**Kolhapur Institute of Technology's
College of Engineering (Autonomous),
Kolhapur**

Department of CSE (AIML)

Syllabus for S. Y B. Tech.

Computer Science and Engineering (AIML)
Scheme: 2022-23

SEM III

Semester III								
Sr. No.	Course Code	Course Name	L	T	P	Hrs. / Week	Credits	Category
1	UAMC0301	Linear Algebra	3	1		4	4	BS
2	UAMC0302	Discrete Mathematics	3	1		4	4	ES
3	UAMC0303	Data Structures	3			3	3	PC
4	UAMC0304	Principles of AI & ML	3			3	3	PC
5	UAMC0305	Object Oriented Programming	3			3	3	PC
6	UAMC0306	Software Engineering	3			3	3	ES
7	UAMC0331	Data Structures Lab			2	2	1	PC
8	UAMC0332	Object Oriented Programming Lab			2	2	1	PC
9	UAMA0361	Indian Constitution	2			2		MAC
**								
		Total:				26	22	

Title of the Course: Linear Algebra Course Code: UAMC0301	L	T	P	Credits
	3	1	---	4

Course Pre-Requisite: Basics of Matrix Algebra, Vectors and Set Theory.

Course Description: This Course contains Linear Algebra concepts for AI and DS.

Course Objectives:

1. To learn mathematical methodologies and models in linear algebra.
2. To develop mathematical skills and enhance logical thinking power of students.
3. To provide students with skills in linear algebra which would enable them to devise engineering solutions for given situations they may encounter in their profession.
4. To increase interest towards the use of mathematics in engineering module.

Course Outcomes:

COs	After the completion of the course the student will be able to	Bloom's Cognitive	
		level	Descriptor
CO1	**Demonstrate** the basic mathematical concepts in AI and data science related to linear algebra and Statistics	2	Demonstrate
CO2	**Explain** Vector Space concepts in dealing with problems in AI and data science.	2	Explain
CO3	**Apply** the knowledge of Statistics to **solve** problems arising in AI and data science.	3	Apply
CO4	**Apply** the knowledge of Linear transformation to **solve** problems arising in AI and data science.	3	Apply

CO-PO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO 12	PS O1	PS O2
CO1	-	2	-	-	-	-	-	-	-	-	-	2	2	2
CO2	3	-	-	-	-	-	-	-	-	-	-	2		
CO3	-	-	3	-	-	-	-	-	-	-	-	2	2	2
CO4	-	-	3	-	-	-	-	-	-	-	-	2	2	

Assessment Scheme:

Two components of In Semester Evaluation (ISE), One Mid Semester Examination (MSE) and one End Semester Examination (ESE) having 20%, 30% and 50% weightage respectively.

Assessment Component	Marks
ISE 1	10
MSE	30
ISE 2	10
ESE	50

ISE 1 and **ISE 2** are based on Assignment / Declared test / Quiz / Seminar / Group discussions / presentation, etc.

MSE is based on 50% of course content (first three units). **ESE** is based on 100% course content with 60-70% weightage for course content (last three units) covered after MSE.	
Course Contents:	
Unit 1: Matrices: **1.1** Matrices and Computation of Eigen value by power method, **1.2** Iterative Method for Ax = b : Gauss Jacobi , Gauss-Siedel Method, **1.3** LU Decomposition, **1.4** Computation with Matrices - Matrix Norms, Condition Numbers, **1.5** Inner and outer products, **1.6** Idea about sparse and dense matrix, symmetric matrix, **1.7** Hermitian, skew-Hermitian and unitary matrices.	7 **Hrs.**
Unit 2 :Vector Algebra: 2.1 Vector Spaces, 2.2 Subspaces, basis, span, 2.3 Linear Independence, Basis and Dimension, 2.4 Four Fundamental Subspaces 2.5 Orthogonality - Orthogonal Vectors and Subspaces, 2.6 Cosines and Projections onto Lines 2.7 Orthogonal Bases and Gram – Schmidt	7 **Hrs.**
Unit 3 :Linear Algebra-I: 3.1 Eigen value Problems: Overview of eigenvalue problems – Diagonalization of a Matrix, 3.2 Difference Equations and Powers, Differential Equations 3.3 Complex Matrices, Similarity Transformations, Positive Definite Matrices - Minima, Maxima 3.4 Saddle Points, Tests for Positive Definiteness 3.5 SVD, Minimum Principles, 3.6 Finite Element Method.	7 **Hrs.**
Unit 4: Linear Algebra-II: 4.1 Linear Transformations Definition and example of linear transformation, 4.2 Null space, range, rank and nullity of linear transformation, 4.3 matrix representation of a linear transformation, dual space, dual basis, double dual, 4.4 Composition of linear transformation and matrix multiplication. 4.5 Transformation Diagonalization : Diagonalizability, 4.6 matrix Limits and Markov Chains	8 **Hrs.**

Unit 5: Exploratory Data Analysis: 5.1 Elements of Structured Data, Rectangular Data, 5.2 Mean , Median, 5.3 Standard Deviation, Percentiles and Boxplots, Mode, 5.4 Expected Value. 5.5 Inference from conditional fuzzy propositions.	8 **Hrs.**
Unit 6: Applications: 6.1 Markov matrices and Economics Model. 6.2 Linear Programming and 6.3 Computer Graphics	5 **Hrs.**

Text Books:

1. Linear Algebra and Its Applications - by Gilbert Strang, 4th Edition, Thomson Brooks/Cole.

2. Stephen H. Friedberg, Arnold J. Insel, Lawrence E. Spence, Linear Algebra, 4th Ed., PrenticeHall of India Pvt. Ltd., New Delhi, 2004.

Reference Books:

1. Numerical Linear Algebra, Allaire, Grégoire, Kaber, Sidi Mahmoud, Springer (2008)
2. Applied Numerical Linear Algebra, by James W. Demmel, SIAM (1997)
3. Numerical Linear Algebra, by Lloyd Trefethen and David Bau III, SIAM, 1997. [Lectures 1-29, 32-35 covered in chapter 1-6 of the Text Book]
4. David C. Lay, Linear Algebra and its Applications, 3rd Ed., Pearson Education Asia, Indian Reprint, 2007.
5. S. Lang, Introduction to Linear Algebra, 2nd Ed., Springer, 2005. 4. Gilbert Strang, Linear Algebra and its Applications, Thomson, 2007.

Title of the Course: Discrete Mathematics	L	T	P	Credits
Course Code: UAMC0302	3	1	--	4

Course Pre-Requisite: Basic Mathematics

Course Description: This course focuses on concepts of Discrete Mathematical Structures such as Set Theory & Relations, Mathematical Logic, Algebraic systems, Lattices, Graphs, Counting Theory Principles etc.

Course Objectives:

1. To use mathematically correct terminology and notations.
2. To understand and critically analyze, formulate and solve the mathematical problems and proofs
3. To understand the concepts of Discrete Mathematics such as Sets, Algebraic Systems, Graphs, Groups and lattices
4. To design and implement experiments on Discrete Structures – Truth tables of statement formula, Set Operations, tree traversal techniques.

Course Outcomes:

COs	After the completion of the course the student will be able to	Bloom's Cognitive	
		level	Descriptor
CO1	**Explain** the discrete mathematical structures such as Sets, Algebraic systems, Groups, Probability in the field of Computer Science.	2	Explain
CO2	**Illustrate** the problems related to the topics on discrete mathematics Computer Science.	2	Illustrate
CO3	**Make** use of discrete mathematical terminology and concepts of counting theory in different areas of Computer Science.	3	Make
CO4	**Apply** the functions and algorithms related to Discrete structures.	3	Apply

CO-PO Mapping:

CO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO 12**	**PS O1**	**PS O2**
CO1	**3**													
CO2		**3**												**1**
CO3		**2**											**1**	
CO4			**2**											**1**

Assessment Scheme:

Two components of In Semester Evaluation (ISE), One Mid Semester Examination (MSE) and one End Semester Examination (ESE) having 20%, 30% and 50% weightage respectively.

Assessment Component	**Marks**
ISE 1	10
MSE	30
ISE 2	10
ESE	50

ISE 1 and **ISE 2** are based on Assignment / Declared test / Quiz / Seminar / Group discussions / presentation, etc.
MSE is based on 50% of course content (first three units).
ESE is based on 100% course content with 60-70% weightage for course content (last three units) covered after MSE.

Course Contents:	
Unit 1: Mathematical logic Statements and Notations, Connectives – negation, Conjunction, disjunction, conditional, bi-conditional, Statement formulas and truth tables, well-formed formulas, Tautologies, Equivalence of formulas, Duality law, Tautological implications, functionally complete sets of connectives, other connectives, Normal and principal normal forms, completely parenthesized infix and polish notations, Theory of Inference for statement calculus – validity using truth table, rules of inference, consistency of Premises and indirect method of proof.	**08 Hrs.**
Unit 2 : Set Theory : Basic concepts of set theory, Operations on sets, Ordered pairs, Cartesian Products, Representation of discrete structures, Relation and ordering - properties of binary relations in a set, Relation matrix and the graph of a relation, Partition and Covering of set, Equivalence relations, Composition of Binary relations, Partial ordering , POSET and Hasse diagram. Functions – types, composition of functions, Inverse functions.	**08 Hrs.**
Unit 3 : Algebraic systems: Algebraic systems, properties and examples ,Semigroups and Monoids, properties and examples, Homomorphism of Semigroups and Monoids , Groups: Definition and examples, Subgroups and homomorphism.	**05 Hrs.**
Unit 4: Lattices and Boolean algebra: Lattice as POSETs , definition , examples and properties ,Lattice as algebraic systems, Special lattices, Boolean algebra definition and examples, Boolean functions	**05 Hrs.**
Unit 5: Permutations, Combinations: The Basics of Counting, The Pigeonhole Principle, Permutations and Combinations, Generalized Permutations and Combinations	**05 Hrs**
Unit 6: Graph Theory: Basic concepts of Graph Theory, Storage Representation and Manipulation of Graphs, Eulerian and Hamiltonian Graphs, Graph Colouring-chromatic, Trees-Definitions, Examples and Properties, PERT & Related Technologies	**09 Hrs**

Text Books:

1. Discrete Mathematical Structures with Application to Computer Science- J. P. Tremblay & R. Manohar (MGH International).

