Progressive Education Society’s

Modern College of Engineering, Shivajinagar,Pune-05.

Department of Artificial Intelligence and Data Science

Curriculum Booklet

2019 Pattern

Semester: I

Class: SE (Artificial Intelligence and Data Science)

Vision of the Institute

- Creation of a collaborative academic environment to foster professional excellence and ethical values.

Mission of the Institute

- To develop outstanding professionals with high ethical standards capable of creating and managing global enterprises.
- To foster innovation and research by providing a stimulating learning environment.
- To ensure equitable development of students of all ability levels and backgrounds.
- To be responsive to changes in technology, socio-economic and environmental conditions.
- To foster and maintain mutually beneficial partnerships with alumni and industry.

Objectives of the Institute

- To develop infrastructure appropriate for delivering quality education
- To develop the overall personality of students who will be innovators and future leaders capable of prospering in their work environment.
- To inculcate ethical standards and make students aware of their social responsibilities.
- Promote close interaction among industry, faculty and students to enrich the learning process and enhance career opportunities.
- Encourage faculty in continuous professional growth through quality enhancement programs and research and development activities.
- Foster a healthy work environment which allows for freedom of expression and protection of the rights of all stakeholders through open channels of communication

Vision of the Department

To create a collaborative academic environment in the field of Artificial Intelligence and Data Science by imparting required skill sets and interactive industry interface for students and inculcate into them social and ethical values

Mission of the Department

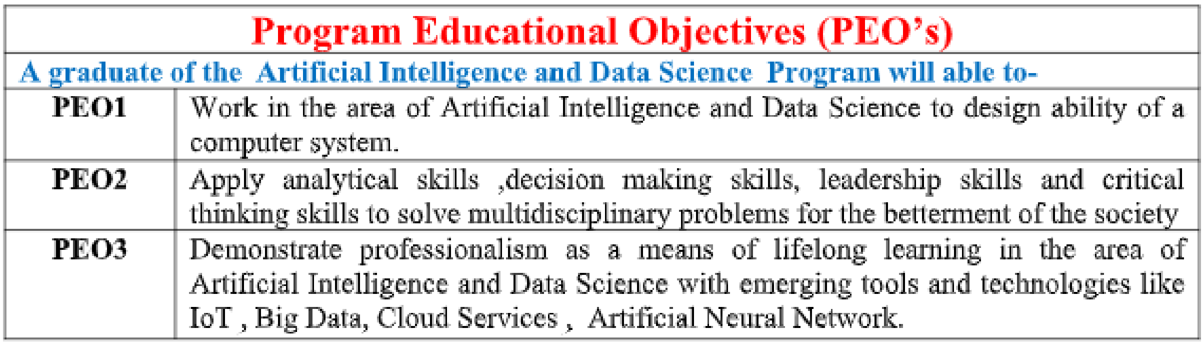
M1: To nurture students with latest technologies in the field of Artificial Intelligence and Data Science.

M2: To build industry academia interface to update the recent trends in field of Artificial Intelligence and Data Science.

M3: To prepare students with professional approach, strong ethical values and research spirit along with leadership skills.

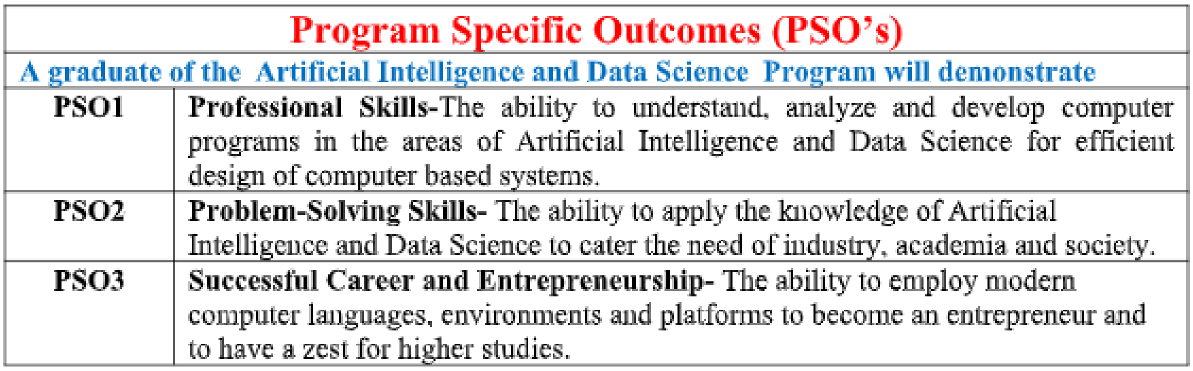
M4: To mentor students in the field of Artificial Intelligence and Data Science research to serve the needs of the society.

Program Educational Objectives (PEOs)



Program Educational Objectives (PEO's)	
A graduate of the Artificial Intelligence and Data Science Program will able to-	
PEO1	Work in the area of Artificial Intelligence and Data Science to design ability of a computer system.
PEO2	Apply analytical skills ,decision making skills, leadership skills and critical thinking skills to solve multidisciplinary problems for the betterment of the society
PEO3	Demonstrate professionalism as a means of lifelong learning in the area of Artificial Intelligence and Data Science with emerging tools and technologies like IoT , Big Data, Cloud Services , Artificial Neural Network.

Program Specific Outcomes (PSOs)



Program Specific Outcomes (PSO's)	
A graduate of the Artificial Intelligence and Data Science Program will demonstrate	
PSO1	**Professional Skills-**The ability to understand, analyze and develop computer programs in the areas of Artificial Intelligence and Data Science for efficient design of computer based systems.
PSO2	**Problem-Solving Skills-** The ability to apply the knowledge of Artificial Intelligence and Data Science to cater the need of industry, academia and society.
PSO3	**Successful Career and Entrepreneurship-** The ability to employ modern computer languages, environments and platforms to become an entrepreneur and to have a zest for higher studies.

Program Outcomes (POs)

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

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

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

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

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

6. The Engineer and Society: Apply reasoning informed by contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant and cultural issues and the consequent responsibilities relevant to professional engineering practice.

7. Environment and Sustainability: Understand the impact of professional engineering solutions in societal and professional engineering solutions in societal and environmental contexts and demonstrate knowledge of and need for sustainable development.

8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of engineering practice.

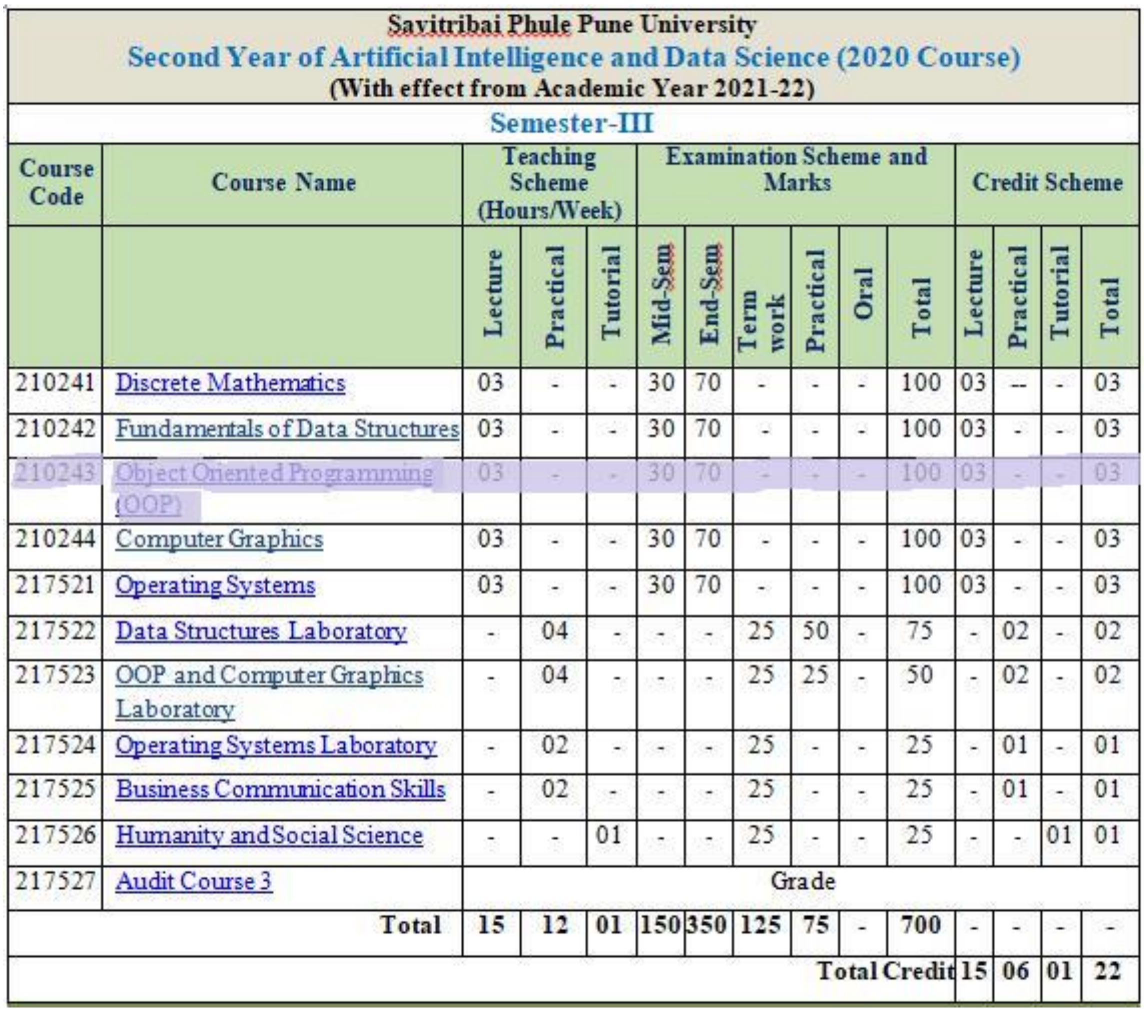
9. Individual and Team Work: Function effectively as an individual, and as a member or leader in diverse teams and in multidisciplinary settings.

10. Communication Skills : 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 clear instructions.

11. Project Management and Finance: Demonstrate knowledge and understanding of 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 multidisciplinary environments.

12. Life-long Learning: Recognize the need for and have the preparation and ability to engage in independent and preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

Course Structure



Savitribai Phule Pune University

Second Year of Artificial Intelligence and Data Science (2020 Course)

(With effect from Academic Year 2021-22)

Semester-III

Course Code	Course Name	Teaching Scheme (Hours/Week)			Examination Scheme and Marks						Credit Scheme			
		Lecture	Practical	Tutorial	Mid-Sem	End-Sem	Term work	Practical	Oral	Total	Lecture	Practical	Tutorial	Total
210241	Discrete Mathematics	03	-	-	30	70	-	-	-	100	03	--	-	03
210242	Fundamentals of Data Structures	03	-	-	30	70	-	-	-	100	03	-	-	03
210243	Object Oriented Programming (OOP)	03	-	-	30	70	-	-	-	100	03	-	-	03
210244	Computer Graphics	03	-	-	30	70	-	-	-	100	03	-	-	03
217521	Operating Systems	03	-	-	30	70	-	-	-	100	03	-	-	03
217522	Data Structures Laboratory	-	04	-	-	-	25	50	-	75	-	02	-	02
217523	OOP and Computer Graphics Laboratory	-	04	-	-	-	25	25	-	50	-	02	-	02
217524	Operating Systems Laboratory	-	02	-	-	-	25	-	-	25	-	01	-	01
217525	Business Communication Skills	-	02	-	-	-	25	-	-	25	-	01	-	01
217526	Humanity and Social Science	-	-	01	-	-	25	-	-	25	-	-	01	01
217527	Audit Course 3	Grade												
	Total	**15**	**12**	**01**	**150**	**350**	**125**	**75**	-	**700**	-	-	-	-
										Total Credit	**15**	**06**	**01**	**22**

Subject 1: Discrete Mathematics

Weekly Work Loads(in Hrs)	Lecture	Tutorial	Practical
	03	-	-

Online/ In-sem	Theory	Practical	Oral	Term-work	Total Marks	Credit
30	70	0	0	0	100	03

Course Objectives

- To use appropriate set, function and relation models to understand practical examples, and interpret the associated operations and terminologies in context.
- Determine number of logical possibilities of events.
- Learn logic and proof techniques to expand mathematical maturity.
- Formulate problems precisely, solve the problems, apply formal proof techniques, and explain the reasoning clearly.

