Department of Computer Engineering

(NAAC Accredited)

S. Y. B. Tech. Course Book

(2020 Pattern)

(With effect from June 2021)

Department of
Computer Engineering

(NAAC Accredited)

Under Graduate (UG) Course Book

S.Y. B. Tech (Computer Engg.)

Semester- III/IV

INDEX

About Computer Engg. Department

- NAAC Accredited Computer Engg.Programme
- Involvement of Experts from IITs, NITs, Govt. Colleges, Reputed Industries, Alumni and Students in development of curriculum
- Choice Based Credit System(CBCS)
- Choice ofElectives
- Remedial Teaching
- Sponsorship for Publications andIPR
- ResearchMentorship
- IndustryInternship
- Provision of Credit Transfer Scheme (CTS)
- Peer Teaching Scheme
- Teacher Guardian Scheme (TGS)
- Various Clubs and HobbyModules
- ProficiencyCourses
- Recognized Research Centre under SavitribaiPhule PuneUniversity
- Industry SupportedLabs.
- MOUs withIndustries.

INSTITUTE VISION AND MISSION

VISION

To achieve excellent standards of quality education by keeping pace with rapidly changing technologies and create technical manpower of global standards with capabilities of accepting new challenges

MISSION

Our efforts are dedicated to impart quality and value based education to raise satisfaction level of all stake-holders. Our strength is directed to create competent professionals. Our endeavor is to provide all possible support to promote research and development activities

DEPARTMENT VISION AND MISSION

VISION

To produce global standards ethical professionals, innovators, and entrepreneurs having strong knowledge and urge to learn latest technologies in the field of Computer Engineering.

MISSION

The department continuously strives to:

M1: Pursue excellence in Computer Engineering, able to adapt changing technologies through effective teaching-learningprocess.

M2: Develop competent professionals for global market with the spirit of self-study, team work, innovation and ethics.

M3: Promote continuous learning, entrepreneurial skills and research.

Programme Educational Objectives (PEOs)

PEO1: Capability to analyze, design and develop cost effective solutions to the real life problems by applying the acquired knowledge.

PEO2: Adoptability to learn latest technological advancement and interdisciplinary approaches by engaging in lifelong learning process.

PEO3: Willingness to pursue higher education, entrepreneurship, and research in the field of Computer Engineering.

PEO4: Being responsible towards society, environment, and ethical responsible team member with interpersonal and leadership skill.

Program Specific Objectives (PSOs)

At the end of the programme students will be able to demonstrate:

PSO1: Theabilitytoanalyze,designanddevelopsoftwaresystems applying theknowledge acquired in computer core courses such as Operating system, database, computer network, computer organization and architecture, software engineering.

PSO2: The utilization of skills assimilated in basic Computer Engineering Courses to build up expertise in advanced areas of Database, Networking such as WSN, VANET, MANET, IoT, Computing etc.

PSO3: Oneself as a global standard computer professional with good morals, ethics and sensitivity towards mankind and as a responsible team member.

Program Outcomes (POs)

Engineering Graduates will be able to:

1.Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineeringsciences.

3.Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4.Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide validconclusions.

5.Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of thelimitations.

6.**The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineeringpractice.

7.Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainabledevelopment.

8.Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineeringpractice.

9.Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinarysettings.

10.Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clearinstructions.

11Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinaryenvironments.

12.**Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technologicalchange.

Scheme of B. Tech. in Computer Engineering

Course Code	Name of Course	Course Category	Teaching Scheme				Credits	Evaluation Scheme					
								Theory			Practical		
			L	T	P	Total Hours		TAE	CAE	ESE	INT	EXT	Total Marks
SEMESTER-III													
UBSL205A	Discrete Mathematics and Graph Theory	BS6	3	1	-	4	4	10	15	50	-	-	75
UCOL201/ UCOP201	Data Structures and Algorithms	C6	3	-	2	5	4	10	15	50	25	25	125
UCOL202	Computer Architecture and Organization	C7	3	-	-	3	3	10	15	50	-	-	75
UCOL203	Formal Languages and Automata	C8	3	-	-	3	3	10	15	50	-	-	75
UITL201/ UITP201	Object Oriented Programming	C9	3		2	5	4	10	15	50	25	25	125
UCOP204	Python for Data Science	A6-A7-A8			6	6	3		-	-	50		50
TOTAL			**15**	**1**	**10**	**26**	**21**	**50**	**75**	**250**	**100**	**50**	**525**

Department of Computer Engineering

Detailed Syllabus

S. Y. B. Tech. Semester-III

UBSL205A: Discrete Mathematics and Graph Theory		
Teaching Scheme:	**Credit:**	**Examination Scheme:**
Lectures: 03 Hrs./Week **TU :1 Hrs**	**4**	**TAE:** 10 Marks,**CAE:** 15 Marks, **ESE :50** Marks

Prerequisite (If any):

Course Objectives: After completing this course, student's will be able to

1. This course introduces size and kind of objects.
2. It also skills to analyze objects meeting the criteria, finding "largest", "smallest", or "optimal" objects.
3. It also introduces combinatorial structures and apply algebraic techniques to combinatorial problems.

Course Outcomes: After completing this course, students will be able to

CO1: Describe the fundamental concepts of discrete mathematics to solve the engineering problems.

CO2: Explain basic terminology, set, relations, functions, groups and rings.

CO3: Solve problems based on graphs, trees and related algorithms.

CO4: Relate, interpret and apply the concepts to various areas of computer science.

Course Outcomes	Program Outcomes and Program Specific Outcomes														
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO3
CO1	3	3	3	2	1	2			1		1	2			1
CO2	3	3	3	2	1	1			1		1	2			1
CO3	3	3	3	2	1	1			2		2	2	2	2	1
CO4	3	3	3	2	1	1			2		2	2	2	2	1

Course Contents	**Hrs.**
Unit I: Set Theory:	**8**
Operations on sets, Laws of algebra of sets, Representation of sets on computer in terms of 0's & 1's. Partition & covering of a set, ordered pair, Product set, Relation–Different types of relations, Graph of relation, Matrix of relation, Transitive closure of relation, Functions, Partial ordering & partially ordered set, Hasse diagram of Poset, totally ordered set, Peano axioms & Mathematical Induction.	
Unit II: Group	**8**
Modular arithmetic, Basic Prime number theory, congruence's, Residue classes & Fermat's theorem**,** Algebra or Algebraic systems like semigroup, monoid and examples. Homomorphism, Isomorphism of semigroup monoid. Groups, properties of algebraic groups. Permutations groups, Subgroups, Lagrange's theorem, properties of cyclic groups, generator of group, Cosets,Normal Subgroup ,quotient group	

Unit III: Rings	**8**
Rings, types of rings, Fields, subring, Integral domain. Simple properties of rings. Lattice as Poset& as algebraic system, Types of lattices, Hasse diagrams, Sub lattice, direct product of Lattices, Lattice Homomorphism, complement of elements of lattices, Various lattices, composition tables	
Unit IV: Graph Theory	**8**
Graphs and its types, Sub graph, Quotient graph, Euler path, complete path, in degree, out degree, reachability, cycle, matrix representation of graph. Adjacency matrix, Graph coloring, shortest path problems, Trees, Representation of trees, binary trees, spanning trees, Kruskal's and Prim's Algorithm for minimum spanning tree	
Unit V:Counting	8
Basics of counting techniques, Pigeonhole principle, Definition of generating functions and examples, Recurrence relations: definitions & examples, Solving Linear Recurrence Relations ,Inclusion and Exclusion principle	

Text Books	1.	"Discrete Mathematics and Its Applications", Kenneth H. Rosen, 7th Edition, Tata McGraw-Hill, 2017, ISBN: 9780073383095.
	2.	"Elements of Discrete Mathematics", C. L. LIU, 4th Edition, Tata McGraw-Hill, 2017, ISBN-10: 1259006395 ISBN-13: 978125 9006395.
	3.	C. L. Liu, ―Elements of Discrete Mathematics‖, TMH, ISBN 10:0-07-066913-9
Reference Books	1.	"Discrete Mathematical Structures", G. Shanker Rao, 2 nd Edition2009, New Age International, ISBN-10: 8122426697, ISBN-13: 9788122426694
	2.	"Discrete Mathematical Structures", B. Kolman, R. Busby and S. Ross, 4th Edition,Pearson Education, 2002, ISBN: 8178085569
	3.	"Discrete Mathematics", Lipschutz, Lipson, 2nd Edition, 1999, Tata McGraw-Hill, ISBN: 007 463710X
	4.	"Discrete Mathematics", R. K. Bisht, H. S. Dhami, Oxford University Press, ISBN: 9780199452798
	5.	Bernard Kolman, Robert C. Busby and Sharon Ross, ―Discrete Mathematical Structures‖, Prentice-Hall of India /Pearson, ISBN: 0132078457, 9780132078450
on line TL Material	1.	https://onlinecourses.nptel.ac.in/noc20_cs37/unit?unit=41&lesson=42

UCOL201:Data Structures and Algorithms		
Teaching Scheme:	**Credit:**	**Examination Scheme:**
Lectures: 03 Hrs./Week	**3**	**TAE:** 10 Marks,**CAE:** 15 Marks, **ESE :50** Marks

Prerequisite (If any):

Course Objectives: After completing this course, student's will be able to

1. This course introduces basic idea of data structure while making aware of methods and structure used to organize large amount of data.
2. It's also aimed at developing skill to implement methods to solve specific problems using basic data structures.
3. The course also provides career opportunities in design of data, implementation of data, technique to sort and searching the data.