2. Discrete Mathematics and its Applications- Kenneth H. Rosen (AT & T Bell Labs)

Reference Books:

1. Discrete Mathematics- Semyour Lipschutz, MarcLipson (MGH)- Schaum’s Outlines

2. C.L. Liu and D. Mohapatra, “Elements of Discrete Mathematics”- SiE Edition, TMGH,2008,ISBN10:07-066913-9

Title of the Course: Data Structures	**L**	**T**	**P**	**Credits**
Course Code: UAMC0303	**3**	**--**	**--**	**3**

Course Pre-Requisite: C/Python Programming Language

Course Description: In this course students will learn Data Structure concepts, Data Structure implementation and various searching and sorting algorithms

Course Objectives:

1. To learn need of Data Structure.
2. To become familiar with advanced data structures such as Stacks, Queues, Trees etc.
3. To analyze and solve problems using advanced data structures such as Lists, Linked Lists, Queues, Stacks, Trees, and Graphs.
4. To write programs on Linked Lists, Doubly Linked Lists, Trees etc.

Course Outcomes:

COs	After the completion of the course the student will be able to	Bloom's Cognitive	
		level	Descriptor
CO1	**Define** the basic terms of Linear Lists, Linked List, Doubly Linked List, Non Linear Data Structures - Binary Trees, AVL Trees, Graphs	1	Define
CO2	**Choose** the appropriate and optimal data structure for a specified application	2	Choose
CO3	**Analyze** Time Complexity and Memory Complexity of different algorithms	4	Analyze
CO4	**Write** programs and applications with Static and Dynamic data structures	1	Write

CO-PO Mapping:

CO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO 12**	**PS O1**	**PS O2**
CO1	3												2	2
CO2	3												2	2
CO3		2											1	1
CO4			2										3	3

Assessment Scheme:

Two components of In Semester Evaluation (ISE), One Mid Semester Examination (MSE) and one End Semester Examination (ESE) having 20%, 30% and 50% weightage respectively.

Assessment Component	**Marks**
ISE 1	10
MSE	30
ISE 2	10
ESE	50

ISE 1 and **ISE 2** are based on Assignment / Declared test / Quiz / Seminar / Group discussions / presentation, etc.

MSE is based on 50% of course content (first three units).

ESE is based on 100% course content with 60-70% weightage for course content (last three units) covered after MSE.	
Course Contents:	
Unit 1: Basics of Data Structure Abstract Data Type (ADT), control structure, array, function, structure, pointer, Algorithm, Space and Time Complexity, Recursion, Towers of Hanoi and Ackermann’s function	**6 Hrs**
Unit 2: Stacks and Queues Stack: Definition, representation, implementation, applications of stack for expression evaluation and conversion Queue: Definition, representation, implementation, applications of queue, circular queue and priority queue	**7 Hrs**
Unit 3: Linked Lists Definition, representation, implementation and operations on singly, doubly and circular linked lists, stack and queue implementation using linked list Hashing: Hashing functions, overflow handling, open and closed hashing, rehashing	**9 Hrs**
Unit 4: Searching and Sorting Techniques Search: Importance of searching, types- sequential search, binary search Sort: Different types: bubble sort, selection sort, insertion sort, merge sort, quick sort, radix sort, heap sort	**6 Hrs**
Unit 5: Trees Basic terminology, binary tree and its representation, binary tree traversal methods, binary search tree, AVL tree, B tree, B+ tree, Heaps and its operations	**7 Hrs**
Unit 6: Graphs Basic terminology and representation of graphs using adjacency matrix, storage representation, graph traversal techniques- Breadth First, Depth First	**5 Hrs**
Text Books: 1. Data Structure using C- A. M. Tanenbaum, Y. Langsam, M. J. Augenstein (PHI) 2. Data Structures- A Pseudo code Approach with C – Richard F. Gilberg and Behrouz A. Forouzon, Cengage Learning, Second Edition. 3. Schaum’s Outlines Data Structures – Seymour Lipschutz (MGH), Tata McGraw-Hill. **Reference Books:** 1. Fundamentals of Data Structures – Horowitz, Sahni CBS India 2. An introduction to data structures with Applications- Jean-Paul Tremblay, Paul. G. Soresan, Tata Mc-Graw Hill International Editions, Second Edition.	

Title of the Course: Principles of Artificial Intelligence and Machine learning **Course Code: UAMC0304**	**L**	**T**	**P**	**Credits**
	3	**--**	**--**	**3**

Course Pre-Requisite: Computer Science concepts

Course Description: This course is about to understand Principles of Artificial Intelligence and machine learning

Course Objectives:

1. To understand the main approaches to Artificial intelligence, Machine learning and Data Science
2. To Explore areas of application based on knowledge representation
3. To develop abilities to apply, build and modify decision models to solve real problems.
4. To Familiarize the Artificial Intelligence, Machine learning and Data Science techniques for building well-engineered and efficient intelligent systems.

Course Outcomes:

COs	After the completion of the course the student will be able to	Bloom's Cognitive	
		level	Descriptor
CO1	Understand the importance and overview of Artificial intelligence and agents strategies, Machine learning techniques with its types in the problems.	1	Understand
CO2	Explain the artificial intelligence, Machine learning knowledge representation, problem solving and learning methods of artificial intelligence.	2	Explain
CO3	Develop abilities to apply, build and modify decision models to solve real problems.	4	Develop
CO4	Familiarize the Artificial Intelligence, Machine learning techniques for building well-engineered and efficient intelligent systems.	3	Familiarize

CO-PO Mapping:

CO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO 12**	**PS O1**	**PS O2**
CO1	**2**										**1**			
CO2		**2**												**1**
CO3			**3**									**1**	**1**	
CO4			**3**								**1**		**1**	**2**
CO5													**1**	**1**

Assessment Scheme:

Two components of In Semester Evaluation (ISE), One Mid Semester Examination (MSE) and one End Semester Examination (ESE) having 20%, 30% and 50% weightage respectively.

Assessment Component	**Marks**
ISE 1	10
MSE	30
ISE 2	10
ESE	50

ISE 1 and **ISE 2** are based on Assignment / Declared test / Quiz / Seminar / Group discussions / presentation, etc.
MSE is based on 50% of course content (first three units).
ESE is based on 100% course content with 60-70% weightage for course content (last three units) covered after MSE.

Course Contents:

Unit 1: Overview of Artificial Intelligence and Agents: Introduction to AI, Types of AI, Intelligent Agents, Agents & environment, nature of environment, structure of agents, goal- based agents, utility-based agents, learning agents.	**4 Hrs**
Unit 2: Problem Solving: Defining the problem as state space search, production system, problem characteristics and issues in the design of search programs. Problem solving agents, searching for solutions.	**5 Hrs**
Unit 3: Search techniques: Informed Search & **Uninformed** search strategies: breadth first search, depth first search, depth limited search, bidirectional search. Heuristic search strategies	**5 Hrs**
Unit 4: Machine Learning Essentials: What is machine learning?, How does machine learning work? Types of machine learning: Supervised , Unsupervised , and reinforcement learning algorithms, ,Linear regression, Logistic Regression, Probability, Dummy variables	**5 Hrs**
Unit 5: Overview of Machine learning Algorithms : Supervised machine learning algorithms, Naïve Bayes classification, Decision trees, Unsupervised learning, K-means clustering, Ensemble techniques, Neural networks, Natural Language Processing	**5 Hrs**
Unit 6: Communicating Data: Identifying effective and ineffective visualizations: Scatter plots, Line graphs, Bar charts, Histograms, Box plots. Graphs and Statistics lie: Correlation versus causation, why/how/what strategy of presenting.	**5 Hrs**

Text Books:

1. Kevin Night and Elaine Rich, Nair B ,“Artificial Intelligence(SIE)”,Mc Graw Hill- 2008.
2. DanW.Patterson,“Introduction to AI and ES”, Pearson Education,2007.
3. Sinan Ozdemir, “Principles of Data Science”, Packt.

Reference Books:

1. Rich E, Knight K,Nair S B, ArtificialIntelligence,3rdedition, Tata McGraw-Hill, 2009.
2. Luger George F, Artificial Intelligence: Structures and Strategies for Complex Problem solving,6th edition, Pearson Education, 2009.
3. Carter M, Minds and Computers: An Introduction to the Philosophy of Artificial Intelligence, Edinburgh University Press, 2007.
4. Stuart Russel and Peter Norvig “AI–A Modern Approach”, 2nd Edition, Pearson Education 2007.

Title of the Course: Object Oriented Programming Course Code: UAMC0305	L	T	P	Credits
	3	--	--	3

Course Pre-Requisite: Knowledge of Programming language basics like C, Python

Course Description: In this course students will learn Object Oriented Programming Design and concepts.

Course Objectives:

1. To expose the students to concepts of Object-Oriented Paradigm
2. To explain fundamental and object-oriented concepts of Java
3. To give hands on exposure to develop applications based on concepts of Object-Oriented approach.
4. To expose students to advanced features in Java

Course Outcomes:

COs	After the completion of the course the student will be able to	Bloom's Cognitive	
		level	Descriptor
CO1	**Use** knowledge of fundamental and OOP concepts for programming	2	Use
CO2	**Select** appropriate approach from procedural and object oriented to solve the given problem	3	Select
CO3	**Apply** knowledge of various concepts of computer science and design solutions for different subjects like computer algorithm,	4	Apply

CO-PO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO 12	PS O1	PS O2
CO1	3												1	1
CO2	2		2		2			1				1	1	1
CO3			3		3			1				1	2	3

Assessment Scheme:

Two components of In Semester Evaluation (ISE), One Mid Semester Examination (MSE) and one End Semester Examination (ESE) having 20%, 30% and 50% weightage respectively.

Assessment Component	Marks
ISE 1	10
MSE	30
ISE 2	10
ESE	50

ISE 1 and **ISE 2** are based on Assignment / Declared test / Quiz / Seminar / Group discussions / presentation, etc.

MSE is based on 50% of course content (first three units).