Course Outcomes:

On completion of the course, learner will be able to :

C241.1: Design and analyze real world engineering problems by applying set theory, propositional logic and mathematical induction

C241.2: Develop skill in expressing mathematical properties of relation and function

C241.3: Identify number of logical possibilities of events to design professional engineering Solutions

C241.4: Model and solve computing problem using tree and graph Analyze the properties of binary operations and evaluate the algebraic structure

C241.5: Apply abstract algebra in combinatorics, coding theory and questions regarding geometric constructions

SYLLABUS

UNIT – I : Set Theory and Logic

Topics :

Introduction and significance of Discrete Mathematics, **Sets–** Naïve Set Theory (Cantorian Set Theory), Axiomatic Set Theory, Set Operations, Cardinality of set, Principle of inclusion and exclusion. **Types of Sets** – Bounded and Unbounded Sets, Diagonalization Argument, Countable and Uncountable Sets, Finite and Infinite Sets, Countably Infinite and Uncountably Infinite Sets, Power set, **Propositional Logic-** logic**,** Propositional Equivalences, Application of Propositional Logic-Translating English Sentences, Proof by Mathematical Induction and Strong Mathematical Induction.

At the end of this unit students will be able to -		**No. of Lectures** – 06
Sr. No.	**Learning Outcomes**	**Bloom's taxonomy Level**
1	Design and analyze real world engineering problems by applying set theory, propositional logic and mathematical induction	

UNIT – II : Relations and Functions

Topics :

Relations and their Properties, n-ary relations and their applications, Representing relations , Closures of relations, Equivalence relations, Partial orderings, Partitions, Hasse diagram, Lattices, Chains and Anti-Chains, Transitive closure and Warshall's algorithm. **Functions-** Surjective, Injective and Bijective functions, Identity function, Partial function, Invertible function, Constant function, Inverse functions and Compositions of functions, The Pigeonhole Principle.

At the end of this unit students will be able to -		**No. of Lectures** – 06
Sr. No.	**Learning Outcomes**	**Bloom's taxonomy Level**
1	Develop skill in expressing mathematical properties of relation and function	

UNIT – III : Counting Principles

Topics :

The Basics of Counting, rule of Sum and Product, Permutations and Combinations, Binomial Coefficients and Identities, Generalized Permutations and Combinations, Algorithms for generating Permutations and Combinations.

At the end of this unit students will be able to -		**No. of Lectures** – 06
Sr. No.	**Learning Outcomes**	**Bloom's taxonomy Level**
1	Identify number of logical possibilities of events to design professional engineering Solutions	

UNIT – IV : Graph Theory

Topics :

Graph Terminology and Special Types of Graphs, Representing Graphs and Graph Isomorphism, Connectivity, Euler and Hamilton Paths, the handshaking lemma, Single source shortest path- Dijkstra's Algorithm, Planar Graphs, Graph Colouring.

At the end of this unit students will be able to -		**No. of Lectures** – 06
Sr. No.	**Learning Outcomes**	**Bloom's taxonomy Level**
1	Model and solve computing problem using graph Analyze the properties of binary operations and evaluate the algebraic structure	

UNIT – V : Trees

Topics : Introduction, properties of trees, Binary search tree, tree traversal, decision tree, prefix codes and Huffman coding, cut sets, Spanning Trees and Minimum Spanning Tree, Kruskal's and Prim's algorithms, The Max flow- Min Cut Theorem (Transport network).		
At the end of this unit students will be able to -		**No. of Lectures** – 06
Sr. No.	**Learning Outcomes**	**Bloom's taxonomy Level**
1	Model and solve computing problem using tree, Analyze the properties of binary operations and evaluate the algebraic structure	

UNIT – VI : Algebraic Structures and Coding Theory **Topics :** The structure of algebra, Algebraic Systems, Semi Groups, Monoids, Groups, Homomorphism and Normal Subgroups, and Congruence relations, Rings, Integral Domains and Fields, Coding theory, Polynomial Rings and polynomial Codes, Galois Theory –Field Theory and Group Theory.		
At the end of this unit students will be able to -		**No. of Lectures** – 06
Sr. No.	**Learning Outcomes**	**Bloom's taxonomy Level**
1	Apply abstract algebra in combinatorics, coding theory and questions regarding geometric constructions.	

Teaching Plan

Sr. No.	Unit	Topics to be covered	Book Referred	Total Lecture Planned
1	I	Set Theory and Logic	T1,T2,R1	6
2	II	Relations and Functions	T1,T2,R1	6
3	III	Counting Principles	T1,T2,R4	6
4	IV	Graph Theory	T1,T2,R2	6
5	V	Trees	T1,T2,R1	6
6	VI	Algebraic Structures and Coding Theory	T1,T2,R3	6

Text Books:

1. T1: C. L. Liu, —Elements of Discrete Mathematics, TMH, ISBN 10:0-07-066913-9.
2. T2: N. Biggs, --“Discrete Mathematics”, 3rd Ed, Oxford University Press, ISBN 0 –19-850717–8.

Reference Books:

1. R1: Bernard Kolman, Robert C. Busby and Sharon Ross, Discrete Mathematical Structures, Prentice-Hall of India /Pearson, ISBN: 0132078457, 9780132078450.
2. R2: Narsingh Deo, “Graph with application to Engineering and Computer Science”, Prentice Hall of India, 1990, 0 – 87692 – 145 – 4.
3. R3: Eric Gossett, “Discrete Mathematical Structures with Proofs”, Wiley India Ltd, ISBN:978-81-265-2758-8.
4. R4: Sriram P & Steven S., “Computational Discrete Mathematics”, Cambridge University Press, ISBN 13: 978-0-521-73311-3.
5. R5 : Kenneth H. Rosen, —Discrete Mathematics and its Applications, Tata McGraw-Hill, ISBN 978-0-07-288008-3

Unit No.-I: Set theory and logic

Lecture No.	Details of the Topic to be covered	References
1	Naïve Set Theory (Cantorian Set Theory), Axiomatic Set Theory, Set Operations, Cardinality of set,	T1,T2,R1
2	Principle of inclusion and exclusion.	T1,T2,R1
3	Types of Sets, Bounded and Unbounded Sets, Diagonalization Argument,	T1,T2,R1
4	Countable and Uncountable Sets, Finite and Infinite Sets, Countably Infinite and Uncountably Infinite Sets, Power set	T1,T2,R1
5	logic, Propositional Equivalences, Application of Propositional Logic,	T1,T2,R1
6	Translating English Sentences, Proof by Mathematical Induction and Strong Mathematical Induction.	T1,T2,R1

Question Bank: Theory & Numerical
Mapped to Course Outcome:

$Q.\ \ What\ is\ multiset. Let\ P\ and\ Q\ are\ two\ multiset\ defined\ as\ P = \{a, a, a, c, d, d\}\ and$

$Q = \{a, a, b, c, c\}. Obtain\ Union, intersection\ and\ difference\ of\ two\ multisets\ P\ and\ Q.$

$Q.\ \ Let\ A\ and\ B\ are\ two\ sets. If\ A \subseteq B, then\ prove\ that\ P(A) \subseteq P(B), where\ P(A)\ and\ P(B)$

$are\ power\ sets\ of\ A\ \&\ B.$

$Q.\ \ Let\ A = \{\emptyset, b\}, construct\ the\ following\ sets:$

$a.\ \ A - \emptyset \qquad b.\ \ \{\emptyset\} - A \qquad c.\ \ A \cup P(A), where\ P(A)\ is\ a\ power\ set.$

$Q.\ \ In\ a\ survey\ of\ 260\ college\ students, the\ following\ data\ were\ obtained:$

$64\ had\ taken\ a\ maths\ course, 94\ had\ taken\ a\ CS\ course, 58\ had\ taken\ a\ business\ course,$

$28\ had\ taken\ both\ a\ maths\ and\ a\ business\ course, 26\ had\ taken\ both\ a\ maths\ and\ a\ CS$

$course, 22\ had\ taken\ both\ a\ CS\ and\ a\ business\ course, 14\ had\ taken\ all\ types\ of\ courses.$

$How\ many\ students\ were\ surveyed\ who\ had\ taken\ none\ of\ the\ three\ types\ of\ courses.$

$Q.\ \ 100\ sportsmen\ were\ asked\ whether\ they\ play\ cricket, football\ or\ hockey. Out\ of\ these$

$45\ play\ cricket, 21\ play\ football, 38\ play\ hockey, 18\ play\ cricket\ and\ hockey, 9\ play\ cricket$

$\&\ football, 4\ play\ football\ and\ hockey\ and\ 23\ play\ none\ of\ these. Find\ the\ number\ of$

$sportsmen\ who\ play:$

$a.\quad exactly\ one\ of\ the\ games \qquad b.\quad exactly\ two\ of\ the\ games.$

$Q.\quad Among\ the\ integer\ 1\ to\ 1000$
$: Howmany\ of\ them\ are\ not\ divisible\ by\ 3\ nor\ by\ 5\ nor$

$by\ 7. How\ many\ are\ not\ divisible\ by\ 5\ and\ 7\ but\ divisible\ by\ 3.$

$Q.\quad Prove\ by\ Mathematical\ induction:$

$$\frac{1}{1.3}+\frac{1}{3.5}+\cdots\cdots+\frac{1}{(2n-1)(2n+1)}=\frac{n}{2n+1}\quad for\ \ n\geq 1.$$

$Q.\quad Use\ Mathematical\ induction\ to\ show\ that, for\ all\ n\geq 1, 1+2+3+\cdots\cdots+n$
$=\frac{n(n+1)}{2}.$

$$Q.\quad Prove: \ for\ all\ n\geq 1, 1^3+2^3+3^3+\cdots\cdots+n^3=\left[\ \frac{n(n+1)}{2}\ \right]^2$$

$Q.\quad Show\ that\ for\ natural\ number\ \ n,$

$$1^3+2^3+3^3+\cdots\cdots+n^3=[1+2+3+\cdots\cdots+n]^2$$

$Q.\quad Show\ that: \ \ 7^{2n}+(2^{3n-3})(3^{n-1})\ is\ divisible\ by\ 25\ for\ all\ natural\ number\ n.$

$Q.\quad Use: \ \ p: I\ will\ study\ discrete\ structure, q: I\ will\ go\ to\ a\ movie, r$
$: I\ am\ in\ a\ good\ mood.$

$Write\ the\ english\ sentence\ that\ corresponds\ to\ each\ of\ the\ following:$

$a.\quad \sim r\rightarrow q \qquad b.\quad \sim q\wedge p \qquad c.\quad q\rightarrow \sim p \qquad d.\quad \sim p\rightarrow \sim r$

$Q.\quad Find\ DNF\ of\ \big((p\rightarrow q)\wedge(q\rightarrow p)\big)\vee p.$

$Q.\quad Find\ CNF\ of\ p\leftrightarrow(\sim p\vee \sim q).$

Unit No.-II: Relations and Functions

Lecture No.	Details of the Topic to be covered	References
1	Relations and their Properties, n-ary relations and their applications, Representing relations ,	T1,T2,R1
2	Equivalence relations ,Transitive Closures of relations, Transitive closure by Warshall's algorithm	T1,T2,R1
3	Partial orderings, Partitions, Hasse diagram, Lattices, Chains and Anti-Chains	T1,T2,R1
4	Functions : Surjective, Injective and Bijective functions, Identity function	T1,T2,R1
5	Partial function, Invertible function, Constant function, Inverse functions and Compositions of functions,	T1,T2,R1
6	The Pigeonhole Principle	T1,T2,R1

Question Bank: Theory & Numerical Mapped to Course Outcome:

$Q.\ \ Let\ X = \{1,2,3,4,5,\cdots\cdots,n\}, R = \{(x,y) \mid x - y\ is\ divisible\ by\ 3\}. Show\ that\ R\ is\ equivalence\ relation. Draw\ the\ diagraph\ for\ R\ where\ n = 7.$

$Q.\ \ Relation\ on\ \{1,2,3,4,5\}. If\ relation\ is\ defined\ as$

$\{(1,1),(2,2),(3,3),(4,4),(5,5),(1,5),(5,1),(3,5),(5,3),(1,3),(3,1)\}.$

$Find\ the\ equivalence\ classes.$

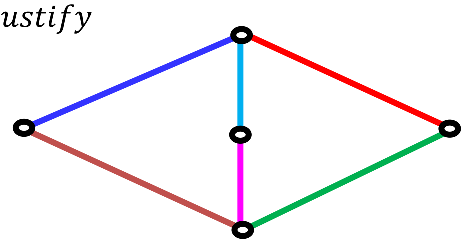
$Q.\ \ A = \{1,2,3,4\}\ and\ B = \{1,4,6,8,9\}; aRb\ iff\ b = a^2. Find\ the\ domain, range\ of\ R. Also$

