Scheme of Instruction, Evaluation

And

Syllabi of

With effect from Academic Year 2023-24

B.E. ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

III & IV Semesters

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
UNIVERSITY COLLEGE OF ENGINEERING
(Autonomous)
Esd.1917 **Hyderabad – 500 007, TS, INDIA** **Estd. 1929**

SCHEME OF INSTRUCTION AND EXAMINATION
B. E (AIML)
SEMESTER- III

SNo	Code	Course Title	Scheme of Instruction			Contact Hrs/Wk	Scheme of Examination			Credits
			L	T	P		Hrs	CIE	SEE	
Theory										
1	MT301BS	Engineering Mathematics – III (Probability & Statistics)	3	0	-		3	40	60	3
2	AI 301 PC	Data Structures	3	0	-		3	40	60	3
3	AI 302 PC	Introduction to Data Science	3	0	-		3	40	60	3
4	AI 303 PC	Java Programming	3	0	-		3	40	60	3
5	ES 301 EC	Basic Electronics	3	0	-		3	40	60	3
6	MC301HS	Managerial Economics & Accountancy	3	0	-		3	40	60	3
Practicals										
7	AI351PC	Data Structures Lab	-	-	2x2		3	25	50	2
8	AI352PC	Data Science with Python Lab	-	-	2		3	25	50	1
9	AI353PC	Java Programming Lab	-	-	2		3	25	50	1
Total			**18**	**0**	**8**		**27**	**315**	**510**	**22**

MT301BS	**ENGINEERING MATHEMATICS-III (PROBABILITY & STATISTICS)**					
Prerequisites			**L**	**T**	**P**	**C**
			3	**0**	**-**	**3**
Evaluation	**CIE**	**40 Marks**	**SEE**		**60 Marks**	

Course Objectives :	
1	To Understand the statistical Techniques
2	To study continuous Random variables and their Properties
3	To study the statistical parameters of three distributions
4	To study empirical laws and curve fitting
5	To introduce tests of significance

Course Outcomes :	
At the end of the course students will be able to	
CO-1	Solve the Problems of Skewness and Kurtosis , Poisson Approximation and other stastiscal Techniques
CO-2	Solve Continuous Random Variables, exponential and gamma densities
CO-3	Solve the Problems related to Probability distributions, Binomial, Poisson and Normal distributions
CO-4	Obtain empirical formulas of curve fitting , solve the problems of correlation regression and rank correlation
CO-5	Solve the problems of test of significance such as large sample test for single proportion

UNIT – I

Measures of Central tendency, Moments, Skewness and Kurtosis, Discrete random variables, Independent random variables, The multinomial distribution, Poisson approximation to the binomial distribution, Infinite sequences of Bernoulli trials, Sums of independent random variables, Expectation of Discrete Random Variables, Variance of a sum.

UNIT – II

Continuous random variables and their properties, Distribution functions and densities, Normal, Exponential and gamma densities.

UNIT – III

Probability distributions, Binomial, Poisson and Normal-evaluation of statistical parameters for these three distributions.

UNIT – IV

Curve fitting by the method of least squares, Fitting of straight lines, Second degree parabolas and more general curves, Correlation, Regression and Rank correlation

UNIT – V

Test of significance, Large sample test for single proportion, Difference of proportions, Single mean, difference of means, and difference of standard deviations. Small Sample test for single mean, Difference of means and correlation coefficients, Test for ratio of variances, Chi-square test for goodness of fit and independence of attributes.

Suggested Reading:

1	R. K. Jain & S.R.K Iyengar, Advanced Engineering Mathematics, Narosa Publications, 4th Edition 2014.
2	Erwin Kreyszi, Advanced Engineering Mathematics, John Wiley, 9th Edition, 2012.
3	B.S. Grewal, Higher Engineering Mathematics, Khanna Publications, 43rd Edition, 2014.
4	S.C Gupta & Kapoor: Fundamentals of Mathematical statistics, Sultan chand& sons, New Delhi.
5	B.V . Ramana, Higher Engineering Mathematics, 23rd reprint, 2015.
6	N.P. Bali and M.Goyal, A text book of Engineering Mathematics, Laxmi Publications, 2010.
7	S. Ross, “A First Course in Probability”, Pearson Education India, 2002.

AI 301 PC	**DATA STRUCTURES**					
Prerequisites	**Programming for Problem Solving**		**L**	**T**	**P**	**C**
			3	**0**	**-**	**3**
Evaluation	**CIE**	**40 Marks**	**SEE**		**60 Marks**	

Course Objectives	
1	To introduce the time and space complexities of algorithms.
2	To discuss the linear and non-linear data structures and their applications
3	To introduce the creation, insertion and deletion operations on binary search trees and balanced binary search trees.
4	To introduce various internal sorting techniques and their time complexities

Course Outcomes On completion of this course, the student will be able to	
CO1	Analyse the time and space complexities of algorithms.
CO2	Implement linear, non-linear data structures and balanced binary trees
CO3	Analyse and implement various kinds of searching and sorting techniques.
CO4	Find a suitable data structure and algorithm to solve a real world problem

UNIT – I

Performance and Complexity Analysis: Space Complexity, Time Complexity, Asymptotic Notation (Big-Oh), Complexity Analysis Examples. Linear List-Array Representation: Vector Representation Multiple Lists Single Array. Linear List-Linked Representation: Single Linked Lists, Circular Lists, Doubly Linked Lists, Applications (Polynomial Arithmetic).Arrays and Matrices: Row And Column Major Representations, Sparse Matrices.

UNIT – II

Stacks: Array Representation, Linked Representation, Applications (Recursive Calls, Infix to Postfix, Postfix Evaluation). Queues: Array Representation, Linked Representation. Skip Lists and Hashing: Skip Lists Representation, Hash Table Representation, Application- Text Compression.

UNIT – III

Trees: Definitions and Properties, Representation of Binary Trees, Operations, Binary Tree Traversal. Binary Search Trees: Definitions, Operations on Binary Search Trees. Balanced Search Trees: AVL Trees, and B-Trees.

UNIT – IV

Graphs: Definitions and Properties, Representation, Graph Search Methods (Depth First Search and Breadth First Search)Application of Graphs: Shortest Path Algorithm (Dijkstra), Minimum Spanning Tree (Prim's and Kruskal's) Algorithms

UNIT – V

Searching: Linear Search and Binary Search Techniques and their complexity analysis. Sorting and Complexity Analysis: Selection Sort, Bubble Sort, Insertion Sort, Quick Sort, Merge Sort, and Heap Sort.

Suggested Reading:

1	Sartaj Sahni, *Data Structures--Algorithms and Applications in C++*, 2nd Edition, Universities Press (India) Pvt. Ltd., 2005.
2	Mark Allen Weiss, *Data Structures and Problem Solving using C++*, Pearson Education International, 2003.
3	Michael T. Goodrich, Roberto Tamassia, David M. Mount, *Data Structures and Algorithms in C++*, John Wiley & Sons, 2010.

AI 302 PC	**INTRODUCTION TO DATA SCIENCE**					
Prerequisites			**L**	**T**	**P**	**C**
			3	**0**	**-**	**3**
Evaluation	**CIE**	**40 Marks**	**SEE**		**60 Marks**	

Course Objectives	
1	To understand the fundamental concepts of data science, its interdisciplinary nature.
2	To introduce various data analysis techniques, including descriptive analysis, diagnostic analytics, and statistical methods and to interpret and draw meaningful insights from data.
3	To understand the packages NumPy and pandas libraries for data manipulation and analysis.
4	To introduce students to supervised and unsupervised machine learning techniques.

Course Outcomes

On completion of this course, the student will be able to

CO1	Explain the scope and significance of data science in various domains such as finance, healthcare, and urban planning.
CO2	Apply descriptive statistics to analyze variables, frequency distribution, measures of centrality, and dispersion in data.
CO3	Apply data cleaning techniques using pandas, handling missing values, outliers, and feature encoding.