ESE is based on 100% course content with 60-70% weightage for course content (last three units) covered after MSE.	
Course Contents:	
Unit 1: Introduction to Object Oriented Design Introduction to procedural & object-oriented programming, Limitations of procedural programming, Need of object-oriented programming, Fundamentals of object-oriented programming: objects, classes, data members, methods, messages, data encapsulation, data abstraction and information hiding, inheritance, polymorphism	**6 Hrs**
Unit 2: Basics of Core Java Programming The Java Programming Environment- JVM, JIT Compiler, Byte Code Concept, A Simple Java Program, Source File Declaration Rules, Comments, Data Types, Variables, Operators, Strings, Input and Output, Control Flow, Big Numbers, Arrays, Jagged Array.	**7 Hrs**
Unit 3: Object Oriented Design in Java Objects and Classes: Object Oriented Programming Concepts, Declaring Classes, Declaring Member Variables, Defining Methods, Constructor, Passing Information to a Method or a Constructor, Creating and using objects, Controlling Access to Class Members, Static Fields and Methods, this keyword, Object Cloning, Class Design	**7 Hrs**
Unit 4: Interface, Inheritance and Packaging Implementing an Interface, Using an Interface as a Type, Evolving Interfaces, Default Methods. Inheritance: Definition, Super classes, and Subclasses, Overriding and Hiding Methods, Polymorphism, Inheritance Hierarchies, Super keyword, Final Classes and Methods, Abstract Classes and Methods, casting, Design Hints for Inheritance, Nested classes & Inner Classes, finalization and garbage collection. Packages: Class importing, creating a Package, Naming a Package, Using Package Members, Managing Source and Class Files. Developing and deploying (executable) Jar File	**6 Hrs**
Unit 5: Exception and I/O Streams Exception: Definition, Dealing with Errors, The Classification of Exceptions, Declaring Checked Exceptions, Throw an Exception, Creating Exception Classes, Catching Exceptions, Catching Multiple Exceptions, Re-throwing and Chaining Exceptions, finally clause, Advantages of Exceptions, Tips for Using Exceptions. I/O Streams: Byte Stream – Input Stream, Output Stream, DataInputStream, DataOutputStream, FileInputStream, FileOutputStream, Character Streams, BufferedStream, Scanner, File, RandomAccesFile	**7 Hrs**
Unit 6: Collection and Database Programming Collections: Collection Interfaces, Concrete Collections- List, Queue, Set, Map, the Collections Framework.	**6 Hrs**
Text Books: 1. Core Java- Volume I Fundamentals: Cay Horstmann and Gary Cornell, Pearson, Eight edition (Unit 1 to Unit 4). 2. Core Java- Volume II Advanced Features: Cay Horstmann and Gary Cornell, Pearson, Eight	

edition (Unit 5 and Unit 6)

Reference Books:

1. The Java Tutorials from ORACLE Java Documentation URL: http://docs.oracle.com/javase/tutorial/ (Refer For All Units)

2. The Java Tutorial: A Short Course on the Basics by Raymond Gallardo, Scott Hommel, Sowmya Kannan, Publisher: Addison-Wesley Professional. (6th Edition)

3. JAVA-The Complete Reference: Herbert Schildt, Oracle Press, Mcgraw Hill,(9th Edition).

4. JAVA™ HOW TO PROGRAM, By Deitel Paul , Deitel Harvey. Publisher: PHI Learning..(10th Edition) 5]Thinking in Java by Bruce Eckel, Prentice Hall,(4th Edition)

Title of the Course: Software Engineering	**L**	**T**	**P**	**Credits**
Course Code: UAMC0306	**3**	**--**	**--**	**3**

Course Description: In This course provides basic concepts, principles of software engineering &basics of Project Management

Course Objectives:

1. To expose the students to basic concepts, principles of software engineering & importance of

SDLC in their project development work.

2. To expose the students to software testing techniques and software quality management.

3. To introduce students' basics of Object-Oriented Modeling and Design.

4. To make the student aware of role of Software Engineering in Project Management.

Course Outcomes:

COs	After the completion of the course the student will be able to	Bloom's Cognitive	
		level	Descriptor
CO1	**Explain** the Software Development Process	2	Explain
CO2	**Illustrate** the Software Testing techniques and Quality Assurance in detail	2	Illustrate
CO3	**Make** use of Project management Concepts in the project development.	3	Make
CO4	**Design** the solution to the problems using Object Oriented Modelling with UML	4	Design

CO-PO Mapping:

CO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO 12**	**PS O1**	**PS O2**
CO1	3													
CO2														
CO3									2		3		1	
CO4		2	3	1	1				3		2		1	2

Assessment Scheme:

Two components of In Semester Evaluation (ISE), One Mid Semester Examination (MSE) and one End Semester Examination (ESE) having 20%, 30% and 50% weightage respectively.

Assessment Component	**Marks**
ISE 1	10
MSE	30
ISE 2	10
ESE	50

ISE 1 and **ISE 2** are based on Assignment / Declared test / Quiz / Seminar / Group discussions / presentation, etc.

MSE is based on 50% of course content (first three units).

ESE is based on 100% course content with 60-70% weightage for course content (last three units) covered after MSE.

Course Contents:

Unit 1: The software Problem Software Problems, Software Engineering Problems; Cost, schedule & Quality, Scale and change; Software Development process Modules; Project Management Process; Software Processes: Process & Project	**5 Hrs**
Unit 2: Requirements Analysis & specification Requirements gathering & Analysis; Software Requirements Specifications; Collecting Requirements, Defining Scope; Creating the Work Breakdown Structure, Validating Scope, Controlling Scope; Basic Principles of Cost Management; Planning Cost Management, Estimating Costs; Determining the Budget, Controlling Costs, Formal System Development Techniques	**6 Hrs**
Unit 3: Design Various Design Concepts; Function Oriented Design; Object Oriented Design; Detail Design; Verification; Metrics	**4 Hrs**
Unit 4: Object Oriented Modeling and Design Object Oriented Design: What is Object Orientation? What is OO Development? OO Themes; Modeling as Design Techniques: Modeling, Abstraction, Three Models; Overview of UML ; Conceptual Model of UML; Architecture View	**6 Hrs**
Unit 5: Coding & Testing Coding & Code Review; Testing; Unit Testing, Black Box Testing, White Box Testing, Integration Testing, System Testing	**7 Hrs**
Unit 6: Quality Management Importance, Planning Quality Management, Performing Quality Assurance, Controlling Quality, Tools and Techniques for Quality Control, Modern Quality Management, Improving IT Project Qualities 9000 SEI capability Maturity Model, Six Sigma, Agile software Development & Extreme Programming Agile Project Management	**7 Hrs**

Text Books:

1. Software Engineering: A precise Approach - Pankaj Jalote (Wiley India)
2. Information Technology Project Management, 7E, Kathy Schwalbe, Cengage Learning (India Edition)
3. Object Oriented Modeling and Design with UML, Michel R Blaha, James R Rambaugh, Second Edition

Reference Books:

1. IT Project Management, 3 E, Joseph Phillips, McGraw Hill Edu. (India) Pvt. Ltd.
2. Software Project Management, Bob Huges, Mike Cotterell, Rajib Mall, 5/E, Tata McGraw Hill Edu. (India) Pvt. Ltd.

Title of the Course: Data Structure-Lab Course Code: UAMC0331	L	T	P	Credits
	--	--	2	1

Course Pre-Requisite: C/Python Programming Language

Course Description: In This course students will Study and implement data structure concepts

Course Objectives:

1. To learn basic concepts of C language structures, Arrays, lists pointers.
2. To become familiar with advanced data structures such as Stacks, Queues, Trees etc.
3. To analyze and solve problems using advanced data structures such as Lists, Linked Lists, Queues, Stacks, Trees, and Graphs.
4. To write programs on Linked Lists, Doubly Linked Lists, Trees etc.

Course Outcomes:

COs	After the completion of the course the student will be able to	Bloom's Cognitive	
		level	Descriptor
CO1	**Define** the basic terms of Linear Lists, Linked List, Doubly Linked List, Non-Linear Data Structures (Binary Trees, AVL Trees, Graphs)	2	Define
CO2	**Choose** the appropriate and optimal data structure for a specified application	3	Choose
CO3	**Write** programs and applications with Static and Dynamic data structures	3	Write

CO-PO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO 12	PS O1	PS O2
CO1	3													
CO2	3													
CO3			2											
CO4														

Assessment Scheme:

Two components of In Semester Evaluation (ISE), One Mid Semester Examination (MSE) and one End Semester Examination (ESE) having 20%, 30% and 50% weightage respectively.

Assessment Component	Marks
ISE	25
ESE(OE)	25

One component of In Semester Evaluation (ISE) and one End Semester Examination (ESE) having 50% and 50% weights respectively.

Course Contents:

Experiment No. 1:- Write a program for matrix multiplication using arrays	2 Hrs

Experiment No. 2:- Write a menu driven & modular program for database management of any entity.	**2 Hrs**
Experiment No. 3:- Write a program to implement polynomial operations using linked list.	**2 Hrs**
Experiment No. 4:- Write a program to implement circular queue using array.	**2 Hrs**
Experiment No. 5:- Write a program to implement doubly linked list.	**2 Hrs**
Experiment No. 6:- Write a program to implement hashing and rehashing.	**2 Hrs**
Experiment No. 7:- Write a program to implement linear search, binary search	**2 Hrs**
Experiment No. 8 :- Write a program to implement binary search tree as ADT.	**2 Hrs**
Experiment No. 9:- Write a program for tree traversal algorithm.	**2 Hrs**
Experiment No. 10:- Write a program for graph traversing DFS, BFS	**2 Hrs**

Text Books:

1. Data Structures- A Pseudo code Approach with C – Richard F. Gilberg and Behrouz A. Forouzon, Cengage Learning, Second Edition.

2. Schaum's Outlines Data Structures – Seymour Lipschutz (MGH), Tata McGraw-Hill.

3. The C Programming langauge – Kernighan and Ritchie

Title of the Course: Object Oriented Programming Lab Course Code: UAMC0332	L	T	P	Credits
	--	--	2	1

Course Pre-Requisite: Knowledge of any Programming language basics

Course Description: This course is designed to develop Java programming expertise. Upon completion, students should be able to write programs in Java. Emphasis is on Class design, Implementation, File Handling, Exception Handling and Collection Framework

Course Objectives:

1. To expose the students to concepts of Object-Oriented Paradigm
2. To make students understand the use of Java for OOP implementation
3. To develop application using Java features

Course Outcomes:

COs	After the completion of the course the student will be able to	Bloom's Cognitive	
		level	Descriptor
CO1	**Use** knowledge of fundamental and OOP concepts for Java programming	2	Use
CO2	**Design** Java program to implement OOP concepts	3	Design
CO3	**Develop** Java application by applying OOP concepts and Java Features	4	Develop

CO-PO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO 12	PS O1	PS O2
CO1	3												1	1
CO2	2		2		2			1				1	1	1
CO3			3		3			1				1	2	3

Assessment Scheme:

Two components of In Semester Evaluation (ISE), One Mid Semester Examination (MSE) and one End Semester Examination (ESE) having 20%, 30% and 50% weightage respectively.