$find\ its\ relation\ matrix\ and\ draw\ the\ diagraph.$

$Q.\ \ Define\ the\ closure\ of\ Relation. Discuss\ about\ the\ following\ closure\ properties\ with\ examples$

1. $Reflexive\ closure.$ 2. $Symmetric\ closure.$ 3. $Transitive\ closure.$

$Q.\ \ The\ following\ is\ the\ Hasse\ diagram\ of\ the\ poset : \{(a,b,c,d),<\}. Is\ it\ lattice\ ? Justify$



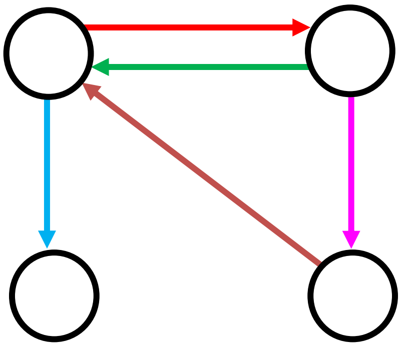
$Q.\ \ Let\ \ A = \{1,2,3,4\}\ consider\ the\ partition\ P = \{\{1,2,3\},\{4\}\}\ of\ A. Find\ the\ equivalence$

$relation\ R\ on\ A\ determined\ by\ P$

$Q.\ \ Let\ \ A = \{1,2,3\}, R\ is\ the\ relation\ on\ A\ whose\ matrix\ is:\ M_R$
$= [\ 1\,1\,1\,0\,0\,1\,0\,0\,1\ \ ]$

$Show\ that\ R\ is\ transitive.$

$Q.\ \ Compute\ the\ transitive\ closure\ of\ given\ diagraph\ using\ Warshall's\ algorithm.$



$Q.\ \ If\ \ R = \{(a,b),(b,a),(b,c),(c,d),(d,a)\}\ be\ the\ relation\ on\ the\ set\ A = \{a,b,c,d\}.$

$Find\ the\ transitive\ closure\ of\ R\ using\ Warshall's\ algorithm.$

$Q.\ \ Let\ \ R = \{(1,4),(2,1),(2,5),(2,4),(4,3),(5,3),(3,2)\}\ on\ the\ set\ A = \{1,2,3,4,5\}. Use$

$Warshall's\ algorithm\ to\ find\ transitive\ closure\ of\ R.$

Unit No.-III: Counting Principles

Lecture No.	Details of the Topic to be covered	References
1	The Basics of Counting, Rule of Sum and Product,	T1,T2,R4
2	Permutations	T1,T2,R4
3	Combinations	T1,T2,R4
4	Binomial Coefficients and Identities	T1,T2,R4
5	Generalized Permutations and Combinations	T1,T2,R4
6	Algorithms for generating Permutations and Combinations.	T1,T2,R4

Question Bank: Theory & Numerical
Mapped to Course Outcome:

$Q.\ \ Write\ an\ algorithm\ for\ generating\ permutations\ of\ \{1,2,3,4,5,\cdots\cdots,n\}. Apply\ it\ for$

$n = 3\ case.$

Q. Find out howmany 5 digit number greter than 30,000 can be formed from the digits 1,2,3,4,5.

Q. The company has 10 members on its board of directors. In how many ways can they elect a president, a vice president, a secretary and a treasurer ?

Q. Solve the following :

1. *How many different car number plates are possible with 2 letters followed by 3 digits*
2. *How many of these numbers plates begin with 'MH'.*

Q. In how many ways can 6 men and 5 women be seates in a line so that no two women sit together ? In how many ways can 6 men and 5 women sit in a line so that women occupy the even places.

Q. If the letters of the word 'REGULATIONS' be arranged at random. What is the chance that there will be exactly 4 letters between R & E ?

Q. From a group of 7 men and 6 women, five persons are to be selected to form a committee so that at least 3 men are there on the committee. In how may ways can it be done ?

Q. A bag contains 6 red and 8 green balls.

1. *If one ball is drawn at random, then what is the probability of the ball being green ?*
2. *If two ball is drawn at random, then what is the probability that one is red and the other is green ?*

Q. In the expansion of $(1+x)^6$, *what is the coefficient of* x^3.

Q. Use Binomial theorem to expand $(x^4+2)^3$.

Unit No.-IV: Graph Theory

Lecture No.	Details of the Topic to be covered	References
1	Graph Terminology and Special Types of Graphs,	T1,T2,R2
2	Representing Graphs and Graph Isomorphism,	T1,T2,R2

3	Connectivity, Euler and Hamilton Paths,	T1,T2,R2
4	The handshaking lemma, Single source shortest path,	T1,T2,R2
5	Dijkstra's Algorithm,	T1,T2,R2
6	Planar Graphs, Graph Colouring.	T1,T2,R2

Question Bank: Theory & Numerical
Mapped to Course Outcome:

Q. Explain the directed and undirected graph with suitable example.

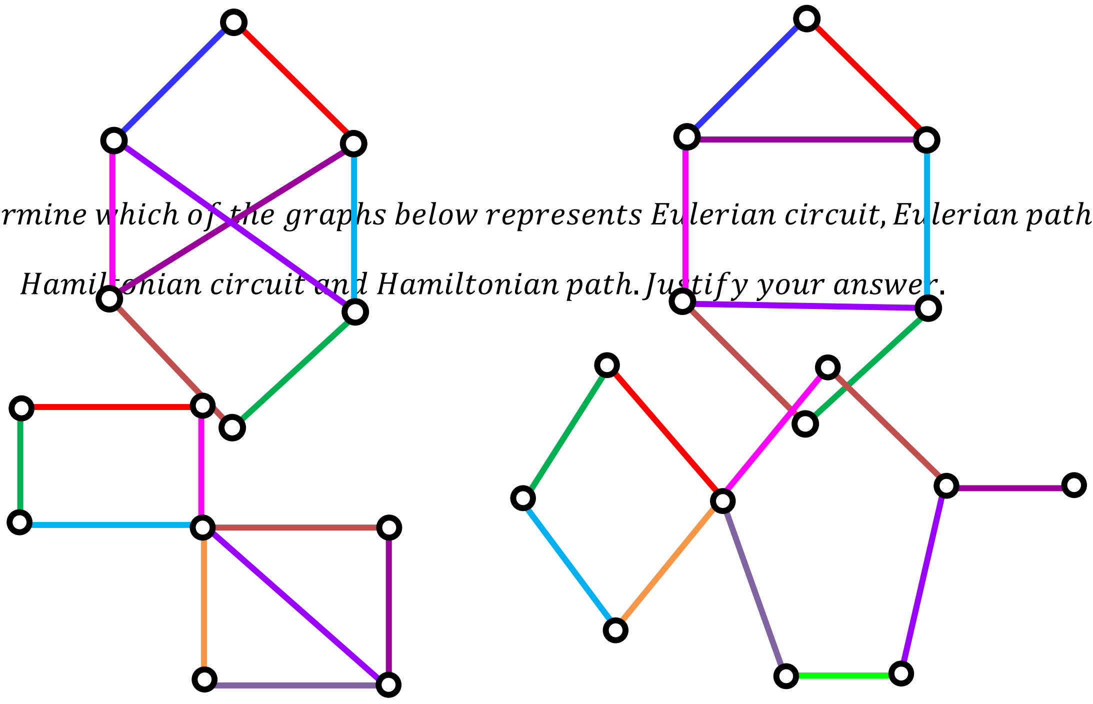
Q. Explain the following in brief :

1. *Subgraphs and spanning subgraph.*
2. *Isomorphic graph.*
3. *Bipartite graph.*
4. *Adjacency matrix & incidence matrix of undirected graph.*

Q. Consider the graph $G(V,E)$ *where* $V = \{v_1, v_2, v_3\}$ & $deg(v_2) = 4$

1. *Does such simple graph exists? If not, why?*
2. *Does such a multigraph exist? If yes, give example.*

Q. Determine whether the two graphs are isomorphic or not. Explain.



Q. Determine which of the graphs below represents Eulerian circuit, Eulerian path, Hamiltonian circuit and Hamiltonian path. Justify your answer.

Q. Use Dijkstra's algorithm to find the shortest path between a and 2.

$Q.\ \ Define\ the\ graph\ K_n\ and\ K_{mn}.$

$Q.\ \ How\ many\ colours\ required\ to\ colour\ \ K_{mn}, why?$

$Q.\ \ Under\ what\ condition\ \ K_{mn}\ will\ have\ Eulerian\ circuit?$

$Q.\ \ Find\ under\ what\ conditions\ K_{m,n}\ the\ complete\ bipartite\ graph\ will\ have\ an\ Eulerian\ circuit.$

Unit No.-V: Trees

Lecture No.	Details of the Topic to be covered	References
1	Introduction, Properties of trees,	T1,T2,R1
2	Binary search tree, Tree traversal, Decision tree,	T1,T2,R1
3	Prefix codes and Huffman coding, Cut sets	T1,T2,R1
4	Spanning Trees and Minimum Spanning Tree,	T1,T2,R1
5	Kruskal‘s and Prim‘s algorithms,	T1,T2,R1
6	The Max flow- Min Cut Theorem (Transport network).	T1,T2,R1

Question Bank: Theory & Numerical Mapped to Course Outcome:

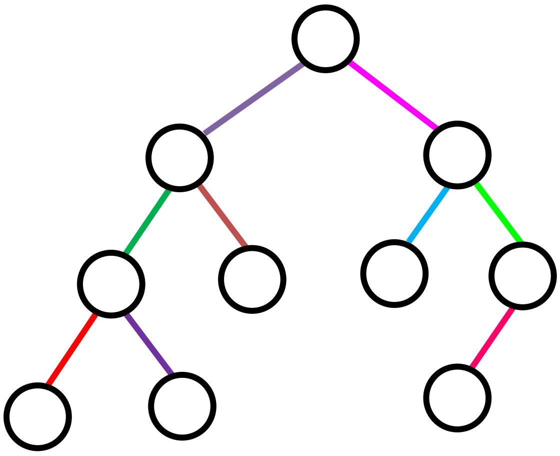
$Q.\ \ Define\ the\ following\ terms\ with\ examples:$

$(a)\ \ Rooted\ tree\qquad (b)\ \ Optimal\ binary\ tree\qquad (c)\ \ Height\ of\ the\ tree.$

$Q.\ \ Explain$

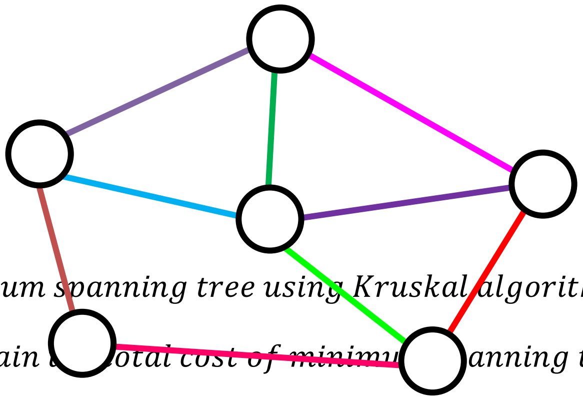
$:\ \ \ (a)\ \ Binary\ search\ tree\qquad (b)\ \ Subtree\qquad (c)\ \ Regular\ m\ ary\ tree$

Q. Find the preorder, postorder and inorder traversals of the following tree :



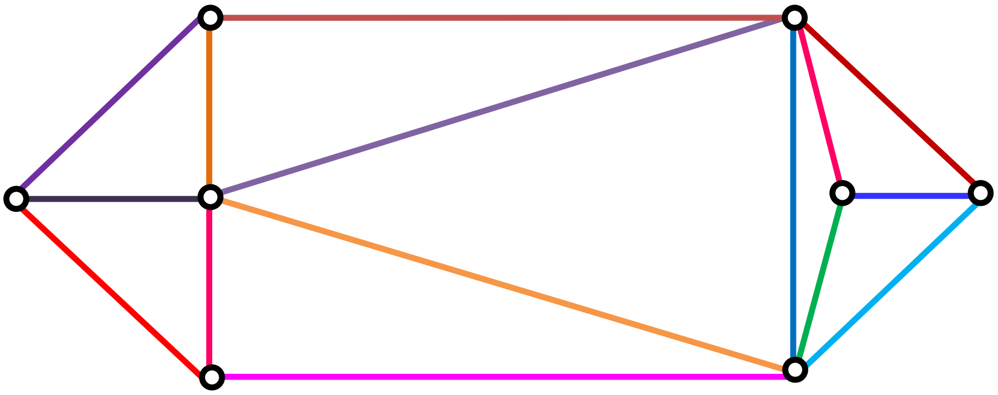
Q. Explain the following terms : 1. Game tree. 2. Kruskal's algorithm.