UNIT – I

Introduction: What Is Data Science? Where Do We See Data Science? Finance, Public Policy, Politics, Healthcare, Urban Planning, Education, Libraries. How Does Data Science Relate to Other Fields: Data Science and Statistics, Data Science and Computer Science, Data Science and Engineering, Data Science and Business Analytics, Data Science, Social Science, and Computational Social Science. The Relationship between Data Science and Information Science: Information vs. Data Users in Information Science, Data Science in Information Schools (iSchools). Computational Thinking. Skills for Data Science, Tools for Data Science, Issues of Ethics, Bias, and Privacy in Data Science, Hands-On Problems.
Data: Introduction, Data Types: Structured Data, Unstructured Data, Challenges with Unstructured Data.
Data Collections: Open Data, Social Media Data, Multimodal Data, Data Storage and Presentation. Data **Pre-processing:** Data Cleaning, Data Integration, Data Transformation, Data Reduction, Data Discretization.

UNIT – II

Techniques: Introduction: Data Analysis and Data Analytics, Descriptive Analysis: Variables, Frequency Distribution, Measures of Centrality, Dispersion of a Distribution.
Diagnostic Analytics: Correlations, Predictive Analytics, Prescriptive Analytics, Exploratory Analysis, Mechanistic Analysis: Regression.
Statistics: Understanding attributes and their types, Types of attributes, Discrete and continuous attributes, measuring central tendency: Mean, Mode, Median, measuring dispersion, Skewness and kurtosis, understanding relationships using covariance and correlation coefficients: Pearson's correlation coefficient, Spearman's rank correlation coefficient, collecting samples, Performing parametric tests.
Descriptive Statistics: Understanding statistics, distribution functions uniform distribution, normal distribution, exponential distribution, binomial distribution. Cumulative distribution function, descriptive statistics.

UNIT – III
NumPy and pandas Understanding NumPy arrays, NumPy array numerical data types, manipulating array shapes, The stacking of NumPy arrays, Partitioning NumPy arrays, Changing the data type of NumPy arrays, Creating NumPy views and copies, Slicing NumPy arrays, Boolean and fancy indexing, Broadcasting arrays, Creating pandas Data Frames, understanding pandas Series, Reading and querying the **Quandl** data, describing pandas Data Frames, Grouping and joining pandas Data Frame, working with missing values, creating pivot tables, Dealing with dates. **linear algebra:** Fitting to polynomials with numpy, determinant, finding the rank of a matrix, matrix inverse using numpy, solving linear equations using numpy, decomposing a matrix using svd, eigne vectors and eigen values using numpy, generating random numbers.

UNIT – IV
Data Cleaning: Exploring data, Filtering data to weed out the noise, Column-wise filtration, row-wise filtration. Handling missing values, dropping missing values, Filling in a missing value, Handling outliers, Feature encoding techniques: one-hot encoding, Label encoding, ordinal encoder. Correlation: Introducing correlation, Types of analysis, understanding univariate analysis, understanding bivariate analysis, Understanding multivariate analysis. Discussing multivariate analysis using the Titanic dataset. **Hypothesis Testing and Regression:** Hypothesis testing, Hypothesis testing principle, stats models library, Average reading time, Types of hypothesis testing, T-test.

UNIT – V
Supervised Learning - Regression Analysis, Linear regression, Multiple linear regression, understanding multicollinearity, removing multicollinearity, Dummy variables, developing a linear regression model, Evaluating regression model performance, R-squared, MSE, MAE, RMSE, fitting polynomial regression, Regression models for classification, Logistic regression, Characteristics of the logistic regression model, Types of logistic regression algorithms, Advantages and disadvantages of logistic regression, implementing logistic regression using scikit-learn. **Unsupervised Learning clustering -** PCA and Clustering: Unsupervised learning, Reducing the dimensionality of data, PCA, Performing PCA. Finding the number of clusters, the elbow method, the silhouette method, Partitioning data using k-means clustering, Hierarchical clustering, DBSCAN clustering, Spectral Clustering: clustering, Evaluating clustering performance, Internal performance evaluation, The Davies-Bouldin index, The silhouette coefficient, External performance evaluation, The Rand score, The Jaccard score, F-Measure or F1-score, The Fowlkes-Mallows score.

Suggested Reading:

1	*A Hands-on introduction to Data Science*, Chirag Shah, by Cambridge University Press, 2020
2	*Python Data Analysis: Perform data collection, data processing, wrangling, visualization, and model building using Python*, 3rd Edition. Avinash Navlani, Armando Fandango, Ivan Idris, Packt Publishing, 2021.
3	*Hands-On Exploratory Data Analysis with Python*, Suresh Kumar Mukhiya, Usman Ahmed, Packt Publishing, 2021.

AI 303 PC	**JAVA PROGRAMMING**					
Prerequisites	**Programming for Problem Solving**		**L**	**T**	**P**	**C**
			3	**0**	**-**	**3**
Evaluation	**CIE**	**40 Marks**	**SEE**		**60 Marks**	

Course Objectives	
1	To introduce fundamental object oriented concepts of Java programming Language-such as classes, inheritance packages and interfaces.
2	To introduce concepts of exception handling and multithreading.
3	To use various classes and interfaces in java collection framework and utility classes.
4	To understand the concepts of GUI programming using AWT and Swing controls.
5	To introduce Java I/O streams and serialization.

Course Outcomes On completion of this course, the student will be able to	
CO1	Use object-oriented programming concepts to solve real world problems.
CO2	Demonstrate the behaviour of programs involving constructs like string, arrays,
CO3	Understand the impact of exception handling to avoid abnormal termination of program and able to solve multi-threaded programs with synchronization.
CO4	Implement real world applications using java collection frame work and I/O classes
CO5	Write Event driven GUI programs using AWT/Swing

UNIT – I
Object Oriented System Development: Understanding object oriented development, Understanding object oriented concepts, benefits of object oriented development.
Java Programming Fundamentals: Introduction, overview of Java, data types, variables and arrays, operators, control statements, classes, methods, inheritance, packages and interface

UNIT – II
Exceptional Handling, Multithreaded Programming, I/O Basics, Reading Console Input and Output, Reading and Writing Files, Print Writer Class, String Handling.

UNIT – III
Exploring Java. Lang, Collections Overview, Collection Interfaces, Collection Classes, Iterators, Random Access Interface, Maps, Comparators, Arrays, Legacy Classes and Interfaces, String Tokenizer, Bit set, Date, Calendar, Observable Timer.

UNIT – IV
GUI Programming & Event Handling: Event Handling Mechanisms, The Delegation Event Model, Event Classes, Source of Events, Event Listener Interfaces, Handling mouse and keyboard events, Adapter classes, Inner classes, Anonymous Inner classes, Introduction, AWT classes working with Graphics, Understanding Layout Managers, Flow Layout, Border Layout, Grid Layout, Card Layout, Grid Bag Layout.
Java Swing: Basics of Swing, Difference between AWT & Swing, MVC Architecture, Components and Container, Exploring Swing Controls-J Label and Image Icon, J Text Field. The Swing Buttons-J Button, J Toggle Button, J Check Box, J Radio Button, J Tabbed Pane, J Scroll Pane, J List, J Combo Box, Swing Menus, Dialogs

UNIT – V
Java I/O Classes and Interfaces, Files, Stream and Byte Classes, Character Streams, Serialization.

Suggested Readings:

1	Herbert Schildt, “The Complete Reference *JAVA*”, Tata McGraw Hill,7th Edition,2005.
2	James M Slack, “*Programming and Problem Solving with JAVA*”, First Edition, Thomson Learning, 2002
3	C.Thomas Wu, “*An Introduction to Object-Oriented Programming with JAVA*”, Tata McGrawHill, 5th Edition, 2005

ES 301 EC	**BASIC ELECTRONICS**					
Prerequisites			**L**	**T**	**P**	**C**
			3	**0**	**-**	**3**
Evaluation	**CIE**	**40 Marks**	**SEE**		**60 Marks**	

Course Objectives	
1	To study the characteristics of P-N Junction diodes and their parameters such as forward and reverse bias behavior, breakdown voltage, and rectification properties.
2	To understand the construction, working, and parameters of BJTs, Junction Field-Effect Transistors(JFETs).
3	To classify feedback amplifiers based on their configurations and characteristics.
4	To study the applications of OP Amps in inverting, non-inverting amplifiers, summer, integrator, and differentiator circuits.
5	To understand the principles of Photoelectric Devices (Photo diode and Photo Transistor), Light Emitting Diodes (LEDs), and Liquid Crystal Displays (LCDs).