Assessment Component	Marks
ISE	25
ESE(POE)	50

One component of In Semester Evaluation (ISE) and one End Semester Examination (ESE) having 50% and 50% weights respectively.

Course Contents:	
Experiment No. 1: Creating classes and initializing objects. **Aim and Objectives:** Applying Classes and Object concepts. **Theoretical Background:** classes and objects **Experimentation:** Develop a Java Program to implement class and create its objects. **Results and Discussions:** Objects are created, data members of class are initialized, members functions can be called. **Conclusion:** You can call non-static members of class using objects, constructor get called when you create object of class	**2 Hrs**
Experiment No. 2: Classes and Objects **Aim and Objectives:** Applying Classes and Object concepts. **Outcomes:** Student will be able to use class, its members and create objects. **Theoretical Background:** classes and objects **Experimentation:** Develop a Java Program to implement class and create its objects. **Results and Discussions:** Objects are created, data members of class are initialized, members functions can be called. **Conclusion:** You can call static members of class using class name, static data member is common between all objects.	**2 Hrs**
Experiment No. 3: Implementing multiple Inheritance in Java. **Aim and Objectives:** Multiple Inheritance in Java. **Outcomes:** Students will be able to create multiple inheritance with the help of Interfaces. **Theoretical Background:** Java do not support multiple inheritance with classes, if programmer need to multiple inheritance structure she needs to use interfaces. **Experimentation:** Developing Java program with interface inheritance. **Results and Discussions:** Interface methods are overridden in class, calling overridden functions using objects is done. **Conclusion:** The Class which inherit interface, must have to provide override methods declared in interface. The default access to members of interface is public	**2 Hrs**
Experiment No. 4: Inheritance in Java **Aim and Objectives:** Inheritance using classes. **Outcomes:** Students will be able to create hierarchical inheritance with the help of classes. **Theoretical Background:** Child class should extend base class to acquire base class properties. **Experimentation:** Create Separate Engine, Tyre, and Door Class. Create a Car class using these classes. And show functionality of each component in the car. **Results and Discussions:** When class extends from another class. The extending class is child and the class which is extended is base. The child class object can call public and protected members of base class. Child class can call its own members despite of its access specifier. When you create object of child class its constructor get called. The child class constructor calls the base class constructor. **Conclusion:** Child class acquire the properties of base class.	**2 Hrs**
Experiment No. 5: Creating Packages. **Aim and Objectives:** Creating JAR(Java Archive)-package. **Outcomes:** Student will be able to create deployable packages of Java application. **Theoretical Background:** Application need multiple classes. You cannot have all classes in single file. So, you have to club together the required classes of your application. This can be done with packages in Java. The archive of these packages is called as JAR. Jar file can be created as executable and non-executable jar files. **Experimentation:** -Develop a mathematical package for Statistical operations like Mean, Median,	**2 Hrs**

Average, Standard deviation. Create a sub package in the math package -convert. In “convert” package provide classes to convert decimal to octal, binary, hex and vice-versa. Develop application program to use this package, and build executable jar file of it. **Results and Discussions:** The default access of class is package. You can create executable version of JAR file. **Conclusion:** The applications developed using Java, which contains multiple “.class” files and hierarchical directory structure should be deployed using JAR	
Experiment No. 6: Understanding Constructors **Aim and Objectives:** Understanding and using parameterized constructors in Java. **Outcomes:** Students will be able to use multiple constructor in single class. **Theoretical Background:** Java provides zero parameter constructor, program do not need to write it. This is the reason, the zero-parameter constructor is called as default constructor. Programmer can write its own constructor which is zero parameter or parameterized constructor. When programmer choose to write constructor in class, java do not provide default constructor. In this case the programmer has to write zero parameter constructor on its own. **Experimentation:** Develop a class Expr to create and evaluate given expression. Constructor accepts the expression as String. For example, Expr("x^2") or Expr("sin(x)+3*x"). If the parameter in the constructor call does not represent a legal expression, then the constructor throws an IllegalArgumentException. The message in the exception describes the error. Provide eval(double num) and eval(int num) method to evaluate given expression and return evaluated answer. For example, if Expr represents the expression 3*x+1, then func.value(5) is 3*5+1, or16.Finally,getDefinition() returns the definition of the expression. This is just the string that was used in the constructor that created the expression object **Results and Discussions:** Calling specific constructor from available ones. Passing arguments to object. Use of “this” keyword to identify current object class members. **Conclusion:** Java class can have overload of constructors, This keyword identifies current object under execution	**2 Hrs**
Experiment No. 7: Exception Handling in Java. **Aim and Objectives:** Handling exception in java program. **Outcomes:** Students will be able to deal/tackle runtime error in java program. **Theoretical Background:** Java have two error reporting mechanisms. One is compile time and one is runtime. The compile time error deals with syntax of Java programming language and you can not get byte code of java program unless your program is error free. The run time error are called as exceptions. They represent semantic issues in program. **Experimentation:** Write a class to represent Roman numerals. The class should have two constructors. One constructs a Roman numeral from a string such as "XVII" or "MCMXCV". It should throw a NumberFormatException if the string is not a legal Roman numeral. The other constructor constructs a Roman numeral from an int. It should throw a NumberFormatException if the int is outside the range 1 to 3999.In addition, the class should have two instance methods. The method toString() returns the string that represents the Roman numeral. The method toInt() returns the value of the Roman numeral as an int **Results and Discussions:** You can handle exceptions using try, catch blocks, The mission critical code which should execute in the both scenarios- i.e whether exception occurred or application executed normally- should be written in finally block. **Conclusion:** Runtime errors can be handled with try, catch and finally block	**2 Hrs**
Experiment No. 8: File Handling **Aim and Objectives:** Performing file handling using Java Program. **Outcomes:** Student will be able to use I/O streaming classes in Java for file handling. **Theoretical Background**: Java provide Input and Output streaming classes. They can be used to deal with input output devices. Here students will learn how to write file operation using Java program. **Experimentation:** Take file name as input to your program, If file is existing the open and display contents of the file. After displaying contents of file ask user – do you want to add the data at the end of file. If a	**2 Hrs**

user gives yes as response, then accept data from user and append it to file. If file in not existing then create a fresh
new file and store user data into it. User should type exit on new line to stop the program.
Results and Discussions: Java can deal with file using byte streams and character streams. It has variety of classes to deal with file operation.
Conclusion: Java application can perform file handling.

Experiment No. 9: Buffered Streams in Java. **2 Hrs**
Aim and Objectives: Using wrapper classes to reduce disk operations in file handling.
Outcomes: Student will be able to develop efficient file handling programs.
Theoretical Background: Buffered Reader/Writer, DataInputStream/ Output Stream, Print Writer are wrapper classes, which improves efficiency of Java program. Student should be able to use buffering of data to avoid disk I/O operations for every read and write of character/word to and from file.
Experimentation: Take Student information such as name, age, weight, height, city, phone from user and store it in the file using DataOutputStream and FileOutputStream and Retrieve data using DataInputStream and FileInputStream and display the result.
Results and Discussions: Using buffering for file handling is efficient than byte or character data.
Conclusion: Students can use buffers to deal with file handling to avoid unnecessary disk access.

Experiment No.10: Collection Framework in Java **2 Hrs**
Aim and Objectives: Using Java Collection Framework Create Objects.
Outcomes: Student will be able to implement List, Map, Queue interface and sub classes
Theoretical Background: List, Map and Queue are java collection Interfaces. Implementation classes like ArrayList, HashMap used to store the object and data
Experimentation: create ArrayList to store department names as CSE, IT, MECH, CIVIL, ENV, PROD, BIO, ENTC.
Include Below ArrayList Methods:
1. add 2. remove 3. size
4. display all elements using system.out.println
5. add an Iterator to ArrayList to retrieve elements and display
6. clear ()
7. display all elements using system.out.println
Results and Discussions: ArrayList, HashMap provides inbuilt methods to store and travel object data.
Conclusion: Students can use ArrayList, HashMap for store and access object data more efficiently

Text Books:

1. Core Java- Volume I Fundamentals: Cay Horstmann and Gary Cornell, Pearson, Eight edition (Unit 1 to Unit 4).

2. Core Java- Volume II Advanced Features: Cay Horstmann and Gary Cornell ,Pearson, Eight edition(Unit 5 and Unit 6)

References:

1] The Java Tutorials From ORACLE Java Documentation URL: http://docs.oracle.com/javase/tutorial/ (Refer For All Units)

2]The Java Tutorial: A Short Course on the Basics by Raymond Gallardo, Scott Hommel, Sowmya Kannan, Publisher: Addison-Wesley Professional. (6th Edition)

3]JAVA-The Complete Reference: Herbert Schildt, Oracle Press, Mcgraw Hill,(9th Edition).

4]JAVA™ HOW TO PROGRAM, By Deitel Paul , Deitel Harvey. Publisher: PHI Learning..(10th Edition)

5]Thinking in Java by Bruce Eckel, Prentice Hall,(4th Edition)

Title of the Course: Constitution of India Course Code: UAMA0361	L	T	P	Credit
	2	-	-	-

Course Pre-Requisite: Basics of Indian History, Independence Movement, Fundamentals of Civics.

Course Description: This Course is an introduction of Indian Constitution and basic concepts highlighted in this course for understanding the Constitution of India. This course is structured to give a deeper insight for making the nexus between the other law subjects.

Course Objectives

At the end of the course the student is expected to have acquired:

1. A basic understanding of Constitution of India.
2. Builds the ability to apply the knowledge gained from the course to current social legal issues.
3. Ability to understand and solve the contemporary challenges.
4. Understanding constitutional remedies.

Course Learning Outcomes:

CO	**After the completion of the course the student should be able to**	**Bloom's Cognitive**	
		level	Descriptor
CO1	**Explain** the significance of Indian Constitution as the fundamental law of the land	II	Cognitive (Understand)
CO2	Exercise his fundamental rights in proper sense at the same time **Identifies** his responsibilities in national building.	II	Cognitive (Applying)
CO3	**Analyze** the Indian political system, the powers and functions of the Union, State and Local Governments in detail	II	Cognitive (Understand)
CO4	**Understand** Electoral Process, Emergency provisions and Amendment procedure.	II	Cognitive (Understand)

CO-PO Mapping:

CO	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**PSO1**	**PSO2**	**PSO3**
CO1						3		3				3			
CO2						3		3	3	3		3			
CO3						3			3			3			
CO4						3			3			3			

Assessments:

Teacher Assessment:

One End Semester Examination (ESE) having 100% weights respectively.