Q. Give the stepwise construction of minimum spanning tree using Prim's Algorithm for the following graph shown in Fig. Obtain the total cost of minimum spanning tree.

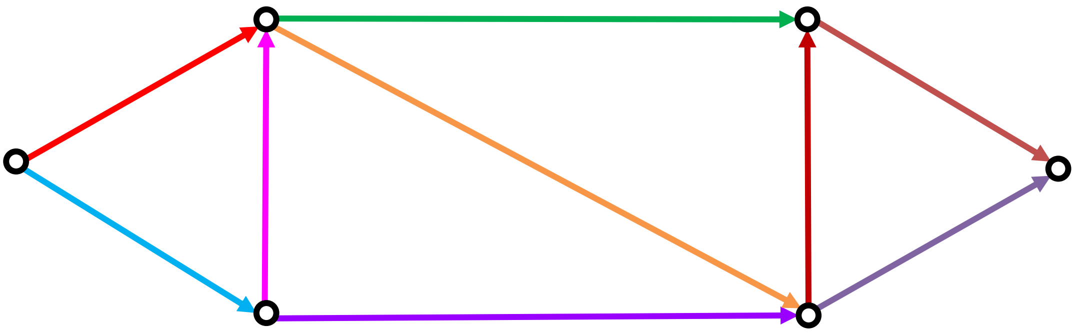


Q. Obtain the minimum spanning tree using Kruskal algorithm for the following graph.

Obtain the total cost of minimum spanning tree.

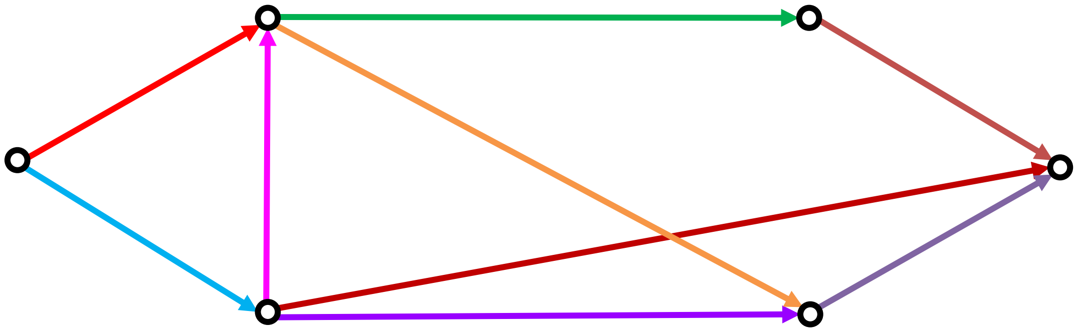


Q. Find maximum flow in the transport network using lebeling procedure. Determine the coresponding min cut.



$Q.\ \ Find\ maximum\ flow\ in\ the\ transport\ network\ using\ labeling\ procedure. Determine$

$the\ coresponding\ min\ cut.$



Unit No.-VI : Algebraic Structures and Coding Theory

Lecture No.	Details of the Topic to be covered	References
1	The structure of algebra, Algebraic Systems, Semi Groups,	T1,T2,R1
2	Monoids, Groups, Homomorphism,	T1,T2,R1
3	Normal Subgroups, Congruence relations,	T1,T2,R1
4	Rings, Integral Domains, Fields,	T1,T2,R1
5	Coding theory, Polynomial Rings and polynomial Codes,	T1,T2,R1
6	**Galois Theory** : Field Theory and Group Theory	T1,T2,R1

Question Bank: Theory & Numerical Mapped to Course Outcome:

$Q.\ \ Explain, in\ brief. the\ following\ terms, giving\ suitable\ example:$

$(a)\ \ Monoid$

$(b)\ \ Cyclic\ group$

$(c)\ \ Ring.$

$Q.\ \ Explain, in\ brief, the\ following\ with\ suitable\ example:$

$(a)\ \ Homomorphism\ of\ Groups$

$(b)\ \ Integral\ Domain$

$(c)\quad Group\ codes.$

$Q.\quad Let\ (Q,*)\ is\ an\ Algebraic\ system.\ \ *\ is\ a\ binary\ operation\ defined\ as$

$a * b\ =\ a\ +\ b\ -\ ab\ for\ every\ a,b \in Q. Determine\ whether\ (Q,*)\ is\ a\ group.$

$Q.\quad Let\ R\ = \{0, 60, 120, 180, 240, 300\}\ and\ \ * =\ binary\ operation\ so\ that\ for\ a\ and\ b$

$in\ R, a$
$*\ b\ is\ overall\ angular\ rotation\ corresponding\ to\ successive\ rotation\ by\ a\ and$

$by\ b. Show\ (R,*)\ is\ a\ group.$

$Q.\quad Let\ Z_n = \{0,1,2,\cdots\cdots,,n-1\}. In\ Z_{12}\ what\ is\ the\ order\ of\ 3,6\ and\ 8.$

$Q.\quad Let\ Z_n = \{0,1,2,\cdots\cdots,,n-1\}. Let\ \ *\ be\ a\ binary\ operation\ such\ that$

$a * b = remainder\ of\ (a+b)\ divided\ by\ n. Construct\ a\ table\ for\ n = 4.$

$Is\ (Z_4,*)\ a\ monoid, semigroup, group\ and\ abelian\ group.$

$Q.\quad Prove\ that\ \left((a+b\sqrt{2}),+,*\right), where\ a,b \in R\ is\ integral\ domain.$

$Q.\quad Explain\ Galois\ theory.$

$Q.\quad Discuss\ in\ brief\ about\ the\ Galois\ theory, Field\ theory\ and\ group\ theory.$

$Q.\quad Show\ that\ (I,\oplus,\odot)\ is\ a\ commutative\ ring\ with\ identity\ where\ \oplus\ and\ \odot\ are\ defined\ as$
:

$$a\oplus b = \ a+b-1\ and\ a\odot b = \ a+\ b$$

$Q.\quad Prove\ that\ the\ set : A = \{0,2,4,6,8\}\ with\ +_{10}\ and\ \ \times_{10}\ \ operation. i.e.\ \ R$
$= \{A, +_{10}, \times_{10}\}\ is$

$a\ ring.$

Subject 2: Fundamental of Data Structure

Weekly Work Loads(in Hrs)	Lecture	Tutorial	Practical
	03	-	-

Online/ In-sem	Theory	Practica l	Oral	Term-work	Total Marks	Credit
30	70	0	0	0	100	03

Course Objectives:

The course is intended to provide the foundations of the practical implementation and usage of Data Structures and Algorithms to ensure that the learner evolves into a competent programmer capable of designing and analyzing implementations of data structures and algorithms for different kinds of problems.

- To understand the standard and abstract data representation methods.
- To acquaint with the structural constraints and advantages in usage of the data.
- To understand various data structures, operations on it and the memory requirements
- To understand various data searching and sorting methods.
- To understand various algorithmic strategies to approach the problem solution

Course Outcomes:

- On completion of the course, learner will be able to–
- **CO1:Design** the algorithms to solve the programming problems, **identify** appropriate algorithmic strategy for specific application, and **analyze** the time and space complexity.
- **CO2:Discriminate** the usage of various structures, **Design/Program/Implement** the appropriate data structures; use them in implementations of abstract data types and Identity the appropriate data structure in approaching the problem solution.
- **CO3:Demonstrate** use of sequential data structures- Array and Linked lists to store and process data.
- **CO4:Understand** the computational efficiency of the principal algorithms for searching and sorting and choose the most efficient one for the application.
- **CO5:Compare** and **contrast** different implementations of data structures (dynamic and static).
- **CO6:Understand, Implement and apply** principles of data structures-stack and queue to solve computational problems.

Syllabus

Unit	Course Content	Hours
I	**Introduction to Algorithm and Data Structures**	
	Introduction: From Problem to Program (Problem, Solution, Algorithm, Data Structure and Program). Data Structures: Data, Information, Knowledge, and Data structure, Abstract Data Types (ADT), Data Structure Classification (Linear and Non-linear, Static and Dynamic, Persistent and Ephemeral data structures). **Algorithms:** Problem Solving, Introduction to algorithm, Characteristics of algorithm, Algorithm design tools: Pseudo-code and flowchart. **Complexity of algorithm:** Space complexity, Time complexity, Asymptotic notation- Big-O, Theta and Omega, finding complexity using step count method, Analysis of programming constructs-Linear, Quadratic, Cubic, Logarithmic. **Algorithmic Strategies:** Introduction to algorithm design strategies- Divide and Conquer, and Greedy strategy.	07
II	**Linear Data Structure Using Sequential Organization**	
	Concept of Sequential Organization, Overview of Array, Array as an Abstract Data Type, Operations on Array, Merging of two arrays, Storage Representation and their Address Calculation: Row major and Column Major, Multidimensional Arrays: Two-dimensional arrays, n-dimensional arrays. Concept of Ordered List, **Single Variable Polynomial**: Representation using arrays, Polynomial as array of structure, Polynomial addition, Polynomial multiplication. **Sparse Matrix:** Sparse matrix representation using array, Sparse matrix addition, Transpose of sparse matrix- Simple and Fast Transpose, Time and Space tradeoff.	07
III	**Searching and Sorting**	
	Searching: Search Techniques-Sequential Search/Linear Search, Variant of Sequential Search- Sentinel Search, Binary Search, Fibonacci Search, and Indexed Sequential Search. **Sorting:** Types of Sorting-Internal and External Sorting, General Sort Concepts-Sort Order, Stability, Efficiency, and Number of Passes, Comparison Based Sorting Methods-Bubble Sort, Insertion Sort, Selection Sort, Quick Sort, Shell Sort, Non-comparison Based Sorting Methods-Radix Sort, Counting Sort, and Bucket Sort, Comparison of All Sorting Methods and their complexities	07
IV	**Linked List**	
	Introduction to Static and Dynamic Memory Allocation, **Linked List:** Introduction, of Linked Lists, Realization of linked list using dynamic memory management,operations, Linked List as ADT, **Types of Linked List:** singly linked, linear and Circular Linked Lists, Doubly Linked List, Doubly Circular Linked List, Primitive Operations on Linked List-Create, Traverse, Search, Insert, Delete, Sort, Concatenate. Polynomial	07

	Manipulations- Polynomial addition. Generalized Linked List (GLL) concept, Representation of Polynomial using GLL.	
V	**Stack**	
	Introduction, properties of trees, Binary search tree, tree traversal, decision tree, prefix codes and Huffman coding, cut sets, Spanning Trees and Minimum Spanning Tree, Kruskal‘s and Prim‘s algorithms, The Max flow- Min Cut Theorem (Transport network).	07
VI	**Queue**	
	The structure of algebra, Algebraic Systems, Semi Groups, Monoids, Groups, Homomorphism and Normal Subgroups, and Congruence relations, Rings, Integral Domains and Fields, Coding theory, Polynomial Rings and polynomial Codes, Galois Theory –Field Theory and Group Theory.	07