Course Outcomes On completion of this course, the student will be able to	
CO1	Analyze the characteristics of P-N Junction diodes and understand their parameters
CO2	Apply the operation and characteristics of different BJT ,FET configurations and their use as amplifiers
CO3	Design and analyze inverting, non-inverting amplifiers, integrators, differentiators, and other OP Amp-based circuits
CO4	Design and analyze inverting, non-inverting amplifiers, integrators, differentiators, and other OP Amp-based circuits
CO5	Understand the working principles of different transducers and their applications in data acquisition system

UNIT – I

Semi-Conductor Theory: Energy Levels, Intrinsic and Extrinsic Semiconductors, Mobility, Diffusion and Drift current. Hall Effect, Characteristics of P-N Junction diode, Parameters and Applications.
Rectifiers: Half wave and Full wave Rectifiers (Bridge, center tapped) with and without filters, ripple regulation and efficiency. Zener diode as voltage regulator.

UNIT – II

Bipolar Junction Transistor: BJT, Current components, CE, CB, CC configurations, characteristics, Transistor as amplifier. Analysis of CE, CB, CC Amplifiers (qualitative treatment only).
JFET: Construction and working, parameters, CS, CG, CD Characteristics, CS amplifier.

UNIT – III

Feedback Concepts – Properties of Negative Feedback Amplifiers, Classification, Parameters. Oscillators – Barkhausen Criterion, LC Type and RC Type Oscillators and Crystal Oscillators. (Qualitative treatment only).

UNIT – IV
Operational Amplifiers – Introduction to OP Amp, characteristics and applications– Inverting and Non-inverting Amplifiers, Summer, Integrator, Differentiator, Instrumentation Amplifier. Digital Systems: Basic Logic Gates, half, Full Adder and Subtractors.

UNIT – V
Data Acquisition Systems: Study of transducer (LVDT, Strain gauge, Temperature,and Force). Photo Electric Devices and Industrial Devices: Photo diode, Photo Transistor, LED, LCD, SCR, UJT Construction and Characteristics only. **Display Systems:** Constructional details of C.R.O and Applications.

Suggested Reading:

1	Jocob Millman, Christos C. Halkias and Satyabrata Jit, Electronics Devices and Circuits, 3rd Edition, McGraw Hill Education (India) Private Limited, 2010.
2	Rama Kanth A. Gaykward, Op-AMPS and Linear Integrated Circuit, 4th Edition, Prentice Hall of India, 2000
3	M. Morris Mano, Digital Design, 3rdEdition, Prentice Hall of India, 2002.
4	William D Cooper, and A.D. Helfrick, Electronic Measurements and Instrumentations Techniques, 2nd Edition, Prentice Hall of India, 2008.
5	S. Shalivahan, N. Suresh Kumar, A. Vallava Raj, Electronic Devices and Circuits, 2nd Edition., McGraw Hill Education (India) Private Limited, 2007.

MC301HS	**MANAGERIAL ECONOMICS & ACCOUNTANCY**					
Prerequisites			**L**	**T**	**P**	**C**
			3	**0**	**-**	**3**
Evaluation	**CIE**	**40 Marks**	**SEE**		**60 Marks**	

Course Objectives	
1	To learn important concepts of Managerial Economics and apply them to evaluate business decisions
2	To understand various parameters that determine the consumers' behavior.
3	To evaluate the factors that affect production.
4	To understand the concepts of capital budgeting and payback period.
5	To study the concepts of various book-keeping methods.

Course Outcomes On completion of this course, the student will be able to	
CO1	Demonstrate the ability to apply fundamental concepts of Managerial Economics to analyze and assess business decisions, considering economic principles and their implications
CO2	Gain insights into the diverse factors influencing consumers' behavior and decision-making processes, allowing the evaluation of market demand and consumer preferences.
CO3	Assess the multiple determinants impacting production processes, including resource allocation, technology, and costs, enabling effective production management decisions.
CO4	Understand and apply concepts related to capital budgeting, including the computation and analysis of payback periods, aiding in effective investment decision-making.
CO5	Acquire knowledge of various book-keeping methods, comprehending their significance and application in financial record-keeping and analysis for informed business decisions

UNIT – I

Meaning and Nature of Managerial Economics: Managerial Economics and its usefulness to Engineers, Fundamental Concepts of Managerial Economics-Scarcity, Marginalism, Equimarginalism, Opportunity costs, Discounting, Time Perspective, Risk and Uncertainty, Profits, Case study method.

UNIT – II

Consumer Behavior: Law of Demand, Determinants, Types of Demand; Elasticity of Demand (Price, Income and Cross-Elasticity); Demand Forecasting, Law of Supply and Concept of Equilibrium. (Theory questions and small numerical problem can be asked)

UNIT – III

Theory of Production and Markets: Production Function, Law of Variable Proportion, ISO quants, Economics of Scale, Cost of Production (Types and their measurement), Concept of Opportunity Cost, Concept of Revenue, Cost-Output relationship, Break-Even Analysis, Price - Output determination under Perfect Competition and Monopoly (theory and problems can be asked)

UNIT – IV

Capital Management: Significance, determination and estimation of fixed and working capital requirements, sources of capital, Introduction to capital budgeting, methods of payback and discounted cash flow methods with problems. (Theory questions and numerical problems on estimating working capital requirements and evaluation of capital budgeting opportunities can be asked)

UNIT – V
Book-keeping: Principles and significance of double entry book keeping, Journal, Subsidiary books, Ledger accounts, Trial Balance, concept and preparation of Final Accounts with simple adjustments, Analysis and interpretation of Financial Statements through Ratios. (Theory questions and numerical problems on preparation of final accounts, cash book, petty cash book, bank reconciliation statement, calculation of some ratios)

Suggested Reading:

1	Mehta P.L., *Managerial Economics-Analysis, Problems and Cases*, Sulthan Chand & Sons Educational Publishers, 2011
2	Maheswari S.N., *Introduction to Accountancy* , Vikas Publishing House, 2005
3	Pandey I.M., *Financial Management* , Vikas Publishing House, 2009

AI 351 PC	**DATA STRUCTURES LAB**					
Prerequisites			**L**	**T**	**P**	**C**
			0	**0**	**2x2**	**2**
Evaluation	**CIE**	**25 Marks**	**SEE**		**50 Marks**	

Course Objectives	
1	To develop skills to design and analyze simple linear and non linear data structures.
2	To identify and apply the suitable data structure for the given real world problem.
3	To gain knowledge in practical applications of data structures

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

CO 1	Analyze the time and space efficiency of the data structure
CO 2	Identity the appropriate data structure for given problem
CO 3	Implement traversal techniques and sorting methods

List of Experiments

1) Implement the following operations on singly linked list:
i) Creation ii) Insertion iii) Deletion iv) Traversal
2. Implement the following operations on doubly linked list:
i) Creation ii) Insertion iii) Deletion iv) Traversal
3. Implement the following operations on circular linked list:
i) Creation ii) Insertion iii) Deletion iv) Traversal
4. Implementation of Stacks, Queues (using both arrays and linked lists).
5. Implementation of circular queue using arrays.
6. Implementation of double ended queue (de queue) using arrays.
7. Implement a program to evaluate a given postfix expression using stacks.
8. Implement a program to convert a given infix expression to postfix form using stacks.
9. Implementation of Polynomial arithmetic using linked list.
10. Implementation of recursive and non recursive functions to perform the following searching operations for a key value in a given list of integers:
i) Linear search ii) Binary search
11. Implementation of hashing with (a) Separate Chaining and (b) Open addressing methods.
12. Implementation of recursive and iterative traversals on binary tree.
13. Implementation of operations on binary tree (delete entire tree, copy entire tree, mirror image, level order, search for a node etc.)
14. Implementation of the following operations on binary search tree (BST):
(a) Minimum key (b) Maximum key (c) Search for a given key (d) Delete a node with given key
15. Implement the following sorting algorithms:
a) Bubble sort b) Selection sort c) Insertion sort (d) Merge sort (e) Quick sort (f) Heap sort

AI352PC	**DATA SCIENCE WITH PYTHON LAB**					
Prerequisites			**L**	**T**	**P**	**C**
			0	**0**	**2**	**1**
Evaluation	**CIE**	**25 Marks**	**SEE**		**50 Marks**	

Course Objectives	
1	To provide students with a comprehensive understanding of Python programming, covering fundamental concepts, control structures, functions, data manipulation, object-oriented programming, GUI development, data analysis, and practical application of various libraries.