Assessment	Marks
ESE	100

ESE: Assessment is based on 100% course content

Course Contents:

Unit 1:- Constitution – Structure and Principles 1.1: Meaning and importance of Constitution 1.2: Making of Indian Constitution – Sources 1.3: Salient features of Indian Constitution	**(03) Hrs**
Unit 2:- Fundamental Rights and Directive Principles 2.1: Fundamental Rights & Fundamental Duties 2.2: Directive Principles of State Policy	**(10) Hrs**

Unit 3:- Union Government & Executive 3.1: President of India – Qualification, Powers and Impeachment 3.2: Lok Sabha & Rajya Sabha Sabha- Composition, Powers & Functions, Scope to amendment in Constitution	**(04) Hrs**
Unit 4:- State Government & Executive 4.1: Governor – Qualification, Appointment, Powers & Functions 4.2: Legislative Assembly & Legislative Council – Composition, Powers & Functions	**(03) Hrs**
Unit 5:- The Judiciary 5.1: Features of Judicial System in India 5.2: Hierarchy of Courts, Composition and Jurisdiction	**(03) Hrs**
Unit 6:- Local Self Government and other constitutional Organizations 6.1: 73rd and 74th Constitutional Amendments 6.2: Public Service Commission, Election Commission, CAG, National Commissions for SC, ST etc.	**(03) Hrs**

Textbooks:

1. M.P. Jain, Indian Constitutional Law
2. M.P. Singh (ed.),V.N. Shukla, Constitutional Law of India
3. D.D. Basu, Commentary on the Constitution of India
4. S.S. Desai, Constitutional Law--I & II

References:

1. Durga Das Basu, Introduction to the Constitution of India, Gurgaon; LexisNexis, 2018 (23rd edn.)
2. J.N. Pandey, The Constitutional Law of India, Allahabad; Central Law Agency, 2018 (55th edn.)
3. Shripad Shridhar Desai, Constitutional Law--I, S.S. Law Publication, 2021
4. Shripad Shridhar Desai, Constitutional Law --II, S.S. Law Publication, 2021
5. Constitution of India (Full Text), India.gov.in., National Portal of India, https://www.india.gov.in/sites/upload_files/npi/files/coi_part_full.pdf
6. Durga Das Basu, Bharatada Samvidhana Parichaya, Gurgaon; LexisNexis Butterworths Wadhwa, 2015

SEM IV

SEMESTER IV								
Sr. No.	Course Code	Course Name	L	T	P	Hrs. / Week	Credits	Category
1	UAMC0401	Statistical and Probability Theory	3	1		4	4	BS
2	UAMC0402	Automata Theory	3	1		4	4	ES
3	UAMC0403	Computer Networks	3			3	3	PC
4	UAMC0404	Computer Architecture and Digital Logic	3	1		4	4	PC
5	UAMC0505	Principle of Data Science	3			3	3	PC
6	UAMC0431	Computer Networks Lab			2	2	1	PC
7	UAMC0432	Data Analytics & Visualization Lab			2	2	1	PC
8	UAMC0433	AI and DS Tools			2	2	1	PC
9	UAMC0451	Mini Project-I			2	2	1	PW
10	UAMA0461	Environmental Studies	2			2		MAC
		Total:				29	22	

Title of the Course: Statistical and Probability Theory Course Code: UAMC0401	L	T	P	Credits
	3	1	---	4

Course Pre-Requisite: Basic terminologies on probability and exploratory data analysis

Course Description: This course contains study of probability distribution, test of significance, regression analysis and analysis of variance.

Course Objectives:

1. To make familiar the prospective computer science engineers with techniques in data analysis techniques, probability, probability distributions and test of significance.
2. To enable students to use statistical techniques learned for the analysis, modeling and solution of realistic engineering problems.
3. To develop abstract, logical and critical thinking and the ability to reflect critically upon their work.

Course Outcomes:

COs	After the completion of the course the student will be able to	Bloom's Cognitive	
		level	Descriptor
CO1	**Understand** various concepts of data, statistical techniques, probability and test of significance.	2	Understanding
CO2	**Solve** problems on tendency of data and bivariate data using statistical techniques.	3	Applying
CO3	**Use** knowledge of probability, probability distributions and test of significance on biological experiments.	3	Applying
CO4	**Apply** the knowledge of probability distributions to the given data and select the appropriate method for testing of significance and analyze the variance.	4	Analyzing

CO-PO Mapping:

CO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO 12**	**PS O1**	**PS O2**
CO1	3	2	-	1	-	-	-	-	-	-	-	2	2	2
CO2	3	2	-	1	-	-	-	-	-	-	-	2	2	
CO3	3	2	-	2	-	-	-	-	-	-	-	2	2	1
CO4	3	2	-	2	-	-	-	-	-	-	-	2		1

Assessment Scheme:

Two components of In Semester Evaluation (ISE), One Mid Semester Examination (MSE) and one End Semester Examination (ESE) having 20%, 30% and 50% weightage respectively.

Assessment Component	**Marks**
ISE 1	10
MSE	30
ISE 2	10
ESE	50

ISE 1 and **ISE 2** are based on Assignment / Declared test / Quiz / Seminar / Group discussions / presentation, etc.

MSE is based on 50% of course content (first three units).

ESE is based on 100% course content with 60-70% weightage for course content (last three units) covered after MSE.	
Course Contents:	
Unit 1: Probability and Probability distributions. 1.1 Statistical Probability. 1.2 Conditional probability. 1.3 Random Variable, Probability mass function and density function. 1.4 Discrete Distributions: Binomial, Poisson distribution and properties. 1.5 Continuous Distributions: Normal distribution and properties.	8 **Hrs.**
Unit 2: Statistical Techniques for data analysis. 2.1 Correlation and Coefficient of correlation. 2.2 Simple Linear Regression. 2.3 Prediction, Interpreting and Diagnostics using Regression. 2.4 Fitting of curves by method of least-squares 2.4.1 Fitting of Parabolic Curves. 2.4.2 Fitting of Exponential curves. 2.5 Multiple Linear Regressions.	7 **Hrs.**
Unit 3: Classification 3.1 Logistic Regression, 3.2 Discriminate Analysis-LDA 3.3 Evaluating Classification Models: Confusion Matrix and The Rare Class Problem Precision, Recall, and Specificity, support, F1 score. 3.4 Strategies for Imbalanced Data.	6 **Hrs.**
Unit 4: Test of Significance - I 4.1 Parameter and Statistic. 4.2 Confidence Interval, p - value. 4.3 Large sample tests: 4.3.1 Test of significance for single population mean. 4.3.2 Test of significance for equality of two population means. 4.4 Small sample tests: 4.4.1 t-test for single mean. 4.4.2 t-test for difference of mean. 4.4.3 Paired t-test for difference of mean.	8 **Hrs.**
Unit 5: Test of Significance - II 5.1 Chi – square distribution. 5.2 Test for single variance. 5.3 Goodness of fit test. 5.4 Test for independence of attributes by Yates's Correction.	6 **Hrs.**

Unit 6: Analysis of Variance	7 Hrs.
6.1 F- distribution. 6.2 Test by using F- Test 6.3 Principles of experimental designs. 6.4 Analysis of variance (ANOVA) and its uses in the designs. 6.5 One Way Analysis of variance followed by t test (pair wise). 6.6 Two Way Analysis of variance followed by t test (pair wise).	

Text Books:

SN	Title	Edition	Author/s	Publisher	Year
1.	Fundamentals of Mathematical statistics	**12**	S. C. Gupta and V. K. Kapoor	Sultan Chand and Sons	**2020**
2.	Design and Analysis of Experiments	**8**	Douglas C. Montgomery	Wiley Student Edition	**2013**

Reference Books:

SN	Title	Edition	Author/s	Publisher	Year
1.	Probability and statistics for Engineers and Scientists.	**9**	Ronald Walpole	Persons Education	**2013**
2.	Applied Statistics and Probability for Engineers	**6**	Douglas C. Montgomery	Wiley Student Edition	**2012**

Title of the Course: Automata Theory Course Code: UAMC0402	L	T	P	Credits
	3	1	--	4

Course Pre-Requisite: Discrete Mathematics, Sets, Cartesian Product and Functions

Course Description: This course deals with the theoretical background of computer science with formal language & Automata

Course Objectives:

1. To expose the students to the mathematical foundations and principles of computer science.
2. To strengthen the students' ability to carry out formal and higher studies in computer science
3. To make the students understand the use of automata theory in Compilers & System programming.
4. To make the student aware of mathematical tools, formal methods & automata techniques for computing.

Course Outcomes:

COs	After the completion of the course the student will be able to	Bloom's Cognitive	
		level	Descriptor
CO1	**Explain** types of formal languages and their acceptors	2	**Explain**
CO2	**Classify** formal languages on the basis of their features	4	**Classify**
CO3	**Relate** the computational models with the modern day computer technologies	2	**Relate**
CO4	**Develop** computational machines of various types for specified problems	3	**Develop**

CO-PO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO 12	PS O1	PS O2
CO1	2													
CO2	2													
CO3		2												1
CO4			3											1

Assessment Scheme:

Two components of In Semester Evaluation (ISE), One Mid Semester Examination (MSE) and one End Semester Examination (ESE) having 20%, 30% and 50% weightage respectively.

Assessment Component	Marks
ISE 1	10
MSE	30
ISE 2	10
ESE	50

ISE 1 and **ISE 2** are based on Assignment / Declared test / Quiz / Seminar / Group discussions / presentation, etc.
MSE is based on 50% of course content (first three units).
ESE is based on 100% course content with 60-70% weightage for course content (last three units) covered after MSE.