Teaching Plan

Sr. No.	Unit	Topics to be covered	Book Referred	Total Lecture Planned
1	I	The Internet, basic internet protocols, the World Wide Web, HTTP Request message, HTTP response message, web clients, web servers.HTML: Introduction, history and versions.HTML elements: headings, paragraphs, line break, colors and fonts, links, frames, lists, tables, images and forms, Difference between HTML and HTML5. CSS: Introduction to Style Sheet, CSS features, CSS core syntax, Style sheets and HTML, Style rule cascading and inheritance, text properties. Bootstrap.	1.1,2.1	7
2	II	JavaScript: Introduction to JavaScript, JavaScript in perspective, basic syntax, variables and data types, statements, operators, literals, functions, objects, arrays, built in objects, JavaScript debuggers. DOM: Introduction to Document Object Model, DOM history and levels, intrinsic event handling, modifying element style, the document tree, DOM event handling, jQuery, Overview of Angular JS.	1.1,1.2,2.1	7
3	III	Servlet: Servlet architecture overview, A “Hello World” servlet, Servlets generating dynamic content, Servlet life cycle, parameter data, sessions, cookies, URL rewriting, other Servlet capabilities, data storage, Servlets concurrency, databases (MySQL) and Java Servlets. XML: XML documents and vocabularies, XML declaration, XML Namespaces, DOM based XML processing, transforming XML documents, DTD: Schema, elements, attributes. AJAX: Introduction, Working of AJAX.	1.1,1.2	7
4	IV	JSP: Introduction to Java Server Pages, JSP and Servlets, running JSP applications, Basic JSP, JavaBeans classes and JSP, Support for the Model-View-Controller paradigm, JSP related technologies. Web Services: Web Service concepts, Writing a Java Web Service, Writing a Java web service client, Describing Web Services: WSDL, Communicating Object data: SOAP. Struts: Overview, architecture,	1.1,1.2	7

		configuration, actions, interceptors, result types, validations, localization, exception handling, annotations.		
5	V	PHP: Introduction to PHP, uses of PHP, general syntactic characteristics, Primitives, operations and expressions, output, control statements, arrays, functions, pattern matching, form handling, files, cookies, session tracking, using MySQL with PHP, WAP and WML. Introduction to ASP.NET: Overview of the .NET Framework, Overview of C#, Introduction to ASP.NET, ASP.NET Controls, Web Services. Overview of Node JS.	1.1,1.2,2.1	7
6	VI	Introduction to Ruby: Origins & uses of Ruby, scalar types and their operations, simple input and output, control statements, fundamentals of arrays, hashes, methods, classes, code blocks and iterators, pattern matching. Introduction to Rails: Overview of Rails, Document Requests, Processing Forms, Rails Applications and Databases, Layouts, Rails with Ajax. Introduction to EJB.	1.2	7

1. Text Books:

1. Jeffrey C.Jackson, "Web Technologies: A Computer Science Perspective", Second Edition, Pearson Education, 2007, ISBN 978-0131856035

2. Robert W. Sebesta,“ Programming the World Wide Web”, 4th Edition, Pearson education, 2008

2. Reference Books:

1. Marty Hall, Larry Brown, “Core Web Programming", Second Edition, Pearson Education, 2001, ISBN 978-0130897930.

2. H.M. Deitel, P.J. Deitel and A.B. Goldberg, "Internet & World Wide Web How To Program", Third Edition, Pearson Education, 2006, ISBN 978-0131752429.

3. Chris Bates, “Web Programming Building Internet Applications”, 3rd Edition, Wiley India, 2006.

4. Xue Bai et al, “The web Warrior Guide to Web Programming”, Thomson, 2003.

Reference Web Links/ Research Paper/ Referred Book other than Mention in Syllabus:

- ***e-Books:***

1. https://www.w3.org/html/
2. HTML, The Complete Reference http://www.htmlref.com/
3. http://w3schools.org/
4. http://php.net/
5. https://jquery.com/
6. https://developer.mozilla.org/en-US/docs/AJAX
7. http://www.tutorialspoint.com/css/

- ***MOOC/ Video Lectures available at:***

1. http://www.nptelvideos.in/2012/11/internet-technologies.html
2. https://freevideolectures.com/course/2308/internet-technology/25video lecture by Prof. Indranil Sengupta, IIT, Kharagpur
3. https://www.digimat.in/nptel/courses/video/106105191/L01.html
4. http://www.nptelvideos.com/php/php_video_tutorials.php

Unit No.-I-

Name of the Unit: Web Essentials and Mark-up language- HTML

Lecture No.	Details of the Topic to be covered	References
1	The Internet, basic internet protocols, the World Wide Web, HTTP Request message, HTTP response message, web clients, web servers	1.1
2	Introduction, history and versions.HTML elements: headings, paragraphs, line break, colors and fonts,	1.1, 2.1
3	Links, frames, lists, tables	1.1, 2.1
4	Images and forms, Difference between HTML and HTML5.	1.1, 2.2
5	Introduction to Style Sheet, CSS features, CSS core syntax, Style sheets and HTML	1.1, 2.3
6	Style rule cascading and inheritance, text properties	1.1, 2.2
7	Bootstrap	1.1, 2.2

Question Bank: Theory & Programs
Mapped to Course Outcome:

Q. 1	Write a difference between HTML and HTML5.

Q. 2	Explain features of CSS.

Q. 3	Explain HTPP request and response message.

Q. 4	Explain text properties of CSS with example.

Q. 5	Give a brief about Bootstrap.

Unit No.-II-

Name of the Unit: Client Side Technologies: JavaScript and DOM

Lecture No.	Details of the Topic to be covered	References
1	Introduction to JavaScript, JavaScript in perspective, basic syntax	1.1, 1.2
2	variables and data types, statements, operators, literals	1.1, 1.2, 2.3
3	functions, objects, arrays,	1.1, 1.2, 2.4
4	built in objects, JavaScript debuggers	1.2, 2.4
5	Introduction to Document Object Model, DOM history and levels	1.1, 1.2, 2.3
6	Intrinsic event handling, modifying element style, the document tree	1.1, 1.2, 2.2
7	DOM event handling, jQuery, Overview of Angular JS	1.1, 1.2, 2.2

Question Bank: Theory & Programs
Mapped to Course Outcome:

Q. 1	Explain JavaScript in detail.

Q. 2	What is DOM? Explain levels of DOM

Q. 3	Write a note on functions in JS.

Q. 4	Explain jQuery.

Q. 5	Write a note on Angular JS.

Unit No.-III

Name of the Unit: Java Servlets and XML

Lecture No.	Details of the Topic to be covered	References
1	Servlet architecture overview, A "Hello World" servlet, Servlets generating dynamic content	1.1, 1.2, 2.4
2	Servlet life cycle, parameter data, sessions, cookies	1.1, 1.2, 2.4
3	URL rewriting, other Servlet capabilities, data storage	1.1, 1.2, 2.4
4	Servlets concurrency, databases (MySQL) and Java Servlets	1.1, 1.2, 2.4
5	XML documents and vocabularies, XML declaration, XML Namespaces	1.1, 1.2, 2.3
6	DOM based XML processing, transforming XML documents, DTD: Schema, elements, attributes	1.1, 1.2, 2.3
7	AJAX: Introduction, Working of AJAX	1.1, 1.2, 2.1

Question Bank: Theory & Programs
Mapped to Course Outcome:

Q. 1	Differentiate between HTML and XML
Q. 2	Write a Java Servlet which will display "Hello Welcome" message.
Q. 3	What is AJAX? Explain Working of AJAX.
Q. 4	How do servlet work?
Q. 5	Explain life cycle of sevlet.

Unit No.-IV

Name of the Unit: JSP and Web Services

Lecture No.	Details of the Topic to be covered	References
1	Introduction to Java Server Pages, JSP and Servlets, running JSP applications, Basic JSP	1.1, 1.2, 2.2
2	JavaBeans classes and JSP, Support for the Model-View-Controller paradigm, JSP related technologies	1.1, 1.2, 2.2
3	Web Services: Web Service concepts, Writing a Java Web Service,	1.1, 1.2, 2.2
4	Writing a Java web service client, Describing Web Services: WSDL	1.1, 1.2, 2.2
5	Describing Web Services: WSDL, Communicating Object data: SOAP	1.1, 1.2, 2.2
6	Struts: Overview, architecture, configuration, actions, interceptors, result types	1.1, 1.2, 2.2
7	validations, localization, exception handling, annotations	1.1, 1.2, 2.2

Question Bank: Theory & Programs
Mapped to Course Outcome:

Q. 1	Draw and explain neat diagram which depict MVC to the struts architecture.
Q. 2	What are the Web services? List and explain layers in protocol stack of web services architecture.
Q. 3	What is JSP? Enlist advantages of JSP over CGI
Q. 4	Write a note on WSDL.
Q. 5	Identify and explain data tags from struts 2 which are used to manipulate data displayed on any web application page.

Unit No.-V

Name of the Unit: Server Side Scripting Languages

Lecture No.	Details of the Topic to be covered	References
1	Introduction to PHP, uses of PHP, general syntactic characteristics, Primitives	1.1, 1.2, 2.1
2	operations and expressions, output, control statements, arrays	1.1, 1.2, 2.2
3	functions, pattern matching, form handling, files, cookies	1.1, 1.2, 2.2
4	session tracking, using MySQL with PHP, WAP and WML	1.1, 1.2, 2.2
5	Overview of the .NET Framework, Overview of C#,	1.1, 1.2, 2.2
6	Introduction to ASP.NET, ASP.NET Controls	1.1, 1.2, 2.2
7	Web Services. Overview of Node JS	1.1, 1.2, 2.2

Question Bank: Theory & Programs
Mapped to Course Outcome:

Q. 1	What are the benefits of using PHP?
Q. 2	How is MySQL Database connected to PHP? Explain PHP OOBC in detail
Q. 3	Classify data types of PHP and describe various data types in each types
Q. 4	What is a session? List the session tracking techniques.
Q. 5	Explain Node JS.

Unit No. -VI

Name of the Unit: Ruby and Rails

Lecture No.	Details of the Topic to be covered	References
1	Origins & uses of Ruby, scalar types and their operations, simple input and output	1.2, 2.2
2	control statements, fundamentals of arrays, hashes	1.2, 2.2
3	methods, classes, code blocks and iterators, pattern matching	1.2, 2.2
4	Overview of Rails, Document Requests	1.2, 2.2
5	Processing Forms, Rails Applications and Databases	1.2, 2.2
6	Layouts, Rails with Ajax	1.2, 2.2
7	Introduction to EJB	1.2, 2.2

Question Bank: Theory & Programs
Mapped to Course Outcome:

Q. 1	Explain input and output in Ruby.

Q. 2	Draw and explain scenario of client accessing remote EJB.

Q. 3	Explain Rails application and databases.

Q. 4	List some EJB clients

Q. 5	What are hashes in Ruby?

Subject 3: Object Oriented Programming

Weekly Work Loads(in Hrs)	Lecture	Tutorial	Practical
	03	-	-

Online/ In-sem	Theory	Practical	Oral	Term-work	Total Marks	Credit
30	70	0	0	0	100	03

Course Objectives

The course is intended to provide the foundations and in-depth understanding of a modern object- oriented language and develop skills in software development, through an algorithmic approach and the application of principles of objected oriented programming.

- To learn the object-oriented programming paradigm, focusing on the definition and use of classes along with the fundamentals of object-oriented design.
- To learn the syntax and semantics of the C++ programming language.
- To understand the concept of data abstraction and encapsulation, how to design C++ classes for code reuse, how to implement copy constructors and class member functions, to overload functions and operators in C++.
- To learn how inheritance and virtual functions implement dynamic binding with polymorphism.
- To learn how to design and implement generic classes with C++ templates and how to use exception handling in C++ programs.

Course Outcomes

At the end of the Course, Students will be able to,

- Apply constructs- sequence, selection and iteration; classes and objects, inheritance, use of predefined classes from libraries
- Design object-oriented solutions for small systems involving multiple objects.
- Use virtual and pure virtual function and complex programming situations
- Apply object-oriented software principles in problem solving.
- Analyze the strengths of object-oriented programming.
- Develop the application using object oriented programming language(C++).

SYLLABUS

UNIT – I

Topics – Introduction to object-oriented programming, Need of object-oriented programming, Fundamentals of object-oriented programming: Namespaces, objects, classes, data members, methods, messages, data encapsulation, data abstraction and information hiding, inheritance, polymorphism. Benefits of OOP, C++ as object oriented programming language.

C++ Programming- C++ programming Basics, Data Types, Structures, Enumerations, control structures, Arrays and Strings, Class, Object, class and data abstraction, Access specifiers, separating interface from implementation.

Functions- Function, function prototype, accessing function and utility function, Constructors and destructor, Types of constructor, Objects and Memory requirements, Static members: variable and functions, inline function, friend function.

At the end of this unit students will be able to -		**No. of Lectures** – 07
Sr. No.	**Learning Outcomes**	**Bloom's taxonomy Level**
1	Learner will be able to understand fundamentals of object oriented programming and can interpret class .	L1

UNIT – II

Topics – Inheritance- Base Class and derived Class, protected members, relationship between base Class and derived Class, Constructor and destructor in Derived Class, Overriding Member Functions, Class Hierarchies, Public and Private Inheritance, Types of Inheritance, Ambiguity in Multiple Inheritance, Virtual Base Class, Abstract class, Friend Class, Nested Class.

Pointers: declaring and initializing pointers, indirection Operators, Memory Management: new and delete, Pointers to Objects, this pointer, Pointers Vs Arrays, accessing Arrays using pointers, Arrays of Pointers, Function pointers, Pointers to Pointers, Pointers to Derived classes, Passing pointers to functions, Return pointers from functions, Null pointer, void pointer.