Course Outcomes **On completion of this course, the student will be able to**	
CO1	Develop programs to solve computational problems, manipulate data, and perform basic operations.
CO2	Apply principles of object-oriented programming, including inheritance and polymorphism, to design and implement classes and objects.
CO3	Engage in multivariate analysis, exploratory data analysis, and statistical analysis on complex datasets.

List of Experiments

1. Introduction to Python Programming:
a. Running instructions in Interactive interpreter and a Python Script.
b. Program to purposefully raise Indentation Error and Correct it
c. Program to compute distance between two points taking input from the user
d. Program add.py that takes 2 numbers as command line arguments and prints its sum.
e. Program to display the following information: Your name, Full Address, Mobile Number, College Name, Course Subjects
f. Program for checking whether the given number is an even number or not.
2. Control Structures, Lists
a. Program to find the largest three integers using if-else
b. Program that receives a series of positive numbers and display the numbers in order and their sum
c. Program to display two random numbers that are to be added, the program should allow the student to enter the answer.

- If the answer is correct, a message of congratulations should be displayed.
- If the answer is incorrect, the correct answer should be displayed.

d. Program using a for loop, that prints the decimal equivalents of 1/2, 1/3, 1/4,1/10.
e. Program using a while loop that asks the user for a number, and prints a countdown from that number to zero.
3. Functions and Recursion
Recursive and non-recursive functions for the following
a. To find GCD of two integers
b. To find the factorial of positive integer
c. To print Fibonacci Sequence up to given number n
d. To display prime numbers from 2 to 16
e. Functions that accept a string as an argument and return the number of vowels and consonants that the string contains
4. Files, Exceptions, Lists, Sets, Random Numbers
a. Program to write a series of random numbers in a file from 1 to n and display.
b. Program to write the content in a file and display it with a line number followed by a colon

c. Program to display a list of all unique words in a text file
d. Program to analyse the two text files using set operations
e. Program to print each line of a file in reverse order.
f. Program to count frequency of characters in a given file. Can you use character frequency to tell whether the given file is a Python program file, C program file or a text file?
g. Program combine lists that combines these lists into a dictionary.
5. Object Oriented Programming
a. Program to implement the inheritance
b. Program to implement the polymorphism
6. GUI Programming
a. Program that converts temperature from Celsius to Fahrenheit
b. Program that displays your details when a button is clicked
c. Write a GUI for an Expression Calculator using tk
7. Python data structures:
a. Program to create, apply different functions on lists, dictionaries, tuples and sets.
8. Program for creation and manipulation of Numpy arrays: Element wise Array operations, sorting, copying, subsetting, sliciing, indexing, transposing, changing shape, adding/removing elements, combining arrays, splitting arrays, broadcasting arrays
9. Program to use Scipy.linalg / Numpy.linalg package: Fitting to polynomials, Eigenvectors , Eigenvalues, Decomposing a matrix using SVD, Generating random numbers
10. (a) Program to create pandas Data frames, grouping and joining Data frames, Pandas series, Creating Pivot tables
(b) Using vectorized string functions with Pandas data frames
11. Program to apply pandas data frame related operations on Toyota Corolla dataset (Kaggle) Reading files Exploratory data analysis Data preparation and pre-processing
12. Program to apply multivariate analysis with Titanic data set.
13. Program to perform time series analysis with Open Power System data.
14. Program to predict price of pre-owned cars using Regression – Kaggle data set
15. Program to perform PCA on toy data set.
16. Program to Wine quality data analysis: loading, applying descriptive statistics, finding correlated columns, analysing columns, adding new attributes, grouping columns, concatenating data frames, univariate analysis, multivariate analysis.

AI353PC	JAVA PROGRAMMING LAB					
Prerequisites			**L**	**T**	**P**	**C**
			0	**0**	**2**	**1**
Evaluation	**CIE**	**25 Marks**	**SEE**		**50 Marks**	

Course Objectives	
1	Ability to learn the concept of classes, inheritance and abstract classes.
2	Learn to demonstrate multithreaded programs with synchronization
3	Demonstrate real world applications using java collection frame work and I/O classes.
4	Model Event driven GUI programs using AWT/Swing

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

CO1	Understand the OOPS features
CO2	Understand the usage of abstract classes and interfaces.
CO3	Write multi-threaded programs with synchronization.
C04	Implement real world applications using java collection frame work and I/O classes
CO5	Write Event driven GUI programs using AWT/Swing

List of Experiments

1.A program to illustrate the concept of class with constructors, methods and overloading.
2. A program to illustrate the concept of Inheritance and Dynamic polymorphism.
3. A program to show the concept of packages.
4. A program to illustrate the usage of interfaces and Abstract class.
5. A program to illustrate exception handling keywords.
6. A program to illustrate user define exception using stack.
7. A program to illustrate user define exception for evaluating a post fix expression.
8. A program to illustrate to handle string in java using String and StringBuffer.
9. A program to illustrate manipulating array in java
10. A program to illustrate Multithreading.
11. A program to illustrate Thread synchronization.
12. A program to illustrate inter thread communication.
3. A program using String tokenizer.
14. A program using Linked list class.
15. A program using Tree set class.
16. A program using Hash set and Iterator classes.
17. A program using Map classes.
18. A program using Enumeration and Comparator interfaces.
19. A program to illustrate Buffered I/O streams and Buffered reader.
20. Write a Java program to read text from file from a specify index or skipping byte using file Input stream.
21. Write a Java program to determine number of byte return to file using data output stream.
22. A program to illustrate ByteArrayI/O Streams.
23. A program to illustrate the usage of Serialization.
24. An application involving GUI with different controls, menus and event handling.
25. A program to implement a simple calculator using grid layout manager.
26. A program to implement Recursive Fibonacci method using swing
27. A program to display digital clock using swing

28. A program to read from a file and write to a file using Applet
29. A program to display a calendar using JCombo box.
30. A program to illustrate event listener interfaces.