Course Contents:

Unit 1: Mathematical Induction, Regular Languages & Finite Automata: The Principle of Mathematical Induction Recursive Definitions, Definition & types of grammars & languages, Regular expressions and corresponding regular languages, examples and applications, unions, intersection & complements of regular languages, Finite automata-definition and representation, on-deterministic F.A,NFA with null transitions, Equivalence of FA's , NFA's and NFA's with null transitions.	**08 Hrs**
Unit 2: Kleene's Theorem: Part I & II statements and proofs, minimum state of FA for a regular language, minimizing number of states in Finite Automata**.**	**04 Hrs**
Unit 3 : Grammars and Languages: Derivation and ambiguity, BNF & CNF notations, Union, Concatenation and *'s of CFLs, eliminating production & unit productions from CFG, Eliminating useless variables from a context Free Grammar. Parsing: Top-Down, Recursive Descent and Bottom-Up Parsing	**10 Hrs**
Unit 4: Push Down Automata: Definition, Deterministic PDA & types of acceptance, Equivalence of CFG's & PDA's.	**04 Hrs**
Unit 5: CFL's and non CFL's: Pumping Lemma and examples, intersections and complements.	**04 Hrs**
Unit 6: Turing Machines: Models of computation, definition of Turing Machine as Language acceptors, combining Turing Machines, Computing a function with a TM, Non-deterministic TM and Universal TM, Recursively enumerable languages.	**10 Hrs**

Text Books:

1. Introduction to languages & Theory of computations – John C. Martin (MGH) –Chapters 1, 2,3,4,5,6,7,8.
2. Discrete Mathematical Structures with applications to Computer Science—J. P. Trembley & Manohar (MGH) Chapter 1

Reference Books:

1. Introduction to Automata Theory, Languages and computation – John E. Hopcraft , Rajeev Motwani , Jeffrey D. Ullman (Pearson Edition).
2. Introduction to Theory of Computations – Michael Sipser (Thomson Brooks / Cole)
3. Theory Of Computation- Vivek Kulkarni, 1st edition OXFORD university Press
4. Theory Of Computation A problem Solving Approach Kavi Mahesh Wiley India

Title of the Course: Computer Networks Course Code: UAMC0403	L	T	P	Credits
	3	--	--	3

Course Pre-Requisite:

Course Description: This course provides a solid understanding of each of the most important networking protocols within the IP suite. The Internet protocol suite provides end-to-end data communication specifying how data should be packetized, addressed, transmitted, routed and received

Course Objectives:

1. To understand fundamental concepts of computer networking
2. To analyze simple protocols & independently study the literature concerning computer networks
3. To make students familiar with architecture and working of protocols like IP, TCP, UDP, DHCP, DNS, FTP, WWW
4. To make students able to identify client-server model and implement it using socket programming

Course Outcomes:

COs	After the completion of the course the student will be able to	Bloom's Cognitive	
		level	Descriptor
CO1	**Explain** the basic concept of Network, Transport and Application Layer.	2	Explain
CO2	**Describe** different terminologies of client server programming	2	Describe
CO3	**Illustrate** different application layer protocol like DHCP, DNS, FTP, HTTP, SMTP and SNMP	3	Illustrate
CO4	**Describe** various protocols supported by multimedia content.	2	Describe

CO-PO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO 12	PS O1	PS O2
CO1	3													
CO2	2	3			1									
CO3	2	3			1									
CO4	2	3			1									

Assessment Scheme:

Two components of In Semester Evaluation (ISE), One Mid Semester Examination (MSE) and one End Semester Examination (ESE) having 20%, 30% and 50% weightage respectively.

Assessment Component	Marks
ISE 1	10
MSE	30
ISE 2	10
ESE	50

ISE 1 and **ISE 2** are based on Assignment / Declared test / Quiz / Seminar / Group discussions / presentation, etc.

MSE is based on 50% of course content (first three units). **ESE** is based on 100% course content with 60-70% weightage for course content (last three units) covered after MSE.	
Course Contents:	
Unit 1: Computer Networks and The Internet What is the Internet; network edge; network core; Delay, Loss and throughput in Packet-Switched Networks; Protocol Layers and their Service Models	5 **Hrs.**
Unit 2 : Application Layer Principles of Network Applications; The Web and HTTP; File Transfer: FTP; Electronic Mail in the Internet; DNS - The Internet's Directory Service; Peer-to-Peer applications; Socket Programming – Creating network applications.	**7 Hrs.**
Unit 3: Transport Layer Introduction and Transport-Layer Services; Multiplexing and Demultiplexing; Connectionless Transport: UDP; Principles of Reliable of Data Transfer; Connection-Oriented Transport: TCP; Principles of Congestion Control, TCP Congestion Control	5 **Hrs.**
Unit 4: Network Layer Introduction; Virtual circuit and datagram networks; What is inside a router; Internet Protocol (IP): Forwarding and Addressing in the Internet; Routing Algorithms; Routing in the Internet; Broadcast and Multicast Routing	8 **Hrs.**
Unit 5: Data Link Layer Introduction to the link layer; Error Detection and Correction Techniques; Multiple Access links and Protocols; Switched local area networks.	7 **Hrs.**
Unit 6: Physical Layer Analog and Digital, Analog Signals, Digital Signals, Analog versus Digital, Data Rate Limits, Transmission Impairment, Digital Transmission: Line coding, Block coding, Transmission mode. Analog Transmission: Modulation of Digital Data; Telephone modems, modulation of Analog signals. Multiplexing: FDM , WDM, TDM , Transmission Media: Guided Media, Unguided media (wireless)	8 **Hrs.**
Text Books: 1. James F. Kurose and Keith W. Ross, “Computer Networking: A top-down approach”, Pearson Education, 6th edition. 2012 2. A.S. Tanenbaum, “Computer Networks”, 5th Edition, PHI 2010 **Reference Books:** 1. Bhavneet Sidhu, “An Integrated Approach to Computer Networks”, Khanna Book 2. Publishing House 2019. 3. G. Keiser, “Local Area Networks”, 2nd Edition, TMH 2002 4. D. Bertesekas and R. Gallager, “Data Networks”, 2nd Edition, PHI 2000	

5. William Stallings, “Data & Computer Communication”, PHI, 10th Edition 2013
6. B.A. Forouzan, “Data communications and networking”, TMH, 5th Edition2012
7. B.A. Forouzan, “Local Area Networks”, TMH. 2002
8. B.A. Forouzan, “TCP/IP Protocol Suite”, TMH.2004

Title of the Course: Computer Architecture and Digital Logic Course Code: UAMC0404	L	T	P	Credits
	3	1	--	4

Course Description: The course is designed to provide knowledge of basic arithmetic and logical operations in digital systems, fundamentals of 8085 Microprocessor. The course gives idea of how assembly language programming works. This course also briefs about concepts of Computer Architecture and memory system organization and architecture

Course Objectives:

1. To provide knowledge of basic arithmetic and logical operations in digital systems.
2. To expose students to basic concepts of computer organization.
3. To provide knowledge about working of microprocessors and assembly language programming
4. To introduce memory architecture of digital computer

Course Outcomes:

COs	After the completion of the course the student will be able to	Bloom's Cognitive	
		level	Descriptor
CO1	Define working of basic operations and number systems.	1	Define
CO2	Develop assembly language program.	3	Develop
CO3	Explain the organization of computers and its functions, instruction types and data formats	2	Explain
CO4	Illustrate control design & memory organization hardware in digital computers.	2	Illustrate

CO-PO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO 12	PS O1	PS O2
CO1	3												1	
CO2		2												
CO3	2													
CO4		3		3										

Assessment Scheme:

Two components of In Semester Evaluation (ISE), One Mid Semester Examination (MSE) and one End Semester Examination (ESE) having 20%, 30% and 50% weightage respectively.

Assessment Component	Marks
ISE 1	10
MSE	30
ISE 2	10
ESE	50

ISE 1 and **ISE 2** are based on Assignment / Declared test / Quiz / Seminar / Group discussions / presentation, etc.

MSE is based on 50% of course content (first three units).

ESE is based on 100% course content with 60-70% weightage for course content (last three units) covered after MSE.

Course Contents:	
UNIT-I: Fundamental Concepts & Logic Design Digital Systems, Number Systems, Binary Arithmetic, Signed Numbers, 2's complement arithmetic, BCD, Octal & Hexadecimal Arithmetic. Adder & Subtractor Circuit Design **Sequential Logic Design:** Multiplexer, De-multiplexer, Encoder and Decoder (BCD to 7 segment decoder) **Combinational Logic Design:** Flip-flops (S-R, J-K, Master-Slave, T, D), Shift Registers, Counters, Designing Synchronous & Asynchronous Counters.	**10 Hrs**
UNIT-II: 8085 Microprocessor Architecture & Assembly Language Microprocessor, 8085 Architecture, Interrupts, Memory Interfacing & Address Decoding. Timing Diagrams for Op-code fetch, Memory / IO Read and Write Operation, 8085 Instruction Groups, Addressing Modes, Writing and execution assembly language program.	**8 Hrs**
UNIT-III: Basic Computer Organization Evolution of computers - Electronic computers-generations, VLSI era, CPU organization, user and supervisor modes, accumulator-based CPU, System bus, types of instruction (zero, one, two and three address machines), RISC& CISC, definition, comparison and examples, Data representation: Fixed-Point Numbers, Floating Point Number- The IEEE 754 floating pointing numbers	**8 Hrs**
UNIT-IV: Computer Arithmetic Fixed point arithmetic - Addition and subtraction, overflow, high speed adders, adder expansion, Fixed point multiplication - Two's complement multiplier, Booth's algorithm, Combinational array multiplier, Fixed point division - Restoring, Non-restoring algorithm, Combinational array divider, Division by repeated multiplication, Floating point arithmetic - Basic operations, Difficulties, Floating point units, Addition, subtraction, multiplication, division.	**10 Hrs.**
UNIT-V: Control Design Introduction, multi cycle operation, implementation methods, Hardwired control, design methods, state tables, GCD processor, Classical method, one hot method, Micro programmed control -Basic concepts, control unit organization, parallelism in microinstructions, Microinstruction addressing, timing, Control unit organization	**6 Hrs.**
UNIT-VI: Memory Organization Types of memory, Memory systems, multi-level, address translation, memory allocation, Caches, memory mapped I/O, I/O mapped I/O.	**8 Hrs.**

Text Books:

1.Kevin Night and Elaine Rich, Nair B ,“Artificial Intelligence(SIE)”,Mc Graw Hill- 2008.

2.DanW.Patterson,“Introduction to AI and ES”, Pearson Education,2007.

3.Sinan Ozdemir, “Principles of Data Science”, Packt.