At the end of this unit students will be able to -		**No. of Lectures** – 07
Sr. No.	**Learning Outcomes**	**Bloom's taxonomy Level**
1	Learner will be able to understand fundamentals of object oriented programming and can interpret class, data types .	L1
2	Learner will be able to apply the Inheritance concept, Virtual Base class, Friend class, Abstract class in real time application.	L2
3	Learner will be able to use feature of pointer to objects, Function pointers, pointer to derived classes.	L3

UNIT – III

Topics – Polymorphism- Introduction to Polymorphism, Types of Polymorphism, Operator Overloading- concept of overloading, operator overloading, Overloading Unary Operators, Overloading Binary Operators, Data Conversion, Type casting (implicit and explicit), Pitfalls of Operator Overloading and Conversion, Keywords explicit and mutable.

Function overloading, Run Time Polymorphism- Pointers to Base class, virtual function and its significance in C++, pure virtual function and virtual table, virtual destructor, abstract base class.

At the end of this unit students will be able to -		**No. of Lectures** – 07
Sr. No.	**Learning Outcomes**	**Bloom's taxonomy Level**
1	Learner will be able to understand fundamentals of object oriented programming and can interpret class, data types .	L1
2	Learner will be able to use the concept of Polymorphism to develop program involving multiple classes.	L2

3	Learner will be able to concept of pointer to base class and virtual function in a programs.	L3

UNIT – IV **Topics –** Data hierarchy, Stream and files, Stream Classes, Stream Errors, Disk File I/O with Streams, File Pointers, and Error Handling in File I/O, File I/O with Member Functions, Overloading the Extraction and Insertion Operators, memory as a Stream Object, Command-Line Arguments, Printer output.		
At the end of this unit students will be able to -		**No. of Lectures** – 07
Sr. No.	**Learning Outcomes**	**Bloom's taxonomy Level**
1	Learner will be able to understand fundamentals of object oriented programming and can interpret class, data types .	L1
2	Learner will be able to use concept of File stream in Disk Handling , Error Handling in I/O.	L2
3	Learner will be able to Apply concept of overloading of Insertion and Extraction operators.	L3
4	Learner will be able to Understand the concepts of Command line arguments	L5
UNIT – V **Topics –** Exception Handling- Fundamentals, other error handling techniques, simple exception handling- Divide by Zero, Multiple catching, re-throwing an exception, exception specifications, user defined exceptions, processing unexpected exceptions, constructor, destructor and exception handling, exception and inheritance. Templates- The Power of Templates, Function template, overloading Function templates, and class template, class template and Nontype parameters, template and friends Generic Functions, The type name and export keywords.		
At the end of this unit students will be able to -		**No. of Lectures** – 07
Sr. No.	**Learning Outcomes**	**Bloom's taxonomy Level**
1	Learner will be able to fundamentals of Exception handling and can implement different types of exception in programming.	L2
2	Learner will be able to apply try catch block in programming to handle exceptions.	L3
3	Learner will be able to Create template, Function template, Class Template and develop program for that.	L6

UNIT – VI **Topics –** Introduction to STL, STL Components, Containers- Sequence container and associative containers, container adapters, Application of Container classes: vector, list, Algorithms- basic searching and sorting algorithms, min-max algorithm, set operations, heap sort, Iterators- input, output, forward, bidirectional and random access. Object Oriented Programming – a road map to future		
At the end of this unit students will be able to -		**No. of Lectures** – 06
Sr. No.	**Learning Outcomes**	**Bloom's taxonomy Level**
1	Learner will be able to Identify the STL Components, Container, Associative Container and Container classes .	L2
2	Learner will be able to use Container Classes such as vector and list in programming.	L3
3	Learner will be able to Create effective programs using searching and sorting algorithm in real time application to solve complex problem.	L5
4	Learner will be able to use Iterator and Set operations to solve complex problem.	L5

Teaching Plan

Sr. No.	Unit	Topics to be covered	Book Referred	Total Lecture Planned
1	I	Introduction to object-oriented programming, Need of object-oriented programming, Fundamentals of object-oriented programming: Namespaces, objects, classes, data members, methods, messages, data encapsulation, data abstraction and information hiding, inheritance, polymorphism. Benefits of OOP, C++ as object oriented programming language. C++ Programming- C++ programming Basics, Data Types, Structures, Enumerations, control structures, Arrays and Strings, Class, Object, class and data abstraction, Access specifiers, separating interface from implementation. Functions- Function, function prototype, accessing function and utility function, Constructors and destructor, Types of constructor, Objects and Memory requirements, Static members: variable and functions, inline function, friend function.	1.1,1.2,2.1	7
2	II	Inheritance- Base Class and derived Class, protected members, relationship between base Class and derived Class, Constructor and destructor in Derived Class, Overriding Member Functions, Class Hierarchies, Public and Private Inheritance, Types of Inheritance, Ambiguity in Multiple Inheritance, Virtual Base Class, Abstract class, Friend Class, Nested Class. Pointers: declaring and initializing pointers, indirection Operators, Memory Management: new and delete, Pointers to Objects, this pointer, Pointers Vs Arrays, accessing Arrays using pointers, Arrays of Pointers, Function pointers, Pointers to Pointers, Pointers to Derived classes, Passing pointers to functions, Return pointers from functions, Null pointer, void pointer.	1.1,1.2, 2.1	7
3	III	Polymorphism- Introduction to Polymorphism, Types of Polymorphism, Operator Overloading- concept of overloading, operator overloading, Overloading Unary Operators, Overloading Binary Operators,	1.1,1.2,2.1 ,2.2,2.4	7

		Data Conversion, Type casting (implicit and explicit), Pitfalls of Operator Overloading and Conversion, Keywords explicit and mutable. Function overloading, Run Time Polymorphism-Pointers to Base class, virtual function and its significance in C++, pure virtual function and virtual table, virtual destructor, abstract base class.		
4	IV	Data hierarchy, Stream and files, Stream Classes, Stream Errors, Disk File I/O with Streams, File Pointers, and Error Handling in File I/O, File I/O with Member Functions, Overloading the Extraction and Insertion Operators, memory as a Stream Object, Command-Line Arguments, Printer output.	1.1, 1.2, 2.1,2.2,2.3	7
5	V	Exception Handling- Fundamentals, other error handling techniques, simple exception handling-Divide by Zero, Multiple catching, re-throwing an exception, exception specifications, user defined exceptions, processing unexpected exceptions, constructor, destructor and exception handling, exception and inheritance. Templates- The Power of Templates, Function template, overloading Function templates, and class template, class template and Nontype parameters, template and friends Generic Functions, The type name and export keywords.	1.2, 2.1, 2.2,2.3,2.4	7
6	VI	Introduction to STL, STL Components, Containers-Sequence container and associative containers, container adapters, Application of Container classes: vector, list, Algorithms- basic searching and sorting algorithms, min-max algorithm, set operations, heap sort, Iterators-input, output, forward, bidirectional and random access. Object Oriented Programming – a road map to future	2.1, 2.2,2.3,2.4	7

1. Text Books:

1. Deitel,“C++ How to Program”, 4th Edition, Pearson Education, ISBN:81-297-0276-2.

2. Robert Lafore, “Object-Oriented Programming in C++‖”, fourth edition, Sams Publishing,

ISBN:0672323087 (ISBN 13: 9780672323089)

2. Reference Books:

1. Herbert Schildt, “C++-The complete reference”‖, Eighth Edition, McGraw Hill Professional, 2011, ISBN:978-00-72226805

2. Matt Weisfeld, “The Object-Oriented Thought Process”, Third Edition Pearson ISBN-13:075- 2063330166

3. E.Balagurusamy, “Object-Oriented Programming with C++”, 7th edition, Graw-Hill Publication, ISBN 10: 9352607996 ISBN 13: 9789352607990

4. Cox Brad, Andrew J. Novobilski, “Object –Oriented Programming: An Evolutionary Approach”‖, Second Edition, Addison–Wesley, ISBN:13:978-020-1548341

Reference Web Links/ Research Paper/ Referred Book other than Mention in Syllabus:

- *e-Books:*

Object-Oriented Programming with C++, 7e - E Balagurusamy - Google Books

C++ AND OBJECT-ORIENTED PROGRAMMING PARADIGM - Google Books

- *MOOC/ Video Lectures available at:*

https://nptel.ac.in/courses/106/105/106105151/
https://swayam.gov.in/nd1_noc20_cs07/preview
https://www.classcentral.com/course/swayam-programming-in-c-6704

Unit No.-I-

Name of the Unit : Fundamentals of Object Oriented Programming

Lecture No.	Details of the Topic to be covered	References
1	Introduction to object-oriented programming, Need of object-oriented programming, Fundamentals of object-oriented programming: Namespaces, objects, classes, data members, methods, messages	1.1,1.2
2	Data encapsulation, data abstraction and information hiding, inheritance, polymorphism	1.1,1.2
3	Benefits of OOP, C++ as object oriented programming language. C++ Programming- C++ programming Basics, Data Types, Structures,	1.1,1.2
4	Enumerations, control structures, Arrays and Strings,	1.1,1.2
5	Class, Object, class and data abstraction, Access specifiers, separating interface from implementation.	1.1,1.2
6	Functions- Function, function prototype, accessing function and utility function, Constructors and destructor, Types of constructor	1.1,1.2
7	Objects and Memory requirements, Static members: variable and functions, inline function, friend function.	1.1,1.2

Question Bank: Theory & Programs
Mapped to Course Outcome:

Q. 1	Describe fundamentals of Object oriented Programming with examples.

Q. 2	What is the need of Object oriented Programming paragdim?

Q. 3	What are features of Object Oriented Programming

Q. 4	What are the different ways to define member functions of a class. What is the role of scope resolution operator in the definition of member function?

Q. 5	What is the need of passing objects as arguments. Discuss different ways to pass objects as arguments to a function.

Unit No.-II-

Name of the Unit : Inheritance and Pointers

Lecture No.	Details of the Topic to be covered	References
1	Inheritance- Base Class and derived Class, protected members, relationship between base Class and derived Class, Constructor and destructor in Derived Class	1.1,2.1
2	Overriding Member Functions, Class Hierarchies, Public and Private Inheritance, Types of Inheritance	1.1,2.1
3	Ambiguity in Multiple Inheritance, Virtual Base Class, Abstract class, Friend Class, Nested Class	1.1,2.1
4	Pointers: declaring and initializing pointers, indirection Operators, Memory Management: new and delete,	1.1,2.1
5	Pointers to Objects, this pointer, Pointers Vs Arrays, accessing Arrays using pointers,	1.1,2.1
6	Arrays of Pointers, Function pointers, Pointers to Pointers,	1.1,2.1
7	Pointers to Derived classes, Passing pointers to functions, Return pointers from functions, Null pointer, void pointer.	1.1,2.1

Question Bank: Theory & Programs

Mapped to Course Outcome:

Q. 1	Write a c++ program to initialize base class members through a derived class constructor.

Q. 2	Discuss with examples, the implications of deriving a class from an existing class by the ‘public’ and ‘protected’ access specifiers.

Q. 3	What is inheritance? How to inherit a base class as protected? Explain it in Multiple base classes?

Q. 4	What is the ambiguity that arises in multiple inheritance? How it can be overcome. Explain with example.

Unit No.-III

Name of the Unit: Polymorphism

Lecture No.	Details of the Topic to be covered	References
1	Polymorphism- Introduction to Polymorphism, Types of Polymorphism,	1.1,1.2
2	Operator Overloading- concept of overloading, operator overloading, Overloading Unary Operators, Overloading Binary Operators,	1.1,1.2
3	Data Conversion, Type casting (implicit and explicit), Pitfalls of Operator Overloading and Conversion,	1.1,1.2
4	Keywords explicit and mutable, Function overloading	1.1,1.2
5	Run Time Polymorphism- Pointers to Base class, virtual function and its significance in C++,	1.1,1.2
6	Pure virtual function and virtual table,	1.1,1.2
7	Virtual destructor, abstract base class.	1.1,1.2

Question Bank: Theory & Programs
Mapped to Course Outcome:

Q. 1	Is the need of virtual function? With an example, explain overriding of Member function of base in derived class?
Q. 2	What is a virtual table? .How does the implementing dynamic polymorphism. Explain with an example.
Q. 3	What is the virtual destructor?
Q. 4	What is the use of operator overloading? Write a program to overload post and pre increment operators.
Q. 5	What are the rules for overloading operators? Why friend f unction is required to overload binary operators?