SCHEME OF INSTRUCTION AND EXAMINATION
B. E (AIML)
SEMESTER- IV

SNo	Code	Course Title	Scheme of Instruction			Contact Hrs/Wk	Scheme of Examination			Credits
			L	T	P		Hrs	CIE	SEE	
Theory										
1	**MT401BS**	Engineering Mathematics – IV (Numerical Methods)	3	0	-		3	40	60	3
2	**AI 401PC**	Artificial Intelligence	3	0	-		3	40	60	3
3	**AI402PC**	Computer Organization and Microprocessors	3	0	-		3	40	60	3
4	**AI 403PC**	Principles of Programming Languages	3	0	-		3	40	60	3
5	**AI 404PC**	Software Engineering	3	0	-		3	40	60	3
6	**ES 402 EC**	Signals and Systems	3	0	-		3	40	60	3
7	**Professional Elective-I**									
	AI 411 PE	Data Visualization	3	0	-		3	40	60	3
	AI 412 PE	Graph Theory								
	AI 413 PE	Microcontrollers and Embedded C								
	AI 414 PE	Operations Research								
	AI 415 PE	Mobile Application Development								
Practicals										
8	**AI451PC**	Artificial Intelligence Lab	-	-	2		3	25	50	1
9	**AI452PC**	Computer Organization and Microprocessors Lab	-	-	2		3	25	50	1
10	**AI453PC**	R Workshop	-	-	2		3	25	50	1
		Total	**21**	**0**	**6**		**30**	**355**	**570**	**24**

MT401BS	**ENGINEERING MATHEMATICS –IV (NUMERICAL METHODS)**					
Prerequisites			**L**	**T**	**P**	**C**
			3	**0**	**-**	**3**
Evaluation	**CIE**	**40 Marks**	**SEE**		**60 Marks**	

Course Objectives :	
1	Apply general methodology to solve linear first order and secondorder partial differential equations
2	To study the classification of second order partial differential equations and solve them by using separation of variables methods
3	To introduce a few numerical methods to solve nonlinear algebraic and transcendental equations and system of linear equations
4	To provide the necessary basic concepts of numerical differentiation, numerical integration
5	To solve the Initial Volume Problems.

Course Outcomes :	
On completion of this course, the student will be able to :	
CO1	Find the solutions of first and second order PDE
CO2	Find solutions of the heat equation, wave equation, and the Laplace equation subject to boundary conditions
CO3	Solve nonlinear equations, system of linear equations
CO4	Find Numerical Integration
CO5	Perform numerical differentiation

UNIT – I

Definition of Partial Differential Equations, First order partial differential equations, Solutions of first order linear PDEs , Solution to homogenous and non-homogenous linear partial differential equations of second order by complimentary function and particular integral method.

UNIT – II

Second-order linear equations and their classification, Initial and boundary conditions, D'Alembert's solution of the wave equation, Heat diffusion and vibration problems, Separation of variables method to Solve simple problems in Cartesian coordinates. The one dimensional diffusion equation and its solution by separation of variables.

UNIT – III

Bisection method, Newton-Raphson method, Solution of linear system of equations- Gauss elimination method, LU decomposition method, Gauss-Jacobi and Gauss-Seidel iteration methods.

UNIT – IV

Interpolation, Lagrange’s interpolation, Newton’s divided difference interpolation, Newton’s Forward and Backward difference interpolations. Numerical differentiation, Interpolation approach, Numerical integration-Trapezoidal rule,Simpson’s 1/3 rule.

UNIT – V

Taylor’s series method, Euler’s method, Picard’s method of successive approximations, Runge-Kutta method of 4th order

Suggested Reading:

1	R. K. Jain & S.R.K Iyengar, Advanced Engineering Mathematics, Narosa Publications, 4th Edition 2014 (Text Book).
2	Erwin Kreyszi, Advanced Engineering Mathematics, John Wiley, 9th Edition, 2012.
3	B.S. Grewal, Higher Engineering Mathematics, Khanna Publications, 43rd Edition, 2014.
4	M.K.Jain, S.R.K.Iyengar and R.K.Jain, Numerical methods for scientific and engineeringcomputation ,6th Edition , New Age International Limited., 2012
5	B .V . Ramana, Higher Engineering Mathematics, 23rd reprint, 2015.
6	S.S.Sastry, Introductory Methods of Numerical Analysis, 5th Edition , PHI Private Limited,2012.
7	H.K. Dass, Er. Rajnish Varma, higher Engineering Mathematics, S.Chand Technical 3rd Edition, 2011.

AI401PC	**ARTIFICAL INTELLIGENCE**					
Prerequisites	**Introduction to Data Science**		**L**	**T**	**P**	**C**
			3	**0**	**-**	**3**
Evaluation	**CIE**	**40 Marks**	**SEE**		**60 Marks**	

Course Objectives	
1	To become familiar with basic principles of AI toward problem solving, inference, perception, knowledge representation, and learning.
2	To Investigate applications of AI techniques in intelligent agents, expert systems, artificial neural networks and other machine learning models.
3	Explore the current scope, potential, limitations, and implications of intelligent systems

Course Outcomes On completion of this course, the student will be able to	
CO1	Apply basic principles of AI in solutions that require problem solving, inference, perception, knowledge representation, and learning.
CO2	Demonstrate awareness and a fundamental understanding of various applications of AI techniques in intelligent agents, expert systems, artificial neural networks and other machine learning models.
CO3	Demonstrate an ability to share in discussions of AI, its current scope and limitations, and societal implications.

UNIT – I

Introduction to Artificial Intelligence: Introduction, Brief History, Intelligent Systems, foundations of AI, Sub-Areas of AI, Applications, Tic-Tac Game Playing, Development of AI Languages, Current Trends in AI.

Agents: Agents and Environments, Good Behavior: The concept of Rationality, Performance measures, The nature of Environments, The Structure of Agents, Simple agents, Rational agents, problem solving agents, intelligent agents.

UNIT – II

Solving Problem by Searching: Problem-Solving Agents, Searching for Solutions, Uninformed search strategies.

Informed Search and Exploration: Informed Search Strategies, Heuristic Functions, Local-Search Algorithms and Optimization Problems.

Adversarial Search: Games, Optimal Decisions in Games, Alpha-Beta Pruning, Iterative Deepening.

UNIT – III

Logic Concepts and Logic Programming: Introduction, Propositional Calculus, Propositional Logic, Natural Deduction System, Axiomatic System, Predicate Logic, Logic Programming.

Knowledge Representation: Introduction, Approaches to Knowledge Representation, Knowledge Representation using Semantic Network, Knowledge Representation using Frames.

UNIT – IV

Probabilistic Reasoning

Expert System: Introduction, Phases in Building Expert Systems, Expert System Architecture, Expert System versus Traditional Systems, Rule-Based Expert Systems.

Uncertainty Measures: Introduction, Probability Theory, Bayesian Belief Networks.

Fuggy Logic Systems: Introduction, Crisp Sets, Fuzzy Sets, Fuzzy Terminology, Fuggy Logic Control, Neuro Fuzzy Systems.

UNIT – V

Connectionist Models: Introduction: Hopfield Networks, learning in Neural Networks, Applications of Neural Networks, Recurrent Networks, Distributed Representations, Connectionist AI and Symbolic AI. **Chatbots**

Suggested Reading:

1	"*Artificial Intelligence*" Saroj Kaushik, Cengage Learning, 2011
2	"*Artificial Intelligence*" Elaine Rich, Kevin Knight, Shivashankar B Nair, Tata McGraw Hill, Third Edition, 2019
3	"*Artificial Intelligence-A Modern Approach*" Second Edition, Stuart Russell, Peter Norvig. Create Space Independent Publishing Platform, 2016

AI402PC	**COMPUTER ORGANIZATION AND MICROPROCESSORS**					
Prerequisites	**Digital Logic Design**		**L**	**T**	**P**	**C**
			3	**0**	**-**	**3**
Evaluation	**CIE**	**40 Marks**	**SEE**		**60 Marks**	

Course Objectives	
1	To understand the Instruction Set Architecture: Instruction format, types, various addressing modes
2	To understand the basic components and design of the CPU: the ALU and control unit .
3	To understand the parallelism both in terms of a single processor and multiple processors
4	To understand the 8085 and 8051 architecture
5	To learn the interfacing with I/O Organization, Interrupt-driven I/O, and DMA

Course Outcomes On completion of this course, the student will be able to	
CO1	Understand the Instruction Set Architecture: Instruction format, types, various addressing modes
CO2	Analyze the components and design of the CPU: the ALU and control unit write multi threaded programs with synchronization
CO3	Apply the parallelism in terms of a single processor and multiple processors environment
CO4	Analyze the 8085 and 8051 architectures
CO5	Apply interfacing with I/O Organization, Interrupt-driven I/O, DMA

UNIT – I

Data Representation: Fixed and Floating Point representations. Overview of Computer Function and Interconnections: Computer components, Interconnection structures, Bus interconnection, Bus structure, and Data transfer. Register Transfer Micro operations: Register Transfer Language, Register Transfer, Bus and Memory Transfers, Arithmetic, Logic and Shift micro-operations, Arithmetic Logic Shift Unit.