Reference Books:

1. Rich E, Knight K,Nair S B, ArtificialIntelligence,3rdedition, Tata McGraw-Hill, 2009.
2. Luger George F, Artificial Intelligence: Structures and Strategies for Complex Problem solving,6th edition, Pearson Education, 2009.
3. Carter M, Minds and Computers: An Introduction to the Philosophy of Artificial Intelligence, Edinburgh University Press, 2007.
4. Stuart Russel and Peter Norvig “AI–A Modern Approach”, 2nd Edition, Pearson Education 2007.

Title of the Course: Principles of Data Science Course Code: UAMC0405	L	T	P	Credits
	3	--	--	3

Course Pre-Requisite: Computer Science basics

Course Description: This course deals with the principles of data science.

Course Objectives: The objective of this course is to impart necessary knowledge of the mathematical foundations needed for data science and develop programming skills required to build data science applications.

1. To expose the students to Demonstrate understanding of the mathematical foundations needed for data science.
2. To strengthen the students' ability to carry out Collect, explore, clean, munge and manipulate data.
3. To make the students understand to Implement models such as k-nearest Neighbors, Naive Bayes, linear and logistic regression, decision trees, neural networks and clustering.
4. To make the student aware of Build data science applications using Python based toolkits**.**

Course Outcomes:

COs	After the completion of the course the student will be able to	Bloom's Cognitive	
		level	Descriptor
CO1	Demonstrate understanding of the mathematical foundations needed for data science.	4	Demonstrate
CO2	Collect, explore, clean and manipulate data.	4	Explore
CO3	Implement models such as k-nearest Neighbors, Naive Bayes, linear and logistic regression, decision trees, neural networks and clustering.	5	Implement
CO4	Build data science applications using Python based toolkits.	5	Build

CO-PO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO 12	PS O1	PS O2
CO1	2													
CO2	2													
CO3		2										1	1	
CO4			3									1	1	2

Assessment Scheme:

Two components of In Semester Evaluation (ISE), One Mid Semester Examination (MSE) and one End Semester Examination (ESE) having 20%, 30% and 50% weightage respectively.

Assessment Component	Marks
ISE 1	10
MSE	30
ISE 2	10
ESE	50

ISE 1 and **ISE 2** are based on Assignment / Declared test / Quiz / Seminar / Group discussions / presentation, etc.

MSE is based on 50% of course content (first three units).

ESE is based on 100% course content with 60-70% weightage for course content (last three units) covered after MSE.

Course Contents:

Unit 1: What is data science? Introduction to Data Science The data science Venn diagram, Terminology, Data science case studies, Summary, Types of Data, **Flavors of Data**: Structured versus unstructured data, Quantitative and qualitative data, **The four levels of data**: Nominal level, Ordinal level, Interval level, and Ratio level	**4 Hrs**
Unit 2: The Five Steps of Data Science, Introduction to data science, Overview of the five steps, Explore the data, obtain the data, model the data, communicate and visualize the results.	**4 Hrs**
Unit 3: Concept of Data Science: Traits of Big data, Web Scraping, Analysis vs Reporting, Introduction to Programming, Tools for Data Science, Toolkits using Python: Matplotlib, NumPy, Scikit-learn, NLTK	**5 Hrs**
Unit 4: Machine Learning Overview of Machine learning concepts – Over fitting and train/test splits, Types of Machine learning – Supervised, Unsupervised, Reinforcement learning	**5 Hrs**
Unit 5: Visualizing Data: Bar Charts, Line Charts, Scatterplots Working with data: Reading Files, Scraping the Web, Using APIs (Example: Using the Twitter APIs), Cleaning and Munging, Manipulating Data, Rescaling	**6 Hrs**
Unit 6: Case Studies of Data Science **Application :** Weather forecasting, Stock market prediction, Object recognition, Real Time Sentiment Analysis	**5 Hrs**

Text Books:

1. Joel Grus, "Data Science from Scratch: First Principles with Python", O'Reilly Media

2. Aurélien Géron, "Hands-On Machine Learning with Scikit-Learn and Tensor Flow: Concepts, Tools, and Techniques to Build Intelligent Systems", 1st Edition, O'Reilly Media

3. Jain V.K., “Data Sciences”, Khanna Publishing House, Delhi

Reference Books:

1.Jain V.K., “Big Data and Hadoop”, Khanna Publishing House, Delhi.

2. Jeeva Jose, “Machine Learning”, Khanna Publishing House, Delhi.

3. Chopra Rajiv, “Machine Learning”, Khanna Publishing House, Delhi.

4.Ian Goodfellow, Yoshua Bengio and Aaron Courville, "Deep Learning", MIT Press

5. Jiawei Han and Jian Pei, "Data Mining Concepts and Techniques", Third Edition, Morgan Kaufmann Publishers

Title of the Course: Computer Networks Lab **Course Code: UAMC0431**	**L**	**T**	**P**	**Credits**
	--	**--**	**2**	**1**

Course Pre-Requisite: Computer Networks Theory and Fundamentals of any programming Language

Course Description: Study top four layers of OSI networking model and implement example programs at different layers and use different networking tools**.**

Course Objectives:

1. Basic concepts of Client Server model of Internet using Socket programming
2. Logical addressing of computers/nodes in LAN/WAN.
3. Application layer protocols such as HTTP, FTP, TELNET, DHCP etc.
4. Networking tools such as Packet Tracer TCPDUMP and Wireshark to analyze protocols

Course Outcomes:

COs	After the completion of the course the student will be able to	Bloom's Cognitive	
		level	Descriptor
CO1	**Design** network for an organization as per the requirements	3	Design
CO2	**Design** UDP and TCP client server program to demonstrate simple, iterative and concurrent server	3	Design
CO3	**Develop** working of different routing protocols and application layer protocols using Wireshark/Packet Tracer/TCPDump	4	Develop
CO4	**Develop** client server program to send and receive email, web pages.	4	Develop

CO-PO Mapping:

CO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO 12**	**PS O1**	**PS O2**
CO1	**2**							**1**	**2**	**2**		**1**		**2**
CO2			**3**					**1**						
CO3		**1**			**2**			**1**					**2**	
CO4			**3**		**2**	**1**	**1**	**1**				**1**	**2**	

Assessment Scheme:

Two components of In Semester Evaluation (ISE), One Mid Semester Examination (MSE) and one End Semester Examination (ESE) having 20%, 30% and 50% weightage respectively.

Assessment Component	**Marks**
ISE	25
ESE(OE)	25

One component of In Semester Evaluation (ISE) and one End Semester Examination (ESE) having 50% and 50% weights respectively.

Course Contents:

Experiment No. 0: Socket Programming API in C/Java	**2 Hrs**
Experiment No. 1: Well Known Server and Client	**2 Hrs**
Experiment No. 2: Routing Algorithm and Protocols A) Implementation of Shortest Path routing algorithm in C /Java programming language B) Implementation of Distance Vector routing algorithm in C /Java programming language. C) Simulation of Routing Protocols	**2 Hrs**
Experiment No. 3: Implementation of C/Java program to find Network ID, Host ID and the network Class of a given input IPv4 Address.	**2 Hrs**
Experiment No. 4: Implementation of Iterative Client / Server Model using TCP Sockets	**2 Hrs**
Experiment No. 5: Implementation of Concurrent Client / Server Model using TCP Sockets.	**2 Hrs**
Experiment No. 6: Implementation of Client / Server Model using UDP sockets.	**2 Hrs**
Experiment No. 7: Communication using IPv6	**2 Hrs**
Experiment No. 8: Packet Capturing and Analysis	**2 Hrs**
Experiment No. 9: Demonstration of working of DHCP, DNS, FTP, SSH, TELNET protocols	**2 Hrs**
Experiment No. 10: Write a simulation of sliding window protocols	**2 Hrs**
Experiment No. 11: Write a program for error detecting codes	**2 Hrs**

Text Books:

1. Computer Networks - A Top-down Approach, Andrew S. Tanenbaum, Fifth Edition, Pearson Education
2. G. Keiser, “Local Area Networks”, 2nd Edition, TMH 2002
3. D. Bertesekas and R. Gallager, “Data Networks”, 2nd Edition, PHI 2000
4. William Stallings, “Data & Computer Communication”, PHI, 10th Edition 2013
5. B.A. Forouzan, “Data communications and networking”, TMH, 5th Edition2012
6. B.A. Forouzan, “Local Area Networks”, TMH. 2002
7. B.A. Forouzan, “TCP/IP Protocol Suite”, TMH.2004
8. Linux User guide available on Internet (freeware)
9. Unix Network Programming – W. Rhichard Stevens Second Edition (PHI)

Title of the Course: Data Analytics and visualization lab Course Code: UAMC0432	L	T	P	Credits
	--	--	2	1

Course Pre-Requisite: Statistics

Course Description: This course is intended to learn data preprocessing and visualization for machine learning modeling

Course Objectives:

1. To Explore the students, understand the data analysis and visualization techniques.
2. To make students do hands-on exercise on data analytics, used to build models.
3. To make students understand how to perform data wrangling, cleaning, and sampling to get a suitable data set.
4. To make students do exploratory data analysis– summarizing results through various visualization techniques and providing interpretable summaries.

Course Outcomes:

COs	After the completion of the course the student will be able to	Bloom's Cognitive	
		level	Descriptor
CO1	Explain and demonstrate various techniques for data collection, data cleaning and exploration using visualizations.	3	Explain and Demonstrate
CO2	Implement data collection, data cleaning and exploration techniques in a programming language.	4	Implement
CO3	Understand and apply analysis techniques on various types of datasets.	3	Understand and apply
CO4	Select methods and create effective visualizations to explain the artifacts in the data, distributions of attributes, relationships between the attributes.	4	Select

CO-PO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO 12	PS O1	PS O2
CO1	2	1			2								1	1
CO2	2				3									2
CO3		2											1	1
CO4	2				2									2

Assessment Scheme:

Two components of In Semester Evaluation (ISE), One Mid Semester Examination (MSE) and one End Semester Examination (ESE) having 20%, 30% and 50% weightage respectively.

Assessment Component	Marks
ISE	25
ESE(POE)	50

One component of In Semester Evaluation (ISE) and one End Semester Examination (ESE) having 50% and 50% weights respectively.

Course Contents: Data science workflow, Automated methods for data collection, Data and Visualization Models, Data wrangling and cleaning, Exploratory data analysis, Dimensionality Reduction.