Unit No.-IV

Name of the Unit: Files and Streams

Lecture No.	Details of the Topic to be covered	References
1	Data hierarchy, Stream and files, Stream Classes.	1.1, 1.2, 2.1,2.2
2	Stream Errors, Disk File I/O with Streams.	1.1, 1.2, 2.1,2.2
3	File Pointers, and Error Handling in File I/O.	1.1, 1.2, 2.1,2.2
4	File I/O with Member Functions.	1.1, 1.2, 2.1,2.2
5	Overloading the Extraction and Insertion Operators.	1.1, 1.2,2.1,2.2
6	Memory as a Stream Object, Command-Line Arguments, Printer output.	1.1, 1.2, 2.1,2.2
7	Program using Stream, Stream object, Command line arguments.	1.1, 1.2, 2.1,2.2

Question Bank: Theory & Programs
Mapped to Course Outcome:

Q. 1 Explain C++Stream Classes.

Q. 2 Explain the process of open,read,write and close files?

Q. 3 Write a C++ program involving working with a single file. Use ifstream and ofstream classes to write and read the information to and from a file using operators:- << and >>. Show how a file can be opened and closed

Q. 4 Write a C++ program involving input/output using overloaded operators << and >> and member functions of I/O stream classes.

Q. 5 Explain the role of seekg(),seekp(),tellg(),tellp(),function in the process of random access in a file.

Unit No.-V

Name of the Unit: Exception Handling and Templates

Lecture No.	Details of the Topic to be covered	References
1	Exception Handling- Fundamentals, other error handling techniques, simple exception handling- Divide by Zero,	2.1, 2.2
2	Multiple catching, re-throwing an exception, exception specifications, user defined exceptions, processing unexpected exceptions,	2.1, 2.2
3	Constructor, destructor and exception handling, exception and inheritance.	2.1, 2.2
4	Templates- The Power of Templates, Function template, overloading , , Function templates	2.1, 2.2
5	Class template, class template and Nontype parameters,	2.1, 2.2
6	Template and friends Generic Functions, The type name and export keywords.	2.1, 2.2

Question Bank: Theory & Programs
Mapped to Course Outcome:

Q. 1	Explain Function Template, Class Template.
Q. 2	Write down a detailed C++ program to demonstate the use of try, catch , throw and nested try.
Q. 3	What is a user defined exception. Write down the scenario where we require user defined exceptions.
Q. 4	When do we need multiple catch blocks for a single try block? Give an example.
Q. 5	Describe error handling techniques.

Unit No. VI

Name of the Unit: Standard Template Library (STL)

Lecture No.	Details of the Topic to be covered	References
1	Introduction to STL, STL Components,	2.1, 2.2
2	Containers- Sequence container and associative containers, container adapters	2.1, 2.2
3	Application of Container classes: vector, list	2.1, 2.2
4	**Algorithms**- basic searching and sorting algorithms, min-max algorithm	2.1, 2.2
5	Set operations, heap sort, **Iterators**- input, output, forward, bidirectional and random access.	2.1, 2.2
6	Object Oriented Programming – a road map to future	2.1, 2.2

Question Bank: Theory & Programs
Mapped to Course Outcome:

Q. 1 Explain the Standard Template Library and how it is working.

Q. 2 Why do we need STL when we have we can perform all the operations using user-defined data structures and functions?

Q. 3 What are Containers in STL? Explain them in detail.

Q. 4 What are Algorithms in STL? Explain them in detail.

Subject 4: Computer Graphics

Weekly Work Loads(in Hrs)	Lecture	Tutorial	Practical
	03	-	-

Online/ In-sem	Theory	Practica l	Oral	Term-work	Total Marks	Credit
30	70	0	0	0	100	03

Syllabus

Unit	Course Content	Hours
I	**Graphics Primitives and Scan Conversion Algorithms**	07
	Introduction, graphics primitives - pixel, resolution, aspect ratio, frame buffer. Display devices, applications of computer graphics. **Introduction to OpenGL** - OpenGL architecture, primitives and attributes, simple modelling and rendering of two- and three-dimensional geometric objects, GLUT, interaction, events and call-backs picking. (**Simple Interaction with the Mouse and Keyboard)** **Scan conversion:** Line drawing algorithms: Digital Differential Analyzer (DDA), Bresenham. Circle drawing algorithms: DDA, Bresenham, and Midpoint.	
II	**Polygon, Windowing and Clipping**	07
	Polygons: Introduction to polygon, types: convex, concave and complex. Inside test. **Polygon Filling:** flood fill, seed fill, scan line fill. **Windowing and clipping:** viewing transformations, 2-D clipping: Cohen – Sutherland algorithm line Clipping algorithm, Sutherland Hodgeman Polygon clipping algorithm, Weiler Atherton Polygon Clipping algorithm.	

III	**2D, 3D Transformations and Projections**	07
	2-D transformations: introduction, homogeneous coordinates, 2-D transformations - Translation, scaling, rotation and shear, rotation about an arbitrary point. **3-D transformations:** introduction, 3-D transformations - Translation, scaling, rotation and shear, rotation about an arbitrary axis. **Projections :** Parallel (Oblique: Cavalier, Cabinet and orthographic: isometric, diametric, trimetric) and Perspective (Vanishing Points – 1 point, 2 point and 3 point)	
IV	**Light, Colour, Shading and Hidden Surfaces**	07
	Colour models: Properties of Light, CIE chromaticity Diagram, RGB, HSV, CMY. **Illumination Models**: Ambient Light, Diffuse reflection, Specular Reflection, and the Phong model, Combined diffuse and Specular reflections with multiple light sources, warn model, **Shading Algorithms**: Halftone, Gauraud and Phong Shading. **Hidden Surfaces** Introduction, Back face detection and removal, Algorithms: Depth buffer (z), Depth sorts (Painter), Area subdivision (Warnock)	
V	**Curves and Fractals**	07
	Curves: Introduction, Interpolation and Approximation, Blending function, B-Spline curve, Bezier curve, **Fractals**: Introduction, Classification, Fractal generation: snowflake, Triadic curve, Hilbert curve, Applications.	
VI	**Introduction to Animation and Gaming**	07
	Segment: Introduction, Segment table, Segment creation, closing, deleting and renaming, Visibility. **Animation:** Introduction, Conventional and computer based animation, Design of animation sequences, Animation languages, Key- frame, Morphing, Motion specification. **Gaming:** Introduction, Gaming platform (NVIDIA, i8060), Advances in Gaming.	

Teaching Plan

Sr. No.	Unit	Topics to be covered	Book Referred	Total Lecture Planned
1	I	Introduction, graphics primitives - pixel, resolution, aspect ratio, frame buffer. Display devices, applications of computer graphics. Introduction to OpenGL - OpenGL architecture, primitives and attributes, simple modelling and rendering of two- and three-dimensional geometric objects, GLUT, interaction, events and call-backs picking. (Simple Interaction with the Mouse and Keyboard) Scan conversion: Line drawing algorithms: Digital Differential Analyzer (DDA), Bresenham. Circle drawing algorithms: DDA, Bresenham, and Midpoint.	1.1,1.2	8
2	II	Polygons: Introduction to polygon, types: convex, concave and complex. Inside test. Polygon Filling: flood fill, seed fill, scan line fill. Windowing and clipping: viewing transformations, 2-D clipping: Cohen – Sutherland algorithm line Clipping algorithm, Sutherland Hodgeman Polygon clipping algorithm, Weiler Atherton Polygon Clipping algorithm.	1.1,2.1	7
3	III	2-D transformations: introduction, homogeneous coordinates, 2-D transformations - Translation, scaling, rotation and shear, rotation about an arbitrary point. 3-D transformations: introduction, 3-D transformations - Translation, scaling, rotation and shear, rotation about an arbitrary axis. Projections : Parallel (Oblique: Cavalier, Cabinet and orthographic: isometric, diametric, trimetric) and Perspective (Vanishing Points – 1 point, 2 point and 3 point)	1.1,1.2,1.3	7
4	IV	Colour models: Properties of Light, CIE chromaticity Diagram, RGB, HSV, CMY. Illumination Models: Ambient Light, Diffuse reflection, Specular Reflection, and the Phong model, Combined diffuse and Specular reflections with multiple light sources, warn model, Shading Algorithms: Halftone, Gauraud	1.1, 1.2, 1.3, 2.1,2.2	7

		and Phong Shading. Hidden Surfaces Introduction, Back face detection and removal, Algorithms: Depth buffer (z), Depth sorts (Painter), Area subdivision (Warnock)		
5	V	Curves: Introduction, Interpolation and Approximation, Blending function, B-Spline curve, Bezier curve, Fractals: Introduction, Classification, Fractal generation: snowflake, Triadic curve, Hilbert curve, Applications.	2.1, 2.2	6
6	VI	Segment: Introduction, Segment table, Segment creation, closing, deleting and renaming, Visibility. Animation: Introduction, Conventional and computer based animation, Design of animation sequences, Animation languages, Key- frame, Morphing, Motion specification. Gaming: Introduction, Gaming platform (NVIDIA, i8060), Advances in Gaming.	2.1, 2.2	6

1. Text Books:

1. S. Harrington, “Computer Graphics”‖, 2nd Edition, McGraw-Hill Publications, 1987, ISBN 0 07 – 100472 – 6.

2. Donald D. Hearn and Baker, “Computer Graphics with OpenGL”, 4th Edition, ISBN-13: 9780136053583.

3. D. Rogers, “Procedural Elements for Computer Graphics”, 2nd Edition, Tata McGraw-Hill Publication, 2001, ISBN 0 – 07 – 047371 – 4.

2. Reference Books:

1. J. Foley, V. Dam, S. Feiner, J. Hughes, “Computer Graphics Principles and Practice”‖, 2nd Edition, Pearson Education, 2003, ISBN 81 – 7808 – 038 – 9.

2. D. Rogers, J. Adams, “Mathematical Elements for Computer Graphics”‖, 2nd Edition, Tata McGraw Hill Publication, 2002, ISBN 0 – 07 – 048677 – 8.

Reference Web Links/ Research Paper/ Referred Book other than Mention in Syllabus:

- *e-Books:*

https://open.umn.edu/opentextbooks/textbooks/introduction-to-computer-graphics

http://www2.cs.uidaho.edu/~jeffery/courses/324/lecture.html

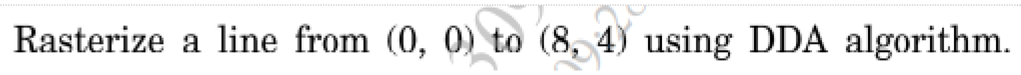
- *MOOC/ Video Lectures available at:*
- *https://nptel.ac.in/courses/106/106/106106090/*
- *https://nptel.ac.in/courses/106/102/106102065/*

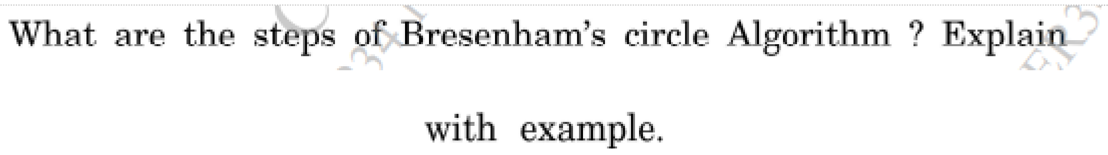
Unit No.-I-

Name of the Unit: Graphics Primitives and Scan Conversion Algorithms

Lecture No.	Details of the Topic to be covered	References
1	Introduction, graphics primitives - pixel, resolution, aspect ratio, frame buffer. Display devices, applications of computer graphics.	1.1,1.2
2	Introduction to OpenGL - OpenGL architecture, primitives and attributes, simple modelling and rendering of two- and three-dimensional geometric objects, GLUT, interaction, events and call-backs picking. (Simple Interaction with the Mouse and Keyboard)	1.1,1.2
3	Scan conversion: Digital Differential Analyzer (DDA), and numerical	1.1,1.2
4	Line drawing algorithms: Bresenham.	1.1,1.2
5	Line drawing algorithms: numerical on Bresenham	1.1,1.2
6	Circle drawing algorithms: DDA	1.1,1.2
7	Circle drawing algorithms: Bresenham	1.1,1.2
8	Circle drawing algorithms: Midpoint.	1.1,1.2

Question Bank: Theory & Numerical
Mapped to Course Outcome:

Q. 1	Explain Bresenham algorithm for line drawing. Write advantages and disadvantages of it over DDA line drawing algorithm.
Q. 2	Write Bresenham circle drawing algorithm. Also explain mathematical foundation of it
Q. 3	 Rasterize a line from (0, 0) to (8, 4) using DDA algorithm.