UNIT – II

Basic Computer Organization and Design: Instruction Codes, Computer Registers, Computer Instructions, Timing and Control, Instruction Cycle, Memory reference instruction, Input-Output and Interrupt. Microprogrammed Control: Control memory, Address Sequencing, Microprogram example, Design of Control Unit.

UNIT – III

Central Processing Unit: General Register Organization, Stack Organization, Instruction formats, Addressing modes, Data Transfer and Manipulation, and Program control. Floating Point Arithmetic Operations. Pipeline Processing: Arithmetic, Instruction and RISC Pipelines.
Memory Organization: Cache memory, Virtual memory, Memory Management
hardware

UNIT – IV

8085 Architecture: Introduction to microprocessors and microcontrollers, 8085 Processor Architecture, Internal operations, Instructions and timings. Programming the 8085 - Introduction to 8085 instructions, Addressing modes and Programming techniques with Additional instructions. Input-Output Organization: Modes of Transfer, Priority Interrupt, Direct Memory
Access (DMA), I/O Processor. Basic Interfacing concepts with 8085, Programmable Interrupt Controller (8259A). Direct Memory Access(DMA) - DMA Controller (Intel 8257

UNIT – V
Introduction to Microcontrollers, 8051 – Architecture, Instruction set, Addressing modes and Programming techniques. Comparison of various families of 8-bit micro controllers. System Design Techniques - Interfacing of LCD, ADC, Sensors, Stepper motor, Keyboard and DAC using microcontrollers. Communication Standards - Serial RS 232 and USB. Features of Multi-Core Processors architectures and Graphics Processing Unit.

Suggested Reading:

1	Morris Mano M “Computer System Architecture”, 3rd Edition, Pearson Education India, 2007.
2	William Stallings “Computer Organization and Architecture”, PHI, 7th Edition, 2008.
3	Ramesh S. Gaonkar “Microprocessor Architecture, Programming, and Applications with 8085”, 5/E, Prentice Hall, 2002.
4	Myke Predko “Programming and Customizing the 8051 Microcontroller”, Tata McGraw Hill, 1997

AI 403 PC	**PRINCIPLES OF PROGRAMMING LANGUAGES**					
Prerequisites	**Programming for Problem Solving**		**L**	**T**	**P**	**C**
			3	**0**	**-**	**3**
Evaluation	**CIE**	**40 Marks**	**SEE**		**60 Marks**	

Course Objectives	
1	To introduce the major programming paradigms, and the principles and techniques involved in design and implementation of modern programming languages
2	To introduce notations to describe syntax and semantics of programming languages.
3	To analyze and explain behavior of simple programs in imperative languages using concepts such as binding, scope, control structures, subprograms and parameter passing mechanisms.
4	To introduce the concepts of ADT and object oriented programming for large scale software development
5	To introduce the concepts of concurrency control and exception handling.

Course Outcomes **On completion of this course, the student will be able to**	
CO1	Understand the programming paradigms of modern programming language
CO2	Describe syntax and semantics of programming languages
CO3	Analyze the behavior of simple programs in imperative languages.
CO4	Understand the concepts of ADT and object oriented programming for large scale software development
CO5	Understand the concepts of functional programming and logic programming

UNIT – I

Preliminary Concepts: Reasons for Studying Concepts of Programming Languages, Programming Domains, Language Evaluation Criteria, Influences on Language Design, Language Categories, Language Design Trade-offs, Implementation Methods, Programming Environments, Evolution of the Major Programming Languages.

Describing Syntax and Semantics: General Problem of Describing Syntax, Formal Methods of Describing Syntax, Attribute Grammars, Describing the Meaning of Programs.

UNIT – II

Names, Binding, Type Checking, and Scopes: Names, Variables, The Concept of Binding, Type Checking, Strong Typing, Type Compatibility, Scope, Scope and Lifetime, Referencing Environments, Named Constants.

Data Types: Primitive Data Types, Character String Types, User- Defined Ordinal Types, Array Types , Associative Arrays, Record Types, Union Types, Pointer and Reference Types, optional types

Expressions and Assignment Statements: Arithmetic Expressions, Overloaded Operators, Type Conversions, Relational and Boolean Expressions, Short-Circuit Evaluation, Assignment Statements, Mixed- Mode Assignment

UNIT – III

Statement-Level Control Structures: Selection Statements, Iterative Statements, Unconditional Branching, Guarded Commands.

Subprograms: Fundamentals and Design Issues for Subprograms, Local Referencing Environments, Parameter –Passing Methods, Parameters That are Subprograms Names, Overloaded Subprograms, Generic Subprograms, Design Issues for Functions, User-Defined Overloaded Operators.

Implementing Subprograms: The General Semantics of Calls and Returns, Implementing “Simple” Subprograms, Implementing Subprograms with Stack-Dynamic Local Variables, Nested Subprograms, Blocks, Implementing Dynamic Scoping.

Abstract Data Types: The Concept of Abstraction, Introduction to Data Abstraction, Design Issues for Abstract Data Types, Language Examples, Parameterized ADT, Encapsulation Constructs, Naming Encapsulation.

UNIT – IV

Object Oriented Programming: Design Issues, Object Oriented Programming in Smalltalk, C++, Java, C#, Ada 95, Ruby, The Object Model of JavaScript, Implementation of Object Oriented Constructs. **Concurrency:** Subprogram level Concurrency, Semaphores, Monitors, Massage Passing, Ada Support for Concurrency, Java Threads, C# Threads, Statement-Level Concurrency. Exception Handling and **Event Handling:** Introduction to Exception Handling, Exception Handling in Ada, C++ and Java, Introduction to Event Handling, Event Handling with Java.

UNIT – V

Functional Programming Languages: Introduction, Mathematical Functions, Fundamentals of FPL, LISP, Introduction to Scheme, COMMON LISP,ML, Haskell, Application of Functional Programming Languages and A Comparison of Functional and Imperative Languages, Functional interfaces (Java 8.0)

Logic Programming Languages: Introduction to Predicate Calculus, Predicate Calculus and Proving Theorems, An Overview of Logic Programming. The Origins, Basic Elements and Deficiencies of Prolog, Applications of Logic Programming.

Scripting Languages: Common Characteristics, Data Types , Object Orientation Names and Scopes, String and Pattern Manipulation Problem Domains, Scripting the World Wide Web

Suggested Reading:

1	"*Concepts of Programming Languages*" Robert.W.Sebesta 12th Edition, Pearson Education, 2019
2	"*Programming Language Pragmatics* " Michal Scott 4th Edition Morgan Kaufmann Publishers, 2015
3	*Java Precisely* Peter Sestoft 3rd Edition, MIT press 2016
4	*Programming Languages: Principles & Practices* Kenneth A. Lambert and Kenneth C. Louden, 3rd Edition, Cengage Learning 2012.
5	*Programming languages* , Watt, Wiley Dreamtech, First Edition , 2004.

AI 404 PC	**SOFTWARE ENGINEERING**					
Prerequisites			**L**	**T**	**P**	**C**
			3	**0**	**-**	**3**
Evaluation	**CIE**	**40 Marks**	**SEE**		**60 Marks**	

Course Objectives	
1	To introduce the basic concepts of software development- processes from defining a product to shipping and maintaining that product
2	To impart knowledge on various phases , methodologies and practices of software development
3	To understand the importance of testing in software development and study various testing strategies and software quality metrics.

Course Outcomes On completion of this course, the student will be able to	
CO1	Acquire working knowledge of alternative approaches and techniques for each phase of software development
CO2	Acquire skills necessary for independently developing a complete software project
CO3	Understand the practical challenges associated with the development of a significant software system.

UNIT – I: Introduction to Software Engineering:

A generic view of Process: Software Engineering, Process Framework, CMM Process Patterns, Process Assessment.