Course Outcomes: After completing this course, students will be able to

CO1: Understand fundamentals of data structures

CO2: Apply searching and sorting techniques in various applications

CO3: Implement different linear data structures to solve real world problems.

CO4: Design and analyze non-linear data structure to find solution for different applications.

Course Outcomes	Program Outcomes and Program Specific Outcomes														
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO3
CO1	3	2	3	2	1						2	2	2	2	
CO2	3	3	3	2	2						1	3	3	2	
CO3	3	3	3	3	2			2	2	2	2	3	3	2	1
CO4	3	3	3	3	2			2	2	2	2	3	3	3	1

Course Contents	**Hrs.**
Unit I: Introduction	**8**
Introduction –Common operations on data structures, Types of data structures, Data structures & Programming, Program Design, Complexities, Time Complexity, order of Growth, Asymptotic Notation. **Sorting and Searching** Introduction, Sorting, Insertion Sort, Selection Sort, Merging, Merge-SortShell Sort, Radix Sort, Searching and Data Modification, Hashing	
Unit II: Arrays and Link List	**8**
Arrays: Introduction, Linear Arrays, Arrays as ADT, Representation of Linear array in Memory, Traversing Linear Arrays, Inserting and deleting, Sorting; Bubble Sort, Searching; Linear Search, Binary Search, : **Linked List**: Introduction: Linked Lists, Representation of Linked Lists in Memory, Traversing a Linked List, Searching a Linked List, Memory Allocation; Garbage Collection, Insertion into a Linked List, Deletion from a Linked List, Header Linked List, Circularly Linked Lists, Two-Way Lists (or Doubly Linked Lists).	

Unit III: Stacks, Queue and Recursion	**8**
Stacks, Queue and Recursion- Introduction, Stacks ,Array Representation of Stacks ,Linked Representation of Stacks, Stack as ADT, Arithmetic Expression; Polish Notation, Application of Stacks, Recursion, Towers of Hanoi, Implementation of Recursive Procedures by Stacks, Queue, Linked Representation of Queues, Queues as ADT, Circular Queues, Deques, Priority Queues, Applications of Queues	
Unit IV: Trees and Binary Trees	**8**
Trees and Binary Trees -Binary Trees • Representation, Operations: Insert, Delete, Traversal: Preorder, Inorder, Postorder,Traversal Algorithms Using Stacks, Header Nodes; Threads, Threaded Binary Trees, Binary Search Trees ,Searching and Inserting in Binary Search Trees, Deleting in a Binary Search Tree, Balanced Binary Trees, AVL Search Trees, Insertion in an AVL Search Tree, Deletion in an AVL Search Tree, m-way Search Trees ,Searching, Insertion and Deletion in an m-way Search tree, B-Trees ,Searching, Insertion and Deletion in a B-tree, B+-Trees Graph Algorithms	
Unit V: Graphs and their Applications	8
Graphs and their Applications - Introduction, Graph Theory Terminology, Sequential Representation of Graphs, Adjacency Matrix; Path Matrix, Linked Representation of a Graph, Operations on Graphs, Traversing a Graph, Posets; Topological Sorting, Spanning Trees	

Text Books	1.	AVAho, J Hopcroft, JD Ullman, Data Structures and Algorithms, Addison- Wesley, 1983.
	2.	THCormen, CF Leiserson, RL Rivest, C Stein, Introduction to Algorithms, 3rd Ed., MIT Press, 2009.
	3.	Sahni, S., "Data Structures, Algorithms, and Applications in C++", WCB/McGraw-Hill.
E--Books	1.	https://apps2.mdp.ac.id/perpustakaan/ebook/Karya%20Umum/Dsa.pdf
Referenc e Books	1.	Data Structures & Algorithms, 1e, Alfred V.Aho, Jeffery D. Ullman , Person.
	2.	MT Goodrich, R Tamassia, DM Mount, Data Structures and Algorithms in Java, 5th Ed., Wiley, 2010. (Equivalent book in C also exists.)
	3.	Wirth, N., "Algorithms and Data Structures", Prentice-Hall of India.
Online TL Material	1.	https://nptel.ac.in/courses/106/102/106102064/
	2.	http://cse01-iiith.vlabs.ac.in/
	3.	https://ds2-iiith.vlabs.ac.in/data-structures-2/

UCOP201:Data Structures and Algorithms lab		
Teaching Scheme:	**Credit**	**Examination Scheme**
Practical: 02 Hrs./Week	**1**	**INT :25 Marks Ext :25 Marks**

Course Outcomes: After completing this course, students will be able to

CO1: Apply searching and sorting techniques in various applications

CO2: Implement different linear data structures to solve real world problems.

CO3: Design and analyze non-linear data structure to find solution for dbifferent applications.

Course Outcomes	Program Outcomes and Program Specific Outcomes														
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO3
CO2	3	3	3	2	2			2	2	2	1	3	3	2	2
CO3	3	3	3	3	2			2	2	2	2	3	3	2	2
CO4	3	3	3	3	2			2	2	2	2	3	3	3	2

Sr.No	List of Laboratory Assignments(*Any 8)
1	Consider a student database of SY COMP class (at least 10 records). Database contains different fields of every student like Roll No, Name and SGPA.(array of objects of class) a) Design a roll call list, arrange list of students according to roll numbers in ascending order (Use Bubble Sort) b) Arrange list of students alphabetically. (Use Insertion sort) c) Arrange list of students to find out first ten toppers from a class. (Use Quick sort) d) Search students according to SGPA. If more than one student having same SGPA, then print list of all students having same SGPA. e) Search a particular student according to name using binary search without recursion.
2	Department of Computer Engineering has student's club named 'COMET'. Students of Second, third and final year of department can be granted membership on request. Similarly one may cancel the membership of club. First node is reserved for president of club and last node is reserved for secretary of club. Write program to maintain club member's information using singly linked list. Store student MIS registration no. and Name. Write functions to a) Add and delete the members as well as president or even secretary. b) Compute total number of members of club c) Display members d) Display list in reverse order using recursion e) Two linked lists exists for two divisions. Concatenate two lists
3	Implement Stack using a linked list. Use this stack to perform evaluation of a postfix expression.
4	Queues are frequently used in computer programming, and a typical example is the creation of a job queue by an operating system. If the operating system does not use priorities, then the jobs are processed in the order they enter the system. Write a program for simulating job queue. Write functions to add job and delete job from queue.
5	A double-ended queue (deque) is a linear list in which additions and deletions may be made at either end. Obtain a data representation mapping a deque into a one-dimensional array. Write a program to simulate deque with functions to add and delete elements from either end of the deque
6	A book consists of chapters, chapters consist of sections and sections consist of subsections. Construct a tree and print the nodes. Find the time and space requirements of your method.
7	Implement binary tree using linked list and perform recursive traversals.

8	Beginning with an empty binary search tree, Construct binary search tree by inserting the values in the order given. After constructing a binary tree i. Insert new node ii. Find number of nodes in longest path iii. Minimum data value found in the tree iv. Change a tree so that the roles of the left and right pointers are swapped at every node v. Search a value
9	Implement graph using adjacency list or matrix and perform DFS or BFS.
10	You have a business with several offices; you want to lease phone lines to connect them up with each other; and the phone company charges different amounts of money to connect different pairs of cities. You want a set of lines that connects all your offices with a minimum total cost. Solve the problem by suggesting appropriate data structures.
	Open Ended Experiments / New Experiments
11	A classic problem that can be solved by backtracking is called the Eight Queens problem, which comes from the game of chess. The chess board consists of 64 square arranged in an 8 by 8 grid. The board normally alternates between black and white square, but this is not relevant for the present problem. The queen can move as far as she wants in any direction, as long as she follows a straight line, Vertically, horizontally, or diagonally. Write C++ program for generating all possible configurations for 4-queen's problem.
12	A Dictionary stores keywords & its meanings. Provide facility for adding new keywords, deleting keywords, updating values of any entry. Provide facility to display whole data sorted in ascending/ Descending order. Also find how many maximum comparisons may require for finding any keyword. Use Binary tree and find the complexity for finding a keyword.

UCOL202:Computer Architecture and Organization		
Teaching Scheme:	**Credit:**	**Examination Scheme:**
Lectures: 03 Hrs./Week	**3**	**TAE:** 10 Marks,**CAE:** 15 Marks, **ESE :50** Marks

Prerequisite (If any):

Course Objectives: After completing this course, student's will be able to

1. To understand the design principles of digital computing systems
2. To provide essential understanding of different subsystems of modern computer system and design aspects these subsystems3
3. To provide overview on performance enhancement methods in instruction execution

Course Outcomes: After completing this course, students will be able to

CO1: To describe the basic components and design of a computer system

CO2: To implement basic binary math operations in computers.

CO3: To apply the concept of various memories and interfacing technologies

CO4: To analyze the different parallel processing technique and high performance computing architecture

Course Outcomes	Program Outcomes and Program Specific Outcomes														
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	2	2	1	1			1	1	1	2	3		
CO2	2	3	2	3	1	1			1	1			2		
CO3	2	3	2	2	2	1			1	2	2	2	2		
CO4	3	3	3	3	2	2			1	1	3	3	3		

Course Contents	Hrs.
Unit I: Introducation	**8**
Basic functional blocks of a computer: CPU, memory, input-output subsystems, control unit. Instruction set architecture of a CPU –registers, instruction execution cycle, RTL interpretation of instructions, addressing modes, instruction set, Instruction set architecture CISC, RISC, Case study –instruction sets of common CPUs	
Unit II: Data Processsing	**8**
Fixed point Addition, Subtraction, Multiplication and Division. Floating Point arithmetic, High performance arithmetic, Sub word parallelism, Booth's algorithm, integer division • Data representation method • Booths multiplication, division algorithm and example IEEE standard single and double precision format and examples	
Unit III: Memory Orgnization	**8**
Memory hierarchy, Memory Chip Organization, Cache memory, Virtual memory. Parallel Bus Architectures, Internal Communication Methodologies, Serial Bus Architectures, Mass storage, Input and Output Devices, Segmentation, TLB, Page	

replacement algorithms	
Unit IV: Pipelining	**8**
Pipelining: Basic concepts of pipelining, throughput and speedup, pipeline hazards Pipelining: Basic concepts of pipelining, Arithmetic and Instruction Pipeline, throughput and speedup, pipeline hazards, Introduction, Logic Design Conventions, Building a Datapath – A Simple Implementation scheme – An Overview of Pipelining – Pipelined Datapath and Control. Data Hazards: Forwarding versus Stalling, Control Hazards, Exceptions, Parallelism via Instructions	
UNIT V:CPU control unit design	8
CPU control unit design: hardwired and micro-programmed design approaches, Case study - design of a simple hypothetical CPU.	

Text Books	1.	Introduction to Parallel Computing by AnanthGrama, Anshul Gupta, George Karypis, Vipin Kumar in Pearson Publication
	2.	Advance computer Architecture by Kai Hwang under Tata McGraw Hill publications
	3.	Introduction to Parallel Processing: Algorithms & Architectures, BehroozParhami in Springer Shop
EBooks	1.	Computer Architecture and Organization by William Stalling http://home.ustc.edu.cn/~leedsong/reference_books_tools/Computer%20Organization%20and%20Architecture%2010th%20-%20William%20Stallings.pdf
Reference Books	1.	Introduction to Parallel Processing by P. Ravi Prakash, M. Sasikumar, Dinesh Shikhare By PHI Publications
	2.	Fundamentals of Parallel Processing by Jordan Harry, Alaghband Gita, PHI Publication
	3	Parallel Computers – Architecture and Programming by V. Rajaraman And C. Siva Ram Murthy.
	4.	Introduction to Parallel Programming by Steven Brawer
Online TL Material	1.	NPTL https://nptel.ac.in/courses/106/105/106105163/

UCOL203: Formal Languages and Automata		
Teaching Scheme:	**Credit:**	**Examination Scheme:**
Lectures: 03 Hrs./Week	**3**	**TAE:** 10 Marks,**CAE:** 15 Marks, **ESE :50** Marks

Prerequisite (If any):

Course Objectives: After completing this course, student's will be able to

1. To provide introduction to some of the central ideas of theoretical computer science from theperspective of formallanguages.
2. To introduce the fundamental concepts of formal languages,grammars and automatatheory.
3. Classify machines by their power to recognize languages and usefinite state machines to solve problems incomputing.
4. To understand deterministic and non-deterministicmachines.
5. Use of Turing Machine and Pushdown Automata in FormalLanguage.

Course Outcomes: After completing this course, student's will be able to

CO1 Understand the abstract machines and modeling of finite state machines .

CO2 Design context free grammars to analyze formal languages and computing problems.

CO3 Apply Formal language to analyze problems based on push down automata.

CO4 Solve problems based on linear bounded automata and Turing machine.

Course Outcomes	Program Outcomes and Program Specific Outcomes														
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3
CO1	3	2										1	3		
CO2	2	2	3	2					1	2		1	2		
CO3	3	3	2	2					1	2		1	2		
CO4	3	3	3	2					1	2		1	3		

Course Contents	Hrs.
Unit I: Introduction	**8**
Introduction-BasicMathematicalNotationandtechniques-FiniteStatesystems–Basic Definitions – Finite Automaton – DFA & NDFA – Finite Automaton with €- moves – RegularLanguages-RegularExpression –EquivalenceofNFAandDFA–Equivalence ofNDFA"s with and without €-moves – Equivalence of finite Automaton.	
Unit II: Regular Expressions	**8**
Regular Expressions- Finite Automata and Regular Expressions, Applications of Regular Expressions, Algebraic Laws for Regular Expressions, Properties of Regular Languages Pumping Lemma for Regular Languages, Applications of the Pumping Lemma, Closure Properties of Regular Languages, Decision Properties of Regular Languages.	

Unit III: Context-FreeGrammars	**8**
Context-FreeGrammars:Chomskyhierarchyoflanguages.DefinitionofContext-Free Grammars, Derivations Using a Grammar, Leftmost and Rightmost Derivations, the Language of a Grammar, Sentential Forms, Parse Tress, Applications of Context-Free Grammars, Ambiguity in Grammars and Languages.	
Unit IV: Push Down Automata-	**8**
Push Down Automata- Definition of the Pushdown Automaton, the Languages of a PDA, Equivalence of PDA's and CFG's, Deterministic Pushdown Automata.	
UNIT V:	8
Definitions of Turing machines – Models – Computable languages and functions – Techniques for Turing machine construction – Multi head and Multi tape Turing Machines – The Halting problem	

Text Books	1	Introduction to Automata Theory, Languages, and Computation, 3nd Edition, John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, Pearson Education.
	2	Introduction to the Theory of Computation, Michael Sipser, 3rd edition, Cengage Learning.
Reference Books	1	Introduction to Languages and The Theory of Computation, John C Martin, TMH.
	2	Introduction to Computer Theory, Daniel I.A. Cohen, John Wiley.
	3	A Text book on Automata Theory, P. K. Srimani, Nasir S. F. B, Cambridge University Press.
	4	Introduction to Formal languages Automata Theory and Computation Kamala Krithivasan, Rama R, Pearson.

UITL201:Object Oriented Programming		
Teaching Scheme:	**Credit:**	**Examination Scheme:**
Lectures: 03 Hrs./Week	**3**	**TAE:** 10 Marks,**CAE:** 15 Marks, **ESE :50** Marks

Prerequisite (If any):

Course Objectives: After completing this course, student 's will be able to

1. This course introduces student's general idea and concepts of object oriented programming.
2. It is also aimed at developing skills to implement these concepts.
3. The course provide carrier opportunities in design of some applications as object oriented concepts plays dominant role in software development

Course Outcomes: After completing this course, student's will be able to

CO1: Understand the basic principles of object oriented programming

CO2:Apply the concepts of overloading, inheritance, polymorphism

CO3: Appraise memory allocation techniques and usage of exception handling, generic programming

CO4: Develop programs using object oriented concepts

Course Outcomes	Program Outcomes and Program Specific Outcomes														
	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO1 10**	**PO1 11**	**PO1 12**	**PSO 1**	**PSO 2**	**PSO 3**
CO1	**3**	**2**	**3**	**2**				**1**	**2**	**2**		**3**	**3**	**2**	**1**
CO2	**3**	**3**	**3**	**2**				**1**	**2**	**2**		**3**	**3**	**2**	**2**
CO3	**3**	**3**	**3**	**3**				**1**	**2**	**2**		**3**	**3**	**2**	**2**
CO4	**3**	**3**	**3**	**3**				**1**	**2**	**2**		**3**	**3**	**2**	**2**

Course Contents	**Hrs.**
Unit I: Principles Of Object Oriented Programming	**8**
Differences between C and C++. A look at procedure Oriented programming, object oriented programming paradigm, basic concepts of OOP, Benefits of OOP, OO languages, A sample program, structure of C++ program. Introduction to OOPS :The origins of C++, What is Object Oriented Programming?, Some C++ fundamentals, Headers & Name Spaces, Introducing C++ Classes, Function overloading, Operator overloading, Inheritance, Constructors & Destructors, Function & Operator Overloading	
Unit II: Overloading	**8**
Constructor functions, Localizing variables, Function overloading & Ambiguity, Finding the address of an overloaded function, this Pointer, Operator overloading, References, Using reference to overload a unary operator, Overloading [], overloading ( ), Applying operator overloading.	
Unit III: Inheritance, Virtual Functions and Polymorphism	**8**
Inheritance and the access specifies, Constructors and Destructors in derived classes, Multiple	

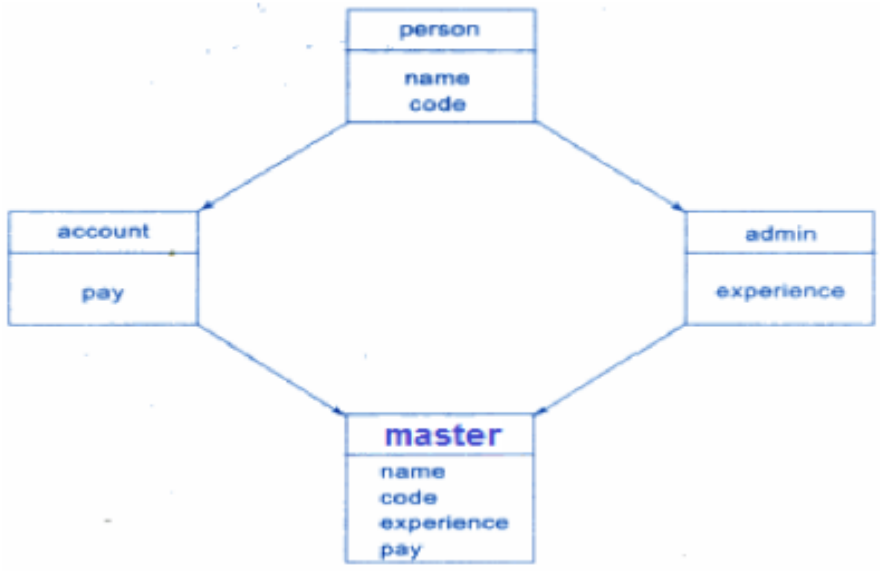
Inheritance, Passing parameters to a basic class, Pointers and references to derived types, Virtual Functions, Why virtual functions?, Pure virtual functions and abstract types, Early Vs Late binding.	
Unit IV: Static & Dynamic allocation	**8**
Static & Dynamic allocation using new and delete, static class members, Virtual base classes, const member functions and mutable, volatile member functions, Using the asm keyword, linkage specification, The .* and ->* operators, Creating conversion functions, Copy constructors, Granting access, namespaces, Explicit constructors, typename and export.	
UNIT V:Templates & Exception Handling	8
Class templates, class templates with multiple parameters, function templates, function templates with multiple parameters, Exception Handling, fundamentals, options the uncaught exception ( ), Applying exception Handling, and RTTI, casting operators, Recent trends in Object Oriented Programming in C++ , Advanced topics & its Application	

Text Books	1.	Object Oriented Programming in C++ -Robert Lafore, edition, Galgotia publications
	2.	The Complete Reference C++, Herbert Schildt, 4th Edition, TMH
Reference Books	1.	Let's C++ by Y. Kanetkar, BPB publications
	2	Object oriented programming with C++, E Balagurusamy, 4th edition, TMH

UITP201:Object Oriented Programming		
Teaching Scheme:	**Credit:**	**Examination Scheme:**
Lectures: 02Hrs./Week	**1**	**INT:25 marks EXT:25 marks**
Prerequisite (If any):		
Course Outcomes: After completing this course, student will be able to		
CO1: Implement Object oriented principles.		
CO2: Demonstrate execution of memory allocation techniques and exception handling techniques		
CO3: Design and develop a solution for real life problems using object oriented concepts.		

Course Outcomes	Program Outcomes and Program Specific Outcomes														
	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO11 0**	**PO11 1**	**PO11 2**	**PSO 1**	**PSO 2**	**PSO 3**
CO1	**3**	**3**	**3**	**2**	**2**			**2**	**3**	**3**		**3**	**3**	**2**	**1**
CO2	**3**	**3**	**3**	**2**	**2**			**2**	**3**	**3**		**3**	**3**	**2**	**2**
CO3	**3**	**3**	**3**	**2**	**3**			**2**	**3**	**3**		**3**	**3**	**2**	**2**

Sr. No.	Name of Experiments / Mini Projects/ Case Studies
1	Write a program to compute the area of triangle and circle by overloading the area () function.
2	Define a class to represent a bank account. Include the following members: Data members: - Name of depositor, Account number, Type of account, Balance amount in the account Member functions: - To assign initial values, To deposit an amount, To withdraw an amount after checking the balance, To display name & balance Write a main program to test program using class and object.
3	Create two classes DM and DB which stores values of distances. DM stores distances in meters and centimeters and DB in feet and inches. Write a program that can read values for the class objects and add one object of DM with another object of DB. Use a friend function to carry out addition operation
4	Create a class MAT of size m * n. Define all possible matrix operations for MAT type objects
5	Create Stud class to display student information using constructor and destructor. (Default constructor, Multiple constructors, Copy constructor, Overloaded constructor)
6	Consider class network of given figure. The class master derives information from both account and admin classes which in turn derive information from the class person. Define all the four classes and write a program to create, update and display the information contained in master objects.

	
7	A book shop sells both books and video tapes. Create a class media that stores the title and price of the publication. Create two derived classes, one for storing number of pages in the book and another for storing playing time of tape. A function display () must be defined in all classes to display class contents. Write a program using polymorphism and virtual function.
8	Write a program to show use of this pointer, new and delete.
9	Write a function template for finding the minimum value contained in an array
10	Write a program containing a possible exception. Use a try block to throw it and catch block to handle it properly.
Open Ended Experiments / New Experiments	
1	Write a class template to represent a generic vector. Include member functions to perform following tasks -To create a vector -To modify the value of given element -To multiply by scalar value. -To display vector.
2	Write a C++ program to design a simple calculator

UCOP104 :Python for Data Science		
Teaching Scheme:	**Credit**	**Examination Scheme**
Practical: 06 Hrs./Week	**3**	**INT :50 Marks**

Course Outcomes: After completing this course, students will be able to

CO1:Implement the basics of python programming

CO 2:Construct various programs using fundamental Python Packages: NumPy, Pandas and Matplotlib.

CO3:Analyze various Machine Learning techniques on different datasets

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO 3	
CO1	3	3	3	2	2			1		3		3	2	2		
CO2	3	3	3	3	3	2		1	2	3	2	3	2	3	1	
CO3	3	3	3	3	3	2		1	2	3	2	3	3	3	1	

Sr.No	List of Laboratory Assignments(*Any 8)
1	Implement basic concept of python.
2	Implement list , tuple concept in python.
3	Create Python program with if else statement.
4	Implement While and For loop using Python.
5	Implement mathematical operation in python.
6	Implement various operations of strings using Python.
7	Implement dictionary concept in Python.
8	Implement Function concept in Python.
9	Create a list of 10 students with their GATE score. Make list within list by making each student and score as a sublist. Retrieve 2 to 5 students. Calculate the average GATE of 10 students. Find the min, max and median of the score of the students. Create a sublist of students who scored more than 50%.
10	Create a 2D numpy array of 6 soccer team players with their [height(meters), weight(kgs)] by reshaping an 1D numpy array. i) Find out the average height ii) median height iii) standard deviation on height iv) Calcute the bmi and print the bmis>25 v) Create two arrays by checking the bmi. a) Arr1: bmi>25 b) Arr2 : bmi<25 and print both the arrays
11	Write a python function that takes two numpy.ndarray objects, checks if they are the same shape (printing ERROR and aborting if they aren't), then raises the numbers b of the second array to the exponents a in the first array. Do this without using numpy functions like numpy.power in your function. Make sure that it works with differently sized numpy arrays arrays of 0 dimensions, 1 dimension and 2 dimensions. Do the same using the numpy.power function.
12	Write a python program to search for a string in anheterogenous array of 7 items by using a user defined function that accepts two arguments. i) If string found return the position of the string ii)if string not found, add the string to the list. Note: Do not use built in functions/methods.
13	i) Write a python program to add two lists using map and lambda function. ii) Write a python program to find the common elements from two lists using filter and lambda function.
14	Write a python program to create a DataFrame of 6 rows from Dictionary. First create a

	Tourism_visitors dictionary from four lists of "Cities", "Visitors", "signups" and "Weekdays". Convert the dictionary to Dataframe importing Pandas library. And create row_labels as the short forms of the country names. And display the DataFrame.
15	Using Pandas library , create sample Dataframe of 10 records and Perform reshaping , grouping and sorting operation .
16	Download 100 CC Records csv from http://eforexcel.com/wp/downloads-17-sample-csv-files-data-sets-for-testing-credit-card/. And read the CSV into a DataFrame. Display the card type, holder name, issuing bank and credit limit of the holders from 20 to 40(row indexes) using loc and iloc commands. Also display all columns and table information.
17	Using Panads library read excel sample file of 100 records into Dataframe. Also write these 100 records inot particular sheet of excel file.
18	Using Panads library read CSV/excel sample file and write into JSON file formation
19	Plot line and scatter charts for students interests in programming against their year in the Engineering college. And derive a distribution of the same over 10 years using a histogram.
20	Use the flavors of cocoa csv and write a python program to create a Box Plot (Box and Whisker) plot for four years 2013 to 2016, the distribution of the ratings.

Department of Computer Engineering

Detailed Syllabus

S. Y. B. Tech. Semester-IV

Scheme of S Y B. Tech. in Computer Engineering

Course Code	Name of Course	Course Category	Teaching Scheme				Credit	Evaluation Scheme					
								Theory			Practical		
			L	T	P	Total Hours		TAE	CAE	ESE	INT	EXT	Total Marks
SEMESTER-IV													
UBSL206A	Transforms and Numerical Methods	BS7	3	1	-	4	4	10	15	50	-	-	75
UCOL205/ UCOP205	Computer Networks	C10	3	-	2	5	4	10	15	50	25	-	100
UCOL206/ UCOP206	Design and Analysis of Algorithm	C11	3	-	2	5	4	10	15	50	25	25	125
UECL207/ UECP207	Applications of Microprocessors and Microcontrollers	C12	3	-	2	5	4	10	15	50	25	-	100
UITL203/ UITP203	Operating System	C13	3	-	2	5	4	10	15	50	25	-	100
UCOL200X	Open Elective - I	OE1	2	-		2	2	10	15	50	-	-	75
UDSP208	Data Analysis using R	A9-A10	-	-	4	4	2	-	-	-	25	-	25
TOTAL			**17**	**1**	**12**	**30**	**24**	**60**	**90**	**300**	**125**	**25**	**600**

UCOL2001	**Basics ofComputer Network(Open Elective)**
UCOL2002	**Cloud Computing**

Department of Computer Engineering

Detailed Syllabus

S. Y. B. Tech. Semester-IV

UBSL206A:Transform and Numerical methods		
Teaching Scheme:	**Credit:**	**Examination Scheme:**
Lectures: 03 Hrs./Week **Tutorial : 01 Hrs./Week**	**4**	**TAE:** 10 Marks,**CAE:** 15 Marks, **ESE :50** Marks

Prerequisite (If any): Differential and Integral calculus ,Fourier series

Course Objectives: After completing this course, student will be able to

1. To develop skills to use Transforms and its applications in the field of Computer Engineering.
2. Transform techniques such as Laplace transform, Fourier transform, Z-Transform and applications to Image processing.
3. To provide suitable and effective Numerical method for obtaining approximate representative numerical results of the problem
4. To solve complex mathematical problems using only simple mathematical operations. The approach involves formulation of mathematical models of physical situations that can be solved with arithmetic operations

Course Outcomes:

CO1: Identify the various methods in Transforms, Numerical that applies to the problems in Computer engineering

CO2: Solve algebraic and transcendental equations and system of linear equations using numerical techniques

CO3: Apply the concept Laplace transform, Fourier transform and Z-transform and its applications to continues and discrete systems and image processing

CO4: Apply the knowledge of numerical techniques to solve ordinary differential equations

Course Outcomes	Program Outcomes and Program Specific Outcomes														
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO3
CO1	3	3	3	3							1	1	1	1	
CO2	3	3	2	2		2	1	1	2	2	1	1	2	2	
CO3	3	3	3	3		2	2	1	1	2	1	1	2	1	
CO4	3	3	3	3		2	2	1	2	2	1	1	3	2	

Course Contents	Hrs.
Unit I: Laplace Transform	**8**
Laplace transform definition and their properties, transform of derivatives and integrals, evaluation of integrals by Laplace Transform, Laplace transforms of periodic function ,Unit step function, Unit Impulse function , Inverse Laplace Transform.& its properties , convolution theorem, applications of Laplace transforms to solve ordinary differential equations**.**	
Unit II: Fourier Transform	**8**
Complex exponential form of Fourier series, Fourier integral theorem, Fourier Sine & Cosine integrals, Fourier transform, Fourier Sine and Cosine transforms and their inverses, Properties of Fourier Transform, Discrete Fourier Transform. Applications of Transforms to boundary	

value Problems	
Unit III: Z-Transform	**8**
Definition, properties of Z- Transforms, Inverse Z- Transform and relation between Z transform and Laplace Transform. Convolution Theorem, Application of Z-Transform to solve difference equations with constant coefficients.	
Unit IV: Numerical Solution of Equations	**8**
Numerical solutions of algebraic and transcendental equations. Iteration method, Bisection method, Regula-Falsi method, Newton-Raphson's method and their convergences, solution of system of linear equations by Gauss elimination method, gauss Jordan method, gauss Seidel iteration method.	
Unit V:Numerical Solution of Ordinary Differential Equations	8
Picard's method, Taylor series method, Euler's method, Modified Euler' s method, Range's method, Runge-Kutta fourth order method, Predicator –corrector methods, Milne's method. Solution of Simultaneous first order and higher order differential equations.	

Text Books	1.	B. S. Grewal, "Higher Engineering Mathematics" Khanna Publication, 43th edition
	2.	H. K. Dass, 'Engineering Mathematics', S. Chand Publication 20e, New Delhi.
	3.	"Introductory Methods of Numerical Analysis '' , S.S.Sastry ,4th edition
E-- Books	1.	http://www.math.ust.hk/~machas/numerical-methods.pdf
Referenc Books	1.	Erwin Kreyszig, "Advanced Engineering Mathematics", 9e, Wiley India
	2.	Robert A.Gabel , Richard A.Roberts ; Siglnals and linear systems ; John Wiley & Sons
	3.	Jain, R.K. and Iyengar,S.R.K, Advanced Engineering Mathematics, 3rd Edition, New Delhi, Narosa Publishers, 2007
Online TL Material	1.	https://onlinecourses.nptel.ac.in/noc19_ge30/preview
	2.	https://nptel.ac.in/courses/111/105/111105123/
	3.	https://nptel.ac.in/courses/111/102/111102129/

UCOL205: Computer Networks		
Teaching Scheme:	**Credit:**	**Examination Scheme:**
Lectures: 03 Hrs./Week	**3**	**TAE:** 10 Marks, **CAE:** 15 Marks, **ESE :50** Marks

Prerequisite (If any):

Course Objectives: After completing this course, student will be able to

1. Build an understanding of the fundamental concepts of computer networking
2. Know about routing mechanisms and different routing protocols
3. Understand transport layer functions
4. Know about different application layer protocols

Couse Outcomes:

CO1. Summarize services offered by layers of OSI model and TCP/IP model

CO2. Determine the different network management techniques of various protocol.

CO3. Explaining wireless network and different wireless standards.

CO4:: Explore various recent trends in networking.

Course Outcomes	Program Outcomes and Program Specific Outcomes														
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO3
CO1	2	1	2	2	2			2		1			3	2	1
CO2	3	2	2	2	3	2	1	1	2	1			2	2	
CO3	1	2	2	3	3			2		2		2	2	2	
CO4	2	2	2	2	2			2		2		2	3	2	

Course Contents	Hrs.
Unit I: Introduction to Computer Networks and Logical Link Medium Access Control	**8**
Introduction – Network architecture -Design. Reference models- The OSI Reference Model- The TCP/IP Reference Model - A Comparison of the OSI and TCP/IP Reference Models Design Issues, Switching Techniques: Circuit and Packet Switching, Connectionless and Connection-oriented Services, Virtual Circuit and Datagram Subnets .Autonomous system	
Unit II: Network Layer-I	**8**
Routing Algorithms: Optimality principle, shortest path routing, flooding, Distance Vector routing, link state routing, hierarchical routing. Network layer services, IP protocol, IPv4, Problems with IPv4,IPV6,Subnetting, Network layer Protocols: ARP, RARP, ICMP, DHCP, Unicast Routing Algorithms: RIP, OSPF, BGP	
Unit III: Transport Layer	**8**
UDP : UDP functionality, UDP Header; TCP : TCP Features, byte-stream, Connection-oriented, TCP Header Format, 2-way, 3-way	

Handshake, TCP State Diagram, TCP Sliding Window, Congestion Control Algorithms, Leaky Bucket, Token Bucket, Congestion Avoidance, TCP Tahoe, Fast Retransmit, Fast Recovery, Timer Management.	
Unit IV: Application Layer	**6**
Domain Name System (DNS), Naming and Address Schemes, DNS servers, Email: MIME, SMTP and POP3. Remote login, File Transfer Protocol (FTP), SNMP, DHCP and BOOTP. World Wide Web, HTTP	
Unit V: WIRELESS LAN,MAN,WAN	6
Introduction (Infrastructure and Ad-hoc Networks),Fundamentals of WLAN – technical issues, Network Architecture, IEEE 802.11- physical layer, Mac Layer Mechanism, CSMA/CA, Bluetooth - Specification, Transport Layer, Middleware Protocol Group, Bluetooth Profiles of IEEE 802.16, Sensor Node Architecture (hardware components),Sensor Network Architectures	

Text Books	1.	Computer Networks (3rd edition), Tanenbaum Andrew S., International edition, 1996.
	2.	Behrouz A. Forouzan, TCP/IP Protocol Suite, McGraw Hill Education, ISBN: 978-0-07-070652-1,4th Edition
	3.	KazemSohraby, Daniel Minoli, TaiebZnati, Wireless Sensor Network", Wiley, ISBN :978-0-471-74300-2.
Referenc e Books	1.	James F. Kurose and Keith W. Ross, "Computer Networking: A Top-Down Approach Featuring the Internet", Pearson Education, 6th Edition, ISBN : 978-02737-68968
	2.	Holger Karl and Andreas Willig, "Protocols and Architectures for Wireless Sensor Networks",

UCOP205: Computer Networks		
Teaching Scheme:	**Credit**	**Examination Scheme**
Practical: 02 Hrs./Week	**1**	**INT :25 Marks**

Course Outcomes : On completion of the course, student will be able to:

CO1: State the various networking Commands and understand Packet Tracer Simulator.

CO2: Applying the simulator concept, implement various protocol

CO3: Implement Socket programming

Course Outcomes	Program Outcomes and Program Specific Outcomes														
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO3
CO1	2	2	2	2	2	3		2		3	3		3	3	2
CO2		3	3	3	3	3	2	2	2	3	3		3	3	1
CO3		2	2	2	2	3	2	2	2	3	3	2	3	3	

Sr. No	List of Laboratory Assignments(*Any 8)
1	Explore and Study of TCP/IP utilities and Network Commands a) Ping b) Tracer c) ipconfig / ifconfig d) Netstat e) Arp f) Whois
2	Using a Network Simulator (e.g. packet tracer) Configure Sub-netting of a given network
3	Using a Network Simulator (e.g. packet tracer) configure **1.** Static Routing 2. RIPv2 routing protocol
4	Using a Network Simulator (e.g. packet tracer) configure **1.** EIGRP 2.OSPF
5	Using a Network Simulator (e.g. packet tracer) configure RIPv2 and EIGRP on same network.
6	Using a Network Simulator (e.g. packet tracer) configure a. VLAN, Dynamic trunk protocol and spanning tree protocol
7	TCP UDP Socket Programming for Client Server Application
8	Using a Network Simulator (e.g. packet tracer) configure WLAN with static IP addressing and DHCP with MAC security and filters
9	Using Network Simulator 2/ OMNET simulate(Any one) a. Local Area Network b. WSN
	Content beyond syllabus
10	Case study of network simulator

UCOL206: Design And Analysis of Algorithms		
Teaching Scheme:	**Credit:**	**Examination Scheme:**
Lectures: 03 Hrs./Week	**3**	**TAE:** 10 Marks,**CAE:** 15 Marks, **ESE :50** Marks

Prerequisite (If any):

Course Objectives: After completing this course, student will be able to

1. This course introduces students the general idea of analysis and design of algorithms while making them aware of basic methods of algorithm analysis and design.
2. It is also aimed at developing skills to solve real life applications which involve algorithm development
3. The course also provides career opportunities in analysis, design and optimization technique in algorithms

Course Outcomes:

CO1: Recall basic concepts of algorithm in analysis and Design of algorithms.

CO2: Examine Recurrence relations, solutions of recurrence of searching sorting methods

CO3: Analyze Greedy methods used for analysis and Design of Algorithm

CO4: Apply Dynamic Programming concepts in designing algorithm

CO5: Evaluate advanced techniques and tools available for algorithm analysis and development

Course Outcomes	Program Outcomes and Program Specific Outcomes														
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3
CO1	3	2	2	2	2	2						1	3	2	
CO2	3	2	2		1	2		2	2	1	2	1	2	1	
CO3	2	2	1			2		2	2	2	2		1	1	
CO4	2	2	1			2		2	2	1	2		1	1	
CO5	2		1	2	2	2			2	1	1	2	2	1	

Course Contents	Hrs.
Unit I: Mathematical foundations & Asymptotic notations	**8**
Algorithm, Mathematical Notations, Algorithm specification, Analysis of Algorithm-Introduction, Analyzing control structures, Asymptotic notations, space complexity, time complexity, Performance measurement, analyzing control structures, best case, worst case and average case analysis, Iterative Algorithm analysis.	
Unit II: Divide and Conquer	**8**
Recurrence relations, solutions of recurrence relations by Master Methods. Divide and conquer basic strategy, binary search, quick sort, merge sort, maximum and minimum finding	
Unit III: Greedy Method	**8**

Greedy method – basic strategy, application to job sequencing with deadlines problem, minimum cost spanning trees, single source shortest path etc.	
Unit IV: Dynamic Programming	**6**
Dynamic Programming basic strategy, multistage graphs, all pairs shortest path, single source shortest paths, optimal binary search trees, traveling salesman problems.	
Unit V:Traversal And Search Techniques	6
Basic Traversal and Search Techniques, breadth first search and depth first search, Backtracking basic strategy, 8-Queen's problem, graph colouring, Hamiltonian cycles.NP, P Problems, Optimization Algorithms.	

Text Books	1.	Thomas H. Cormenet. al. "Introduction to Algorithms", Prentice Hall of India.
	2.	Design & Analysis of Computer Algorithms by Aho,. Horowitz, Sahani, Rajsekharam, Pearson education
Reference Books	1.	"Computer Algorithms", Galgotia Publications Pvt. Ltd. Brassard, Bratley, "Fundamentals of Algorithms", Prentice Hall
	2.	Computer Algorithms: Introduction to Design and analysis, 3rd Edition, By Sara Baase& A. V. Gelder Pearson Education.
On-line TL Material	1.	NPTEL course on Design and Analysis of Algorithms: https://www.class-central.com/course/nptel-design-and-analysis-of-algorithms-3984

UCOP206: Design And Analysis of Algorithms		
Teaching Scheme:	**Credit**	**Examination Scheme**
Practical: 02 Hrs./Week	**1**	**INT :25 Marks Ext: 25 Marks**

Course Outcomes : On completion of the course, student will be able to–

CO1: Describe concepts of specific algorithmic characteristics

CO2: Interpret various problem solving techniques using algorithmic types

C**O3**:Explain performance of algorithms using mathematical formulas

CO4: Demonstrate design strategies for solving various applications

CO5:Analyze complexity of problems for advanced computing areas

Sr. No	List of Laboratory Assignments(*Any 8)
1	Write C++ program to find factorial of a given number using (i) Recursion (ii) Iteration Compare time and space complexity of both the designs.
2	Implement Binary search program with Divide and Conquer design strategy for n numbers using C++. Discuss Best, Average and Worst time complexity.
3	Sort a given set of n integer elements using Quick Sort method and compute its time complexity. Run the program for varied values of n and record the time taken to sort. The elements can be read from a user or can be generated using the random number generator. Demonstrate using C++ how the divide and conquer method works along with its time complexity analysis: worst case, average case and best case.
4	A business house has several offices in different countries; they want to lease phone lines to connect them with each other and the phone company charges different rent to connect different pairs of cities. Business house want to connect all its offices with a minimum total cost. Solve the problem by suggesting appropriate data structures in C++.
5	From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm. Write the program in C++.
6	Implement a program in C++ for 0/1 Knapsack problem using Dynamic Programming method.
7	Write C++ program to implement Travelling Sales Person problem using Dynamic programming.
8	Implement C++ program for solving N-Queen's problem using Back tracking. (Assume N=4)
9	Implement Travelling salesman problem using branch and bound approach using C++.
10	Write C++ Program to demonstrate the implementation of Rabin-Karp Algorithm with discussion of time complexity.
	Content beyond syllabus
11	Matrix Chain Multiplication using Dynamic Programming
12	Case Study of Optimization Algorithms for complexity problems

UECL207:Applications of Microprocessors and Microcontrollers		
Teaching Scheme:	**Credit:**	**Examination Scheme:**
Lectures: 03 Hrs./Week	**3**	**TAE:** 10 Marks,**CAE:** 15 Marks, **ESE :50** Marks

Prerequisite (If any):

Course Objectives: After completing this course, student's will be able to

1. To studyandunderstand various microcontrollersand embedded systems
2. Tounderstandthedesignparametersofembeddedsystemsapplications.
3. Tostudyandimpartdifferenttoolsforembeddedsystemand IoTapplication design.

Course Outcomes:

CO1: DemonstratetheprincipleofembeddedsystemsandMicrocontroller

CO2: Designtheinterfacingofdevicesandperipherals

CO3: Developprogrammingforapplicationsdevelopealtimeapplications

CO4: MakeUse of Arduino Controller for Designing of Embedded Applications.

CO5: Design andDevelopdifferent embeddedsystemand IoTapplications.

CourseOutcomes	ProgramOutcomesandProgramSpecificOutcomes														
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PS01	PS02	PS03
CO1	2	2	3	2	3						2	1	3	2	3
CO2	2	2	3	2	3	2	2	2	2		2	1	3	2	3
CO3	3	2	3	2	3	2	2	2	2	2	2	1	3	2	3
CO4	3	2	3	2	3	2	2	2	2	2	2	1	3	2	3
CO5	3	2	3	3	3	2	2	2	2		2	1	3	2	3

Course Contents	Hrs.
Unit I: Introduction	**8**
MicroprocessorTechnology:8085/8086-architecturaloverview&Programmingmodel.	
Unit II: Microcontrollers	**8**
Introductiontomicrocontrollers,8051architecture,data typesanddirectives, flagbitsandPSWregister,registerbankandstack.	