Experiment No. 1: Learn how to collect data via web-scraping, APIs and data connectors from suitable sources as specified by the instructor	**2 Hrs**
Experiment No. 2: Perform EDA on a given dataset and summarize the interpretation using Tools.	**2 Hrs**
Experiment No. 3: Perform dimensionality reduction on a given dataset and create various visualizations like histograms, scatter-plots, etc.	**2 Hrs**
Experiment No. 4: Perform various types of data cleaning operations on the data collected in the previous lab using data exploration, imputation etc.	**2 Hrs**
Experiment No. 5: Perform association analysis on a given dataset and evaluate its accuracy.	**2 Hrs**
Experiment No. 6: Visual Encodings and Basic Dashboards in Tableau.	**2 Hrs**
Experiment No. 7: Hierarchical and Topographical Data Visualizations in Tableau.	**2 Hrs**
Experiment No. 8: Time Series Data Visualization in Python.	**2 Hrs**
Experiment No. 9: Create PowerBI dashboard on given dataset.	**2 Hrs**
Experiment No. 10: Dashboards, Actions and Storytelling in Tableau.	**2 Hrs**

Text Books / References:

1. Skiena, Steven S, The Data Science Design Manual, CRC press AICTE Model Curriculum for UG Degree Course in Computer Science and Engineering (Artificial Intelligence and Data Science (AI&DS)) 94

2. Pang-Ning Tan, Michael Steinbach, Anuj Karpatne, Vipin Kumar, Introduction to Data Mining (Second Edition)

3. V.K. Jain, Data Science and Analytics (with Python, R and SPSS Programming), Khanna Book Publishing Company.

4. V.K. Jain, Big Data and Hadoop, Khanna Book Publishing Company, 2022.

5. Tamara Munzner, “Visualization Analysis and Design”, A K Peters/CRC Press; 1st edition (December 1, 2014)

6. Michael Minelli, Michelle Chambers, and Ambiga Dhiraj, "Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses", Wiley, 2013.

7. Matthew O. Ward, Georges Grinstein, Daniel Keim, “Interactive Data Visualization: Foundations, Techniques, and Applications”, 2nd Edition, CRC press, 2015.

Title of the Course: AI and DS Tools Lab Course Code: UAMC0433	L	T	P	Credits
	--	--	2	1

Course Pre-Requisite: AIML and Data Science Concepts

Course Description: This course is intended to learn Python packages and Tools

Course Objectives:

1. To understand and execute Python script using types and expressions
2. To utilize high level data types such as lists and dictionaries
3. To use latest python libraries for data science in real time paradigms
4. To Visualize, organize data, and design dashboards to empower more meaningful business decisions using different tools like Tableu., TensorFlow library for solving supervised and unsupervised Learning Problems, Keras library for solving supervised and unsupervised learning Problems

Course Outcomes:

CO	After the completion of the course the student should be able to	Bloom's Cognitive	
		level	Descriptor
CO1	Recognize the general principles and good Algorithmic problem solving	1	Recognize
CO2	Structure simple Python programs for solving problems.	2	Structure
CO3	Decompose a Python program into functions. Represent compound data using Python lists, tuples and dictionaries.	3	Decompose
CO4	Read and write data from data sheets and analyze data. Illustrate how to build visualizations, organize data, and design dashboards to empower more meaningful business decisions using different tools like Tableu, PowerBI, Matplotlib.	4	Illustrate
CO5	Utilize TensorFlow for solving supervised and unsupervised Learning Problems and Build supervised and unsupervised learning models using the Keras library	4	Utilize

CO-PO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2										1			
CO2		2												
CO3		2		2							2		1	1
CO4		2									2	1	1	1
CO5		2									1		1	2

Assessment Scheme:

Two components of In Semester Evaluation (ISE), One Mid Semester Examination (MSE) and one End Semester Examination (ESE) having 20%, 30% and 50% weightage respectively.

Assessment Component	Marks
ISE	25
ESE(POE)	50

One component of In Semester Evaluation (ISE) and one End Semester Examination (ESE) having 50% and 50% weights respectively.	
Course Contents:	
Unit 1: Algorithmic Problem Solving, Data, Expressions and Statements: Algorithms, Building Blocks of Algorithms (Statements, State, ControlFlow, Functions), Notation (Pseudo Code, Flow Chart, Programming Language), Algorithmic Problem Solving, Simple Strategies for Developing Algorithms (Iteration, Recursion). Illustrative Problems: Find Minimum In A List, Insert A Card In A List Of Sorted Cards, Guess An Integer Number In A Range, Towers of Hanoi. - Python Interpreter and Interactive Mode; Values and Types: Int, Float, Boolean, String, And List; Variables, Expressions, Statements, Tuple Assignment, Precedence of Operators	**8 Hrs**
Unit 2: Lists, Dictionaries with respect to data science: List Operations, List Slices, List Methods, List Loop, Mutability, Aliasing, Cloning Lists, List Parameters; Tuples: Tuple Assignment, Tuple as Return Value; Dictionaries: Operations and Methods, Exception handling, Files-reading and writing.	**7 Hrs**
Unit 3: Python Libraries for Data Science: Basics for Data Science: Loading the Data from CSV file, Cleaning the Data, Visualization, Numpy and Numpy Operations, Pandas and pandas operations, Matplotlib: types of plots. Case study: Analyze the academic performance of students and plot a graph.	**6 Hrs**
Unit 4: Introduction to Tableau: Tableau Prep, Connecting to Data. Visual Analytics: Sorting, Grouping Working with sets, Tableau Filters, Basic tools. Mapping and calculating: Maps in tableau, Spatial Files, Table calculations, LOD Expressions	**6 Hrs**
Unit 5: POWERBI Tableau Prep, Connecting to Data. Visual Analytics, BI Tools, Creating Dashboards, Matplotlib Visualization	**8 Hrs**
Unit 6: TensorFlow: TensorFlow Basic Syntax, TensorFlow Graphs, Variables and Placeholders--TF Regression and Classification:, Keras: - Deep Learning Libraries, Regression Models with Keras	**7 Hrs**
Text Books: 1. AllenB. Downey, Think Python: HowtoThinkLikeaComputerScientist, 2ndedition, Updated for Python 3, Shroff/O'Reilly Publishers,2016. (http://greenteapress.com/wp/think-python/) 2. Guidovan Rossumand Fred L. Drake Jr, "An Introduction to Python"–Revised and Updated for Python3.2, Network Theory Ltd ,2011 3. FabioNelli, PythonDataAnalytics: DataAnalysisandscienceusingpandas, matplotlib And python programming language", A press	

References:

1. Robert Sedge wick, Kevin Wayne, RobertDondero, Introduction to Programming in Python: An Inter-disciplinary Approach, PearsonIndiaEducationServicesPvt.Ltd 2016

2. TimothyA.Budd, ExploringPython, McGrawHillEducation (India)PrivateLtd.,2015

3. JohnV Guttag, Introduction to Computation and Programming Using Python, Revised andexpandedEdition, MITPress,2013

4. Peter Morgan, Data Analysis from scratch with python: Beginner guide using python, pandas, Numpy, SCIKIT-learn, IPython, TensorFlowandMatplotlib, AISciences,2018

Web Reference:

1. http://nptel.ac.in/courses/106106145/
2. https://www.codecademy.com/learn/learn-python
3. https://www.coursera.org/learn/python-data-analysis#syllabus
4. https://www.tableau.com/learn/training/20194
5. https://www.tensorflow.org/tutorials
6. https://www.udemy.com/course/complete-guide-to-tensorflow-for-deep-learningwith-python/
7. https://www.datacamp.com/community/tutorials/tensorflow-tutorial/
8. https://www.programiz.com/python-programming

Title of the Course: Mini Project-I	**L**	**T**	**P**	**Credits**
Course Code: UAMC0451	**--**	**--**	**2**	**1**

Course Pre-Requisite: UAMC0305 Object Oriented Programming

UAMC0332 Object Oriented Programming Lab

Course Description: Implementation of Mini Project using Programming Concepts.

Course Objectives: To expose students to: -

1. To identify the problem definition
2. To follow the methods and tasks of Software engineering
3. To utilize the techniques, skills and modern engineering tools necessary for building the project
4. To effectively demonstrate and present the ideas, methodology and technology used for the project

Course Outcomes:

COs	After the completion of the course the student will be able to	Bloom's Cognitive	
		level	Descriptor
CO1	**Define** the problem statement of the software project	1	Define
CO2	**Design** an effective project plan with clear and finite objective and documents the synopsis and project reports.	2	Design
CO3	**Develop** the model of project with the help of DFDs, Flowcharts, develop the modules of proposed system.	4	Develop
CO4	**Demonstrate** the test cases for validation of proposed system	4	Demonstrate

CO-PO Mapping:

CO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO 12**	**PS O1**	**PS O2**
CO1	**2**	**3**												**2**
CO2		**2**								**2**	**2**		**2**	**2**
CO3			**3**		**2**								**3**	**2**
CO4			**3**						**2**			**2**	**2**	**2**

Assessment Scheme:

Two components of In Semester Evaluation (ISE), One Mid Semester Examination (MSE) and one End Semester Examination (ESE) having 20%, 30% and 50% weightage respectively.

Assessment Component	**Marks**
ISE	50

One component of In Semester Evaluation (ISE) and one End Semester Examination (ESE) having 50% and 50% weights respectively.

Course Contents:

The mini project should be undertaken preferably by a group of 3-4 students who will jointly work and implement the project. The group will select a project with the approval of the guide and submit the name of the project with a synopsis, of the proposed work, of not more than 02 to 03 pages. The mini project should consist of defining the problem, analyzing, designing the solution and implementing it using a suitable programming language or tool. A presentation and demonstration based on the above work is to be given by the group. The work will be jointly assessed by a panel of teachers of the department. A hard copy of project report of the work done is to be submitted along with the softcopy of the project during ESE.

Rubrics for Evaluation

Sr. No	Parameter	Unacceptable (E)	Marginal (D)	Adequate-Good (B+C)	Excellent (A)
1	Requirement Analysis	Irrelevant	Partially	Properly with few points left	Requirements Analysis with all possible strategies defined
2	Design	No Meaningful Design	Incomplete System Design	Presence of system design but no Proper Detailed Design	Presence of Correct System Design and Detailed Design
3	Coding & Testing	Code will not run	Code Runs Partially	Code runs with few errors or warnings	Code runs without errors for defined test cases
4	Report (Content)	Not Proper	Relevant but no reference and details	Content with relevant data and few spelling errors	Good Content with no spelling errors