Process Models: Prescriptive Models, Waterfall Model, Incremental Process Models, Evolutionary Process Models, Specialized Process Models, The Unified Models, Personal and Team Process Models, Process Technology, Product and Process.

An Agile view of Process: Introduction to Agility and Agile Process, Agile Process Models.

UNIT – II

Software Engineering Principles: SE Principles, Communication Principles, Planning Principles, Modeling Principles, Construction Principles, Deployment.System Engineering: Computer-based Systems, The System Engineering Hierarchy, Business Process Engineering, Product Engineering, System Modeling.

Requirements Engineering: A Bridge to Design and Construction, Requirements Engineering Tasks, Initiating Requirements Engineering Process, Eliciting Requirements, Developing Use-Cases, Building the Analysis Model, Negotiating Requirements, Validating Requirements.

UNIT – III

Building the Analysis Model: Requirements Analysis Modeling Approaches, Data Modeling Concepts, Object-Oriented Analysis, Scenario-based Modeling, Flow-oriented Modeling, Class based Modeling, Creating a Behavioral Model.

Design Engineering: Design within the context of SE, Design Process and Design Quality, Design Concepts, The Design Model, Pattern-based Software Design..

UNIT – IV

Creating an Architectural Design: Software Architecture, Data Design, Architectural Styles and Patterns, Architectural Design, Assessing Alternative Architectural Designs, Mapping Data Flow into a Software Architecture.

Modeling Component-Level Design: Definition of Component, Designing Class-based Components, Conducting Component-level Design, Object Constraint Language, Designing Conventional Components.

Performing User Interface Design: The Golden Rules, User Interface Analysis and Design, Interface Analysis, Interface Design Steps, Design Evaluation.

UNIT – V

Software Quality Assurance: Basic Elements, Tasks, Goals and Metrics, Formal Approaches, Statistical Software Quality Assurance, Software Reliability, ISO 9000 Quality Standards, SQA Plan.

Testing Strategies: A Strategic Approach to Software Testing, Strategic Issues, Test Strategies for O-O Software, Validation Testing, System Testing, The Art of Debugging.

Testing Tactics: Software Testing Fundamentals, Black-box and White-box Testing, Basis Path Testing, Control Structure Testing, O-O Testing Methods, Testing Methods applicable on the Class Level, Inter Class Test Case Design, Testing for Specialized Environments, Architectures and Applications, Testing Patterns.

Product Metrics: Software Quality, A Framework for Product Metrics, Metrics for the Analysis Model, Metrics for the Design Model, Metrics for Source Code, Metrics for Testing, Metrics for Maintenance.

Suggested Reading:

1	Roger S.Pressman, *Software Enigneering: A Practitioner"s Approach*, 7th Edition, McGraw Hill, 2009.
2	Ali Behforooz and Frederick J.Hudson, *Software Engineering Fundamentals*, Oxford University Press, 1996.
3	Pankaj Jalote , *An Integrated Approach to Software Engineering*, 3rd Edition, Narosa Publishing House, 2008.

EC 402 ES	SIGNALS AND SYSTEMS					
Prerequisites	**Basic Electronics**		**L**	**T**	**P**	**C**
			3	**0**	**-**	**3**
Evaluation	**CIE**	**40 Marks**	**SEE**		**60 Marks**	

Course Objectives	
1	To explain signals and systems representations/classifications and also describe the time and frequency domain analysis of continuous time signals with Fourier series, Fourier transforms and Laplace transforms
2	To understand Sampling theorem, with time and frequency domain analysis of discrete time signals with DTFS, DTFT and Z-Transform
3	To present the concepts of convolution and correlation integrals and also understand the properties inthe context of signals/systems and lay down the foundation for advanced courses.
4	To Understand Fourier Analysis of discrete-time signals, including the periodic signal representation using the Discrete-Time Fourier Series.
5	To Understand system realization in the Z-Transform domain.

Course Outcomes On completion of this course, the student will be able to	
CO1	Apply the knowledge of linear algebra topics like vector space, basis, dimension, inner product, normand orthogonal basis to signals.
CO2	Analyze the spectral characteristics of continuous-time periodic and a periodic signals using Fourieranalysis.
CO3	Understand the process of sampling and the effects of under sampling. Classify systems based ontheir properties and determine the response of LSI system using convolution.
CO4	Analyze system properties based on impulse response and Fourier analysis.
CO5	Apply the Laplace transform and Z- transform for analyze of continuous-time and discrete-timesignals and systems

UNIT – I

Some useful operations on signals: Time shifting, Time scaling, Time inversion. Signal models: Impulse function, Unit step function, Exponential function, Even and odd signals. Systems: Linear and Non-linear systems, Constant parameter and time varying parameter systems, Static and dynamic systems, Causal and Non-causal systems, Lumped Parameter and distributed parameter systems, Continuous-time and discrete-time systems, Analog and digital systems

UNIT – II

Fourier Series: Signals and Vectors, Signal Comparison: correlation, Signal representation by orthogonal signal set, Trigonometric Fourier Series, Exponential Fourier Series, LTI system response to periodic inputs.

UNIT – III

Continuous-Time Signal Analysis: Fourier Transform: A periodic signal representation by Fourier integral, Fourier Transform of some useful functions, Properties of Fourier Transform, Signal transmission through LTI Systems, ideal and practical filters, Signal energy. Laplace transform: Definition, some properties of Laplace transform, solution of differential equations using Laplace transform.

UNIT – IV
Discrete-time Signals and Systems : Introduction, some useful discrete-time signal models, Sampling continuous-time sinusoids and aliasing, Useful signal operations, examples of discrete-time systems. Fourier Analysis of discrete-time signals, periodic signal representation of discrete-time Fourier Series, aperiodic signal representation by Fourier integral.

UNIT – V
Discrete-time Signal Analysis : Z-Transform, some properties of Z-Transform, Solution to Linear difference equations using Z transform, System realization. Relation between Laplace transform and Z transform. DTFT: Definition, Properties of DTFT, comparison of continuous-time signal analysis with discrete-time signal analysis.

Suggested Reading:

1	B. P. Lathi, Linear Systems and Signals, Oxford University Press, 2nd Edition, 2009
2	Alan V O P Penheim, A. S. Wlisky , Signals and Systems, 2nd Edition, Prentice Hall.
3	Rodger E. Ziemer, William H Trenter, D. Ronald Fannin, Signals and Systems, 4th Edition, Pearson1998.
4	Douglas K. Linder, Introduction to Signals and Systems, McGraw Hill, 1999
5	P. Ramakrishna Rao, Signals and Systems, TMH, 2013.

PROFESSIONAL ELECTIVE-I

AI 411 PE	**DATA VISUALIZATION**					
Prerequisites	**Introduction to Data Science**		**L**	**T**	**P**	**C**
			3	**0**	**-**	**3**
Evaluation	**CIE**	**40 Marks**	**SEE**		**60 Marks**	

Course Objectives	
1	Gain a deep understanding of the key principles and components of data visualization, including the importance of conviction.
2	Acquire the skills needed to work with diverse types of data, including data acquisition, examination, transformation, and exploration, enhancing data literacy and analytical capabilities
3	Learn various visualization techniques for representing amounts, distributions, proportions, x-y relationships, geospatial data, and uncertainty, utilizing tools like bar plots, heatmaps,histograms

Course Outcomes

On completion of this course, the student will be able to

CO1	Apply the principles of trustworthy, accessible, and elegant data visualization to real-world scenarios
CO2	Skillfully navigate data acquisition, examination, and transformation processes to uncover hidden patterns and insights
CO3	Demonstrate proficiency in creating a wide range of visualizations, including bar plots, pie charts, scatterplots, time series plots
CO4	Showcase the ability to translate complex data into meaningful visual stories, utilizing annotation, color selection.
CO5	Develop interactive visualizations using libraries like Seaborn and Bokeh, employing interactivity features.