Unit III: AssemblyLanguageProgramming	**8**
Jump,LoopandCallInstructions,I/O PortProgramming,Addressingmodes,Arithmetic,Logicinstructionsandprograms,datatypesandtime delay.InterfacingtoExternalMemory.	
Unit IV: Programming	**6**
Programming: Timer/counter,Interruptsandserialcommunications,SerialI/O,ProgrammingTools,Program usingC **Interfacingwith8051:**ADCandDACinterfacesformicrocontrollers,Real timeinterfacingwith LED,Keypad, LCDdisplay, Sensorsinterfacing	
Unit V:Arduino	6
IntroductiontoArduino,Pinconfigurationandarchitecture,codingofArduniousingIDE.Interfacings	

TextBooks	1.	MuhammadAliMazidi,the8051Micro-controller&EmbeddedSystemusing assembly&C, PearsonEducation,2008,Second
	2	Muhammad Ali Mazidi, ARM Assembly language programming And Architecture,Second
	3	RajKamal,Microcontrollers:Architecture,Programming,Interfacingand SystemDesign,PearsonEducation India,2009,Second
Reference Books	1.	ShibuK.V.Introduction toEmbeddedSystem,TheMcGrawHill,2011
	2.	AjayV.Deshmukh,Micro-controllers-TheoryandApplications,TataMcGraw Hill,
	3.	KennethJ.Ayala,The8051Micro-controller–Architecture,Programming& Applications,PenramInternational &ThomsonAsia,1996,Second
on lineTL Material	1.	https://nptel.ac.in/courses/108/105/108105102/

UECP207:Applications of Microprocessors and Microcontrollers Lab		
Teaching Scheme:	**Credit**	**Examination Scheme**
Practical: 02 Hrs./Week	**1**	**INT :25 Marks Ext: NA**
Course Outcomes : On completion of the course, student will be able to–		
CO1: Demonstratetheprincipleofembeddedsystemsand Microcontroller		
CO2: Designtheinterfacingofdevicesandperipherals		
CO3: Developprogrammingforapplicationsdeveloprealtimeapplications		
CO4: MakeUse of Arduino Controller for Designing of Embedded Applications.		
CO5: Design andDevelopdifferent embeddedsystemand IoTapplications.		

Sr.No	List of Laboratory Assignments(*Any 8)
1	WriteaprogramtoperformArithmaticoperationsusing8051 microcontroller
2	Writa programto performdatatransfer between twomemoryblocksusing8051microcontroller
3	Writaprogramtofindsmallestnumberfrommemoryblocksusing8051microcontroller
4	InterfaceLED/7Segmentdisplaywith8051Microcontroller(Proteous based)
5	Interface different sensor like LDR, IR with 8051 Microcontroller (Proteousbased)
6	Interface LCDdisplaywithArduino(Handson)
7	Design andperformdifferentembeddedsystem and IoTApplications
8	ProjectModule-1
9	ProjectModule-2
10	ProjectModule-3
	Content beyond syllabus
11	Study any simulator tool for microprocessor and micro controller

UITL206:OPERATING SYSTEM		
Teaching Scheme:	**Credit:**	**Examination Scheme:**
Lectures: 03 Hrs./Week	**3**	**TAE:** 10 Marks, **CAE:** 15 Marks, **ESE :50** Marks

Prerequisite (If any):

Course Objectives: After completing this course, student will able to

1.Introduces general idea, structure and functions of operating system

2.Making students aware of basic mechanisms used to handle processes, memory, storage devices and files.

3.Recent trends in the operating system

Couse Outcomes:

CO1:Identify basic structure and purpose of operating system.

CO2. Interpret the concepts of process and illustrate various CPU scheduling algorithms.

CO3. Interpret the concepts of inter process communication.

CO4. Schematize Deadlock & security mechanisms in operating systems.

CO5. Analyze different memory management techniques with advantages and disadvantages.

Course Outcomes	Program Outcomes and Program Specific Outcomes														
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO3
CO1	2	2	3	2	2								1	1	1
CO2	2	2	3	3	3								3	3	
CO3	3	3	3	3	3								3	3	
CO4	3	3	2	3	3								3	3	1

Course Contents	Hrs.
UNIT I: Introducation	**8**
Evolution of OS, Types of OS, Basic h/w support necessary for modern operating systems, services provided by OS, system programs and system calls, system design and implementation.	
UNIT II: Process & Its Scheduling	**8**
Process & Its Scheduling Process concept, process control block, Types of scheduler, context switch, threads, multithreading model, goals of scheduling and different scheduling algorithms	
UNIT III:Process management and synchronization	**8**
Process management and synchronization: Concurrency conditions, Critical section problem, software and hardware solution, semaphores, conditional critical regions and monitors, classical inter process communication problems	

UNIT IV:Deadlock	**7**
Deadlock definitions, Prevention, Avoidance, detection and Recovery, Goals of Protection, access matrix, Deadlock implementation	
UNIT V: File systems	8
File systems: File concept, Access methods space allocation strategies, disk arm scheduling strategies. Contiguous allocation, Relocation, Paging, Segmentation, Segmentation with paging, demand paging, Virtual Memory Concepts, page faults and instruction restart , page replacement algorithms, working sets, Locality of reference, Thrashing, Garbage Collection.	

Text Books	1.	Operating System concepts – Silberchatz; Galvin, Addison Wesley, 6 thEdn.
	2.	Modern Operating Systems – Tanenbaum, Pearson Edn. 2 ndedn
	3.	Operating Systems: Internals and Design Principles -- William Stallings
Reference Books	1.	Operating Systems – S R Sathe, Macmillan Publishers, India, 2008
	2.	Operating System –Milan Milenkovik, McGraw-Hill, 1987
	3.	Operating Systems - 3 rd Edition by Gary Nutt, Pearson Education.
on line TL Material	1.	https://nptel.ac.in/courses/106/108/106108101/

UITL206:OPERATING SYSTEM Lab		
Teaching Scheme:	**Credit**	**Examination Scheme**
Practical: 02Hrs./Week	**1**	**INT :25 Marks Ext: NA**
Course Outcomes : On completion of the course, student will be able to–		
CO1:Identify basic structure and purpose of operating system.		
CO2: Interpret the concepts of process and illustrate various CPU scheduling algorithms.		
CO3:Interpret the concepts of inter process communication.		
CO4: Schematize Deadlock & security mechanisms in operating systems.		

Sr.No	List of Laboratory Assignments(*Any 8)
1	Study of Unix/Linux general purpose utility command list obtained from (man, who, cat, cd, cp, ps, ls, mv, rm, mkdir, rmdir, echo, more, date, time, kill, history, chmod, chown, finger, pwd, cal, logout, shutdown) commands
2	Write a program in shell for printing table of any number.
3	Implement program in shell script: a) To find factorial of given number b) To find greatest of three number
4	Implement program in shell script : a) To print given number in reverse order b) To find even and odd numbers from given array
5	Write a menu driven shell script program to develop a calculator.
6	Write a menu driven program by using switch case for following: a) To find factorial of given number b) To find greatest of three number c) To print given number in reverse order d) To find even and odd numbers from given array
7	Write a program for creating child process by fork () command.
8	To implement BANKER'S algorithm for deadlock avoidance.
9	Implement of following Non pre-emptive CPU scheduling algorithms: First Come First Serve, Shortest Job First, Priority
10	Implement of following Pre-emptive CPU scheduling algorithms: Shortest Job First, Priority, and Round Robin.
	Content Beyond Syllabus
11	Implementation of Page replacement algorithms a) First In First Out b) List Recently Used c) Optimal Page replacement algorithm
12	Write a program for creating child process other than fork () command.
13	Write a program through which run any Unix command

UCSL2001:Basics of Computer Network(Open Elective 1)		
Teaching Scheme:	**Credit:**	**Examination Scheme:**
Lectures: 02 Hrs./Week	**2**	**TAE:** 10 Marks, **CAE:** 15 Marks, **ESE :50** Marks

Prerequisite (If any):

Course Objectives: After completing this course, student's will be able to

1. Build an understanding of the fundamental concepts of data communication and computer networking
2. Know about routing mechanisms and different routing protocols
3. Understand transport layer functions
4. Know about different application layer protocols

Couse Outcomes:

CO1. Summarize services offered by layers of OSI model and TCP/IP model

CO2. Determine the different network management techniques of various protocol.

CO3. Understand and building the skills of subnetting and routing mechanisms

CO4: Explore various recent trends in networking.

Course Outcomes	Program Outcomes and Program Specific Outcomes														
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO3
CO1	2	1	2	2	2			2		1			3	2	1
CO2	3	2	2	2	3	2	1	1	2	1			2	2	
CO3	1	2	2	3	3			2				2	2	2	
CO4	2	2	2	2	2			2		2		2	3	2	

Course Contents	Hrs.
Unit I: Introduction to Computer Networks and Logical Link Medium Access Control	**8**
Introduction – Network architecture -Design. Reference models- The OSI Reference Model- The TCP/IP Reference Model - A Comparison of the OSI and TCP/IP Reference Models Design Issues, Switching Techniques: Circuit and Packet Switching, Connectionless and Connection-oriented Services, Virtual Circuit and Datagram Subnets	
Unit II: Network Layer-I	**8**
Routing Algorithms: Optimality principle, shortest path routing, flooding, Distance Vector routing, link state routing, hierarchical routing. Network layer services, IP protocol, IPv4, Problems with IPv4,IPV6, Sub-netting, Network layer Protocols: ARP, RARP, ICMP, DHCP, Unicast Routing Algorithms: RIP, OSPF, BGP	
Unit III: Transport Layer	**8**
UDP : UDP functionality, UDP Header; TCP : TCP Features, byte-stream, Connection-oriented, TCP Header Format, 2-way, 3-way Handshake, TCP State Diagram, TCP Sliding Window, Congestion Control Algorithms,	

Leaky Bucket, Token Bucket, Congestion Avoidance, Fast Retransmit, Fast Recovery,	
Unit IV: Application Layer	**8**
Domain Name System (DNS), Naming and Address Schemes, DNS servers, Email: MIME, SMTP and POP3. Remote login, File Transfer Protocol (FTP), SNMP, DHCP and BOOTP. World Wide Web, HTTP	

Text Books	1.	Computer Networks (3rd edition), Tanenbaum Andrew S., International edition, 1996.
	2.	Behrouz A. Forouzan, TCP/IP Protocol Suite, McGraw Hill Education, ISBN: 978-0-07-070652-1,4th Edition
	3.	KazemSohraby, Daniel Minoli, TaiebZnati, Wireless Sensor Network", Wiley, ISBN :978-0-471-74300-2.
Referenc e Books	1.	James F. Kurose and Keith W. Ross, "Computer Networking: A Top-Down Approach Featuring the Internet", Pearson Education, 6th Edition, ISBN : 978-02737-68968
	2.	Holger Karl and Andreas Willig, "Protocols and Architectures for Wireless Sensor Networks",

UCOL2002: Cloud Computing (Open Elective 1)

Teaching Scheme:	Credit:	Examination Scheme:
Lectures: 02 Hrs./Week	**2**	**TAE:** 10 Marks, **CAE:** 15 Marks, **ESE :50** Marks

Prerequisite (If any):

Course Objectives: After completing this course, student's will be able to

1. To study cloud computing concepts.
2. To learn Key concepts of virtualization.
3. Enhancing cloud computing environment.
4. To study various platforms and Storage structure of cloud

Couse Outcomes:

1. Understand the cloud computing concepts.
2. Describe importance of virtualization along with their technologies.
3. Define Cloud Computing and memorize the different Cloud service and deployment models
4. Understand and apply different Storage structure of cloud.

Course Outcomes	Program Outcomes and Program Specific Outcomes														
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO3
CO1	2	1	2	2	2			2		1			3	2	1
CO2	3	2	2	2	3	2	1	1	2	1			2	2	
CO3	1	2	2	3	3			2				2	2	2	
CO4	2	2	2	2	2			2		2		2	3	2	

Course Contents	Hrs.
Unit I: Introduction	**8**
Limitations, Security Concerns. Software as a Service (SaaS)- Understanding the Multitenant Nature of SaaS Solutions, Understanding SOA. Platform as a Service (PaaS)-IT Evolution Leading to the Cloud, Benefits of Paas Solutions, Disadvantages of PaaS Solutions. Infrastructure as a Service (IaaS)-Understanding IaaS, Improving Performance through Load Balancing, System and Storage Redundancy, Utilizing Cloud-Based NAS Devices, Advantages, Server Types. Identity as a Service (IDaaS).	
Unit II: Virtualization	**8**
Introduction, Characteristics of Virtualized environments, Taxonomy of Virtualization techniques, Pros and Cons of Virtualization, Technology examples: Xen, KVM, Vmware, Microsoft Hyper-V.	
Unit III: Storage in Cloud	**8**
Cloud file systems: GFS and HDFS, BigTable, HBase and Dynamo Cloud data stores: Datastore and Simple DB GautamShrauf, Cloud Storage-Overview, Cloud Storage Providers. [Anthony T. Velte]3 Securing the Cloud- General Security Advantages of Cloud-Based Solutions, Introducing Business	

Continuity and Disaster Recovery. Disaster Recovery- Understanding the Threats.	
Unit IV: Cloud computing platforms	**8**
Infrastructure as Service, best-of breed cloud infrastructure components, cloud ready converged infrastructure, Virtual machine provisioning and migration services, Anatomy of Cloud infrastructure, Distributed management of virtual infrastructure, scheduling techniques, SLA Commitment	

Text Books	1.	RajkumarBuyya, James Broberg, Andrzej M. Goscinski , " Cloud computing principles and paradigms", Wiley Publishing ©2011 ISBN: 9780470887998
	2.	Gautam Shroff," Enterprise Cloud Computing", Cambridge University Press, December 2010,Online ISBN: 9780511778476 Online ISBN: 9780511778476.
	3.	BorkoFurht, Armando Escalante, Handbook of Cloud Computing, Springer Publication, ISBN:1441965238 978 1441965233
Reference Books	1.	Dr. Kumar Saurabh, " Cloud Computing", Wiley Publication@2012 ISBN 10: 8126536039
	2.	Greg Schulz, "Cloud and virtual data storage networking", CRC Press Auerbach Publications ,Published August 26, 2011 ,ISBN 9781439851739.
		1. Barrie Sosinsky,"Cloud Computing", Wiley India @2011 ,ISBN: 9780470903568.

UDSP208:Data Analysis using Python/R		
Teaching Scheme:	**Credit:**	**Examination Scheme:**
Practical: 04 Hrs./Week	**2**	**INT:** 25 marks

Prerequisite (If any):

Course Objectives:

1. To understand the basics of Data Analysis and Statistics.
2. To understand the Expert concepts of Machine learning using R programming.
3. To explore the concept of principal component analysis and Hypothesis testing.
4. To explore tools and practices for working with R

Course Outcomes: After completing this course, students will be able to

CO1. Apply the concepts of data analysis for a domain.

CO2. Understand and apply the data analysis technique for Machine Learning Model Designing

CO3. Understand pull data from different sources (small dataset and large datasets),clean and manipulate data

CO4. Preprocess and clean data for ML model design.

Course Outcomes	Program Outcomes and Program Specific Outcomes														
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO 12	PSO1	PSO 2	PSO3
CO1	1	1	2	2										1	1
CO2		1	2	1										1	
CO3		1	2	2										1	
CO4		1	1	1										1	

Course Contents	Hrs.
Unit I: Introduction to data analysis	**6**
Introduction to data analysis: Overview, Data Science vs Data Analsis, Business Analytics classification, Data Science Project workflow, Project Roles, Introduction to R programming, R Studio, Applications of R	
Unit II: Statistics for Data Analysis	**6**
Statistics basic terminologies, Sampling methods, Cluster Sampling, Systematic & Biased Sampling. Sampling Error, EDA, EDA – **Measures of Central Tendency**: Mean,Median, Mode, Mid-range. **Measures of Dispersion**: Range, Variance, Mean Deviation, StandardDeviation.	
Unit IV:Introduction to Hypothesis	**4**
Bayestheorem,Basicsandneedofhypothesisandhypothesistesting,PearsonCorrelation,Sample Hypothesis testing.	
Unit V: Basic Data Analysis through RStudio	**6**

Basic Data Analysis through RStudio, Essentials of R Programming: Data Types and Objects in R, Control Structures (Functions) in R, Useful R Packages. Exploratory Data Analysis in R : Basic Graphs, Treating Missing values, Working with Continuous and Categorical Variables.	
Unit VI:Data Manipulation in R	**6**
Data Manipulation in R: **Feature Engineering,** Label Encoding and One Hot Encoding. Predictive Modeling using Machine Learning: Linear (Multiple) Regression, Decision Trees, Rando Forest.	

Text Book	1	Hands-on Programming with R, Garrett Grolemund.
	2	R for Everyone: Advanced Analytics and Graphics, Jared Lander
	3	Data Analytics: The Complete Beginner's Guide: the Black book,Byron Francis, Create **Space Ind**pendent Publishing Platform, 2016
E-Books	1	R in Nutshell, Joseph Adler, O'Reilly Publications
	2	Introduction to Statistical learning with R, Gareth James, Daniela Written, Trevor Hastie, Robert Tibshivani, Springer Publications
	3	Data Analytics for Beginner: Paul Kinley
Reference Books	1	Applied predictive modeling by Max Kuhn and Kejell Johnson
	2	Introduction to statistical learning by Trevor Hastie
	3	Data Manipulation with R, Springer Publications
On line TL Material	1	NPTEL Course: https://nptel.ac.in/courses/110/106/110106072/
	2	Coursera Course: https://www.coursera.org/specializations/statistics
	3	Swayam Course: https://swayam.gov.in/nd1_noc20_ma53/preview

Sr.No.	NameofExperiment
1	a. InstallationandConfigurationof R/Python Studio. b. Write an R/Python program to take input from the user (name and age) and display the values. Also print the version of R installation.
2	Write a R/Python program to get the first 10 Fibonacci numbers.
3	Write a R/Python program to extract first 10 english letter in lower case and last 10 letters in upper case and extract letters between 22nd to 24th letters in upper case.
4	Write a R/Python program to create a list of random numbers in normal distribution and count occurrences of each value.
5	Write a R/Python program to create a 5 x 4 matrix , 3 x 3 matrix with labels and fill the matrix by rows and 2 × 2 matrix with labels and fill the matrix by columns.
6	Write an R/Python program to find the maximum and the minimum value of a given

	vector.
7	Write a R/Python program to create a simple bar plot of five subjects marks.
8	Write a R/Python program to create bell curve of a random normal distribution.
9	Write a R/Python program to compute sum, mean and product of a given vector elements.
10	Write a R program to create a list of heterogeneous data, which include character, numeric and logical vectors. Print the lists.
11	Write a R/Python program to read the .csv file and display the content. Write a R/Python program to create a Data Frames which contain details of 5 employees and display summary of the data.
12	Download the Iris flower dataset or any other dataset into a DataFrame. (eg https://archive.ics.uci.edu/ml/datasets/Iris) Use R/Python and Perform following – Create a histogram for each feature in the dataset to illustrate the feature distributions. Plot each histogram. Create boxplot for each feature in the dataset. All of the boxplots should be combined into a single plot. Compare distributions and identify outliers.