Q. 4	 What are the steps of Bresenham's circle Algorithm ? Explain with example.

Q. 5	Derive equation for decision parameter of Bresenham's circle algorithm.

Unit No.-II-

Name of the Unit : Polygon, Windowing and Clipping

Lecture No.	Details of the Topic to be covered	References
1	Polygons: Introduction to polygon, Polygon types: convex, concave and complex, Inside test.	1.1,2.1
2	Polygon Filling , : flood fill,	1.1,2.1
3	Polygon Filling seed fill, scan line fill.	1.1,2.1
4	Polygons: Introduction to polygon Windowing and clipping: viewing transformations,	1.1,2.1
5	2-D clipping: Cohen – Sutherland algorithm line Clipping algorithm	1.1,2.1
6	Sutherland Hodgeman Polygon clipping algorithm	1.1,2.1
7	Weiler Atherton Polygon Clipping algorithm.	1.1,2.1

Question Bank: Theory & Numerical Mapped to Course Outcome:

Q. 1	Explain Sutherland-Hodgeman algorithm for polygon clipping. Compare it Cohen-Sutherland Clipping.

Q. 2	 (*b*) Explain Cohen Sutherland line clipping method with suitab example. [

Q. 3	Write pseudo code for Boundary fill algorithm. Compare boundary fill algorithm with scan line algorithm.

Q. 4	Explain Midpoint subdivision line clipping method with suitable example.

Unit No.-III

Name of the Unit

2D, 3D Transformations and Projections

Lecture No.	Details of the Topic to be covered	References
1	2-D transformations: introduction, homogeneous coordinates,	1.1,1.2,1.3
2	2-D transformations – Translation , scaling, rotation and shear.	1.1,1.2,1.3
3	Rotation about an arbitrary point	1.1,1.2,1.3
4	3-D transformations: introduction, Translation, scaling, rotation and shear	1.1,1.2,1.3
5	Rotation about an arbitrary axis.	1.1,1.2,1.3
6	Projections : Parallel (Oblique: Cavalier, Cabinet and orthographic: isometric, diametric, trimetric)	1.1,1.2,1.3
7	Perspective (Vanishing Points – 1 point, 2 point and 3 point)	1.1,1.2,1.3

Question Bank: Theory & Numerical
Mapped to Course Outcome:

Q. 1	For origin centered unit square, rotate 45º clockwise, scale by a factor 2 in x–direction. Find resultant coordinates of square (write required matrices).

Q. 2	Explain rotation about an arbitrary point in 2D

Q. 3	Explain 3D viewing transformation.

Q. 4	Explain basic transformations on 3D.

Q. 5

Explain 3D reflection about xy, yz and xz plane.

Unit No.-IV

Name of the Unit

Light, Colour, Shading and Hidden Surfaces

Lecture No.	Details of the Topic to be covered	References
1	Colour models: Properties of Light, CIE chromaticity Diagram	1.1, 1.2, 1.3, 2.1,2.2
2	RGB, HSV, CMY	1.1, 1.2, 1.3, 2.1,2.2
3	Illumination Models: Ambient Light, Diffuse reflection, Specular Reflection, and the Phong model,	1.1, 1.2, 1.3, 2.1,2.2
4	Combined diffuse and Specular reflections with multiple light sources, Warn model	1.1, 1.2, 1.3, 2.1,2.2
5	Shading Algorithms, Halftone, Gauraud and Phong Shading.	1.1, 1.2, 1.3, 2.1,2.2
6	Hidden Surfaces Introduction, Back face detection and removal .	1.1, 1.2, 1.3, 2.1,2.2
7	Algorithms: Depth buffer (z), Depth sorts (Painter), Area subdivision (Warnock) .	1.1, 1.2, 1.3, 2.1,2.2

Question Bank: Theory & Numerical
Mapped to Course Outcome:

Q. 1	Explain Back face Detection and removal.

Q. 2	Explain and compare point source and diffuse illumination

Q. 3	Explain Phong Shading Algorithm

Q. 4	Explain Gaurad Shading algorithm.

Q. 5	Write a note on Phong Reflection Model.

Unit No.-V

Name of the Unit

Curves and Fractals

Lecture No.	Details of the Topic to be covered	References
1	Curves: Introduction, Interpolation and Approximation	2.1, 2.2
2	Blending function, B-Spline curve, Bezier curve,	2.1, 2.2
3	Fractals: Introduction, Classification	2.1, 2.2
4	Fractal generation: snowflake	2.1, 2.2
5	Triadic curve	2.1, 2.2
6	Hilbert curve, Applications	2.1, 2.2

Question Bank: Theory & Numerical Mapped to Course Outcome:

Q. 1	Explain blending function for B-spline curve

Q. 2	Explain Hilbert curve with example

Q. 3	Explain Koch curve with example.

Q. 4	Why cubic Bezier curves are chosen ? Explain any Bezier curve generation method

Q. 5	Define fractals with examples. Give various categories in which fractals are classified.

Unit No.-VI-

Name of the Unit

Introduction to Animation and Gaming

Lecture No.	Details of the Topic to be covered	References
1	Segment: Introduction, Segment table, Segment creation, closing, deleting and renaming, Visibility.	2.1, 2.2
2	Introduction, Conventional and computer based animation, Design of animation sequences	2.1, 2.2
3	Animation languages, Key- frame, Morphing, Motion specification	2.1, 2.2
4	Gaming: Introduction	2.1, 2.2
5	Gaming platform (NVIDIA, i8060)	2.1, 2.2
6	Advances in Gaming	2.1, 2.2

Question Bank: Theory & Numerical
Mapped to Course Outcome:

Q. 1	Write any four important features of NVIDIA gaming platform. Explain need of NVIDIA workstation in gaming

Q. 2	What is animation ? Explain the basic rules required for Animation.

Q. 3	Draw and explain block diagram of i860 microprocessor.

Q. 4	What are the steps in design in animation sequence ? Describe about each step briefly. [7]

Subject 5: Operating System

Weekly Work Loads(in Hrs)	Lecture	Tutorial	Practical
	03	-	-

Online/ In-sem	Theory	Practical	Oral	Term-work	Total Marks	Credit
30	70	0	0	0	100	03

Unit	Course Content	Hours
I	**Fundamental Concepts of Operating system**	06
	Operating system functions and characteristics, historical evolution of operating systems, issues in operating system design, User's view of the OS, Types of OS: Batch, time sharing, multiprogramming, distributed, network and real-time systems, Operating-System Services, Types of System Calls, System Programs. BASH Shell scripting: Basic shell commands, shell as a scripting language.	
II	**Process Management**	06
	Concept of Sequential Organization, Overview of Array, Array as an Abstract Data Type, Operations on Array, Merging of two arrays, Storage Representation and their Address Calculation: Row major and Column Major, Multidimensional Arrays: Two-dimensional arrays, n-dimensional arrays. Concept of Ordered List, **Single Variable Polynomial**: Representation using arrays, Polynomial as array of structure, Polynomial addition, Polynomial multiplication. **Sparse Matrix:** Sparse matrix representation using array, Sparse matrix addition, Transpose of sparse matrix- Simple and Fast Transpose, Time and Space tradeoff.	
III	**Process Coordination**	08
	Searching: Search Techniques-Sequential Search/Linear Search, Variant of Sequential Search- Sentinel Search, Binary Search, Fibonacci Search, and Indexed Sequential Search. **Sorting:** Types of Sorting-Internal and External Sorting, General Sort Concepts-Sort Order, Stability, Efficiency, and Number of Passes,	

	Comparison Based Sorting Methods-Bubble Sort, Insertion Sort, Selection Sort, Quick Sort, Shell Sort, Non-comparison Based Sorting Methods-Radix Sort, Counting Sort, and Bucket Sort, Comparison of All Sorting Methods and their complexities	
IV	**Memory Management**	06
	Introduction to Static and Dynamic Memory Allocation, **Linked List:** Introduction, of Linked Lists, Realization of linked list using dynamic memory management,operations, Linked List as ADT, **Types of Linked List:** singly linked, linear and Circular Linked Lists, Doubly Linked List, Doubly Circular Linked List, Primitive Operations on Linked List-Create, Traverse, Search, Insert, Delete, Sort, Concatenate. Polynomial Manipulations- Polynomial addition. Generalized Linked List (GLL) concept, Representation of Polynomial using GLL.	
V	**I/O and File Management**	06
	Introduction, properties of trees, Binary search tree, tree traversal, decision tree, prefix codes and Huffman coding, cut sets, Spanning Trees and Minimum Spanning Tree, Kruskal‘s and Prim‘s algorithms, The Max flow- Min Cut Theorem (Transport network).	
VI	**Linux**	06
	The structure of algebra, Algebraic Systems, Semi Groups, Monoids, Groups, Homomorphism and Normal Subgroups, and Congruence relations, Rings, Integral Domains and Fields, Coding theory, Polynomial Rings and polynomial Codes, Galois Theory –Field Theory and Group Theory.	

Teaching Plan

Sr. No.	Unit	Topics to be covered	Book Referred	Total Lecture Planned
1	I	Operating system functions and characteristics, historical evolution of operating systems, issues in operating system design, User's view of the OS, Types of OS: Batch, time sharing, multiprogramming, distributed, network and real-time systems, Operating-System Services, Types of System Calls, System Programs. BASH Shell scripting: Basic shell commands, shell as a scripting language.	T1,T2,T3	06
2	II	Process concept, Process Control Block (PCB), Process Operations, **Process Scheduling**: Types of process schedulers, Types of scheduling: Preemptive, Non preemptive. Scheduling algorithms: FCFS, SJF, RR, Priority, Inter process Communication (IPC). **Threads**: multithreaded model, implicit threads, threading issues	T1,T2	06
3	III	**Synchronization**: Principles of Concurrency, Requirements for Mutual Exclusion, Mutual Exclusion: Hardware Support, Operating System Support (Semaphores and Mutex), Programming Language Support (Monitors). Classical synchronization problems : Readers/Writers Problem, Producer and Consumer problem, Inter-process communication (Pipes, shared memory: system V) **Deadlock**: Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock	T1,T2	08
4	IV	Memory Management: Memory Management Requirements, Memory Partitioning: Fixed Partitioning, Dynamic Partitioning, Buddy System, Relocation, Paging, Segmentation. Virtual Memory: Hardware and Control Structures, Operating System	T2	06

		Software		
5	V	I/O Management: I/O Devices, Organization of I/O function, I/O Buffering, Disk Scheduling Disk Scheduling policies like FIFO, LIFO, STTF, SCAN, C-SCAN. File Management: Concept, Access methods, Directory Structure, Protection, File System implementation, Directory Implementation, Allocation methods, Free Space management.	T1,T2	06
6	VI	History Of Unix and Linux, Overview Of Linux - Linux Goals, Interfaces to Linux, The Shell, Linux Utility Programs, Kernel structure, Processes in Linux –Process management system calls in Linux, Implementation of process and threads in Linux, Process scheduling Linux, Booting	T3	06

Text Books:

1. Abraham Silberschatz, Peter Baer Galvinand Greg Gagne, Operating System Concepts, WILEY, ISBN978-1-118-06333-0, 9th Edition
2. William Stallings, Operating System: Internals and Design Principles, Prentice Hall, ISBN-10: 0-13-380591-3, ISBN-13: 978-0-13-380591-8, 8th Edition
3. Andrew S. Tanenbaum & Herbert Bos, Modern Operating System, Pearson, ISBN-13: 9780133592221, 4th Edition

Reference Books:

1. Tom Adelstein and Bill Lubanovic, Linux System Administration, O’Reilly Media, ISBN-10: 0596009526, ISBN-13: 978-0596009526
2. Harvey M. Deitel, Operating Systems, Prentice Hall, ISBN-10: 0131828274, ISBN-13: 978-0131828278
3. Thomas W. Doeppner, Operating System in depth: Design and Programming, WILEY, ISBN: 978-0-471-68723-8
4. Mendel Cooper, Advanced Shell Scripting, Linux Documentation Project

Reference Web Links/ Research Paper/ Referred Book other than Mention in Syllabus:

- https://www.javatpoint.com/operating-system

- https://youtube.com/playlist?list=PLsylUObW5M3CAGT6OdubyH6FztKfJCcFB&si=Cl2pC9Z4CblII504
- https://www.geeksforgeeks.org/operating-systems/