UNIT – I

Defining Data Visualization:
The Components of Understanding, The Importance of Conviction, Principle 1: Good Data Visualization is Trustworthy, Principle 2: Good Data Visualization is Accessible, Principle 3: Good Data Visualization is Elegant, Distinctions and Glossary.
Visualization Workflow: The Importance of Process, The Process in Practice.
Formulating Your Brief: What is a Brief? Establishing Your Project's Context, Establishing Your Project's Vision.

UNIT – II

Working with data: Data Literacy, Data Acquisition, Data Examination, Data Transformation, Data Exploration.
Establishing Your Editorial Thinking: What is Editorial Thinking? The Influence of Editorial Thinking.
Directory of Visualizations: Amounts, Distributions, Proportions, x–y relationships, Geospatial Data, uncertainty.
Visualizing Amounts: Bar Plots, Grouped and Stacked Bars, Dot Plots and Heatmaps.
Visualizing Distributions: Histograms and Density Plots: Visualizing a Single Distribution, Visualizing Multiple Distributions at the Same Time.
Visualizing Distributions: Empirical Cumulative Distribution Functions and Q-Q Plots: Empirical Cumulative Distribution Function, Highly Skewed Distributions, Quantile-Quantile Plots.
Visualizing Many Distributions at Once: Visualizing Distributions Along the Vertical Axis, Visualizing Distributions Along the Horizontal Axis.

UNIT – III
Visualizing Proportions: A Case for Pie Charts, A Case for Side-by-Side Bars, A Case for Stacked Bars and Stacked Densities, Visualizing Proportions Separately as Parts of the Total. **Visualizing Nested Proportions:**Nested Proportions Gone Wrong, Mosaic Plots and Treemaps, Nested Pies, Parallel Sets. **Visualizing Associations Among Two or More Quantitative Variables:** Scatterplots, Correlograms, Dimension Reduction, Paired Data. **Visualizing Time Series and Other Functions of an Independent Variable:** Individual Time Series, Multiple Time Series and Dose–Response Curves, Time Series of Two or More Response Variables. **Visualizing Trends:** Smoothing, Showing Trends with a Defined Functional Form, Detrending and Time-Series Decomposition. **Visualizing Geospatial Data:** Projections, Layers, Choropleth Mapping, Cartograms. **Visualizing Uncertainty:** Framing Probabilities as Frequencies, Visualizing the Uncertainty of Point Estimates, Visualizing the Uncertainty of Curve Fits, Hypothetical Outcome Plots.

UNIT – IV
Interactivity: Features of Interactivity: Data Adjustments, Features of Interactivity: Presentation Adjustments, Influencing Factors and Considerations. **Annotation:** Features of Annotation: Project Annotation, Features of Annotation: Chart Annotation, Typography, Influencing Factors and Considerations. **Colour:** Overview of Colour Theory**,** Features of Colour: Data Legibility, Features of Colour: Editorial Salience, Features of Colour: Functional Harmony, Influencing Factors and Considerations. **Composition:** Features of Composition: Project Composition**,** Features of Composition: Chart Composition, Influencing Factors and Considerations.

UNIT – V
Data Visualization: Visualization using Matplotlib: Accessories for charts Scatter plot, Line plot, Pie plot, Bar plot, Histogram plot, Bubble plot 146, pandas plotting, **Advanced visualization using the Seaborn package:** lm plots, Bar plots, Distribution plots, Box plots, KDE plots, Violin plots, Count plots, Joint plots, Heatmaps, Pair plots, **Interactive visualization with Bokeh:** Plotting a simple graph, Glyphs, Layouts, Nested layout using row and column layouts, Multiple plots, Interactions, Hide click policy, Mute click policy, Annotations, Hover tool, Widgets, Tab panel, Slider

Suggested Reading:

1	"Data visualization A hand book for data driven design " Andy krick Sage publication limited, 2019
2	"Fundamentals of data visualization A primer on making informative and compelling figures " Claus o. wilke, O'Reilly Media, 2019.
3	Python Data Analysis: Perform data collection, data processing, wrangling, visualization, and model building using Python, 3^{rd} Edition Avinash Navlani , Armando Fandango , Ivan Idris, , Packt Publishers, 2021

AI 412 PE	GRAPH THEORY					
Prerequisites	**Data Structures**		**L**	**T**	**P**	**C**
			3	**0**	**-**	**3**
Evaluation	**CIE**	**40 Marks**	**SEE**		**60 Marks**	

Course Objectives	
1	To familiarize a variety of different problems in Graph Theory
2	To learn various techniques to prove theorems
3	To understand and analyse various graph algorithms

Course Outcomes On completion of this course, the student will be able to	
CO1	Write precise and accurate mathematical definitions of objects in graph theory
CO2	Validate and critically assess a mathematical proof
CO3	Develop algorithms based on diverse applications of Graphs in different domains

UNIT – I

Preliminaries: Graphs, isomorphism, subgraphs, matrix representations, degree, operations on graphs, degree sequences Connected graphs and shortest paths: Walks, trails, paths, connected graphs, distance, cut vertices, cut-edges, blocks, connectivity, weighted graphs, shortest path algorithms Trees: Characterizations, number of trees, minimum spanning trees.

UNIT – II

Special classes of graphs: Bipartite graphs, line graphs, chordal graphs Eulerian graphs: Characterization, Fleury‟s algorithm, chinese-postman-problem.

UNIT – III

Hamilton graphs: Necessary conditions and sufficient conditions Independent sets, coverings, matchings: Basic equations, matchings in bipartite graphs, perfect matchings, greedy and approximation algorithms.

UNIT – IV

Vertex colorings: Chromatic number and cliques, greedy coloring algorithm, coloring of chordal graphs, Brook‟s theorem

Edge colorings: Gupta-Vizing theorem, Class-1 graphs and class-2 graphs, equitable edge-coloring.

UNIT – V

Planar graphs: Basic concepts, Eulers formula, polyhedrons and planar graphs, characterizations, planarity testing, 5-color-theorem.

Directed graphs: Out-degree, in-degree, connectivity, orientation, Eulerian directed graphs, Hamilton directed graphs, tournaments

Suggested Reading:

1	F.Harry, Graph theory, Narosa Publications, 1988.
2	C.Berge: Graphs and Hypergraphs, North Holland/Elsevier, 1973
3	J A Bondy and U.S. R Murthy, Graph Theory with Applications, Elsevier Science Ltd, 1976.
4	Douglas B West, Introduction to Graph Theory, Prentice Hall, 2004.

AI 413 PE	**MICROCONTROLLERS AND EMBEDDED C**					
Prerequisites			**L**	**T**	**P**	**C**
			3	**0**	**-**	**3**
Evaluation	**CIE**	**40 Marks**	**SEE**		**60 Marks**	

Course Objectives	
1	To learn the architecture and programming of typical microcontroller.
2	To introduce the basic concepts of small and medium scale embedded system design using microcontroller
3	Collect knowledge of architecture of ARM 7processor
4	Understand the peripherals of LPC2148 microcontroller
5	Learn to design, construct, program, verify, analyze and troubleshoot 8051 and LPC2148b ARM controller C language programs and supporting hardware.

Course Outcomes

On completion of this course, the student will be able to

CO1	Describe the importance and function of each pin of AVR ATmega32 Microcontroller
CO2	Develop embedded C language programs for AVR Microcontroller
CO3	Able to explain the architecture and programming model of ARM 7TDMI
CO4	Ability to undertake problem identification, formulation and selection of an appropriate Microcontroller
CO5	Ability to design and build functional prototype for real world applications

UNIT – I

The Microcontroller AVR: Role of microcontrollers in embedded Systems. Overview of the AVR family, Architecture and instruction set of 8-bit AVR Microcontroller: AVR Microcontroller architecture: Registers, AVR status register, Memory Space, ATmega32 pin-configuration &amp; function of each pin, Addressing mode and instruction set of AVR microcontroller, Data transfer, Arithmetic, Logic and Compare, Rotate and Shift, Branch and Call instructions, Bit manipulation instructions.

UNIT – II

Embedded C Programming for