J. Dongarra, “Distributed and Cloud Computing: From Parallel Processing to Internet of Things”, 1<sup>st</sup> Edition, Elsevier, 2012, ISBN: 978-9381269237.

#### **Online Source**

1. <https://www.javatpoint.com/cloud-computing-tutorial>
2. [https://www.tutorialspoint.com/cloud\\_computing/index.htm](https://www.tutorialspoint.com/cloud_computing/index.htm)

The total allotted marks 100 are divided into the following way

**Internal Assessment (50 marks)**

- First seminar (25 marks – in between 25 to 30 days after commencement of class work)
- This seminar includes the study of existing system, literature survey, problem definition.
- Second seminar (25 marks – in between 55 to 60 days after commencement of class work)
- This seminar includes the requirements specification, analysis, design and partial implementation.

**External Assessment (50 marks)**

- The students should submit one page of synopsis on the Dissertation work for display on the notice board.
- The Dissertation presentation is for 15 minutes followed by 05 minutes for discussion.
- The student should submit a technical write-up on the Dissertation.
- At least two faculties will be associated with the project seminar to evaluate students for the award of
- Sessional marks which will be based on performance in all the 3 items (synopsis, presentation, technical write-up).

**The Dissertation seminar presentation should include the following components of the project**

- Problem definition and specification.
- Literature survey (familiarity with research journals).
- Broad knowledge of available techniques to solve a particular problem.
- Planning of the work, preparation of bar(activity) charts.
- Presentation-oral and written.

The project report should be in the following format

Title Page

Certificates by HoD, Guide (s)

Declaration

Acknowledgements

Abstract

Contents

List of Figures

List of Tables

Chapters:

1. Introduction (with problem statement, chapterization)
2. Literature Survey /Related Work
3. Background Study
4. Proposed Methodology
5. Results and Discussion
6. Conclusion and Future Work
7. References

References and Bibliography should be written in the format given below:

Author(s) Title of the Paper, Publisher, Volume No., Issue No., Year

\*Student may refer MLA/APA format to cite the references.

Text format: Font Type: Time New Roman; Font Size: 12; Line Space: 1 ½

Pages, Figures, Tables and Algorithms should be titled and numbered

The report should be organized into chapters, chapter into sections, sections into subsections etc. Hierarchical numbering should be followed in numbering the chapters, sections, subsections etc. (1, 1.1, 1.1.1), three (3 copies of the project report hard bound should be submitted to the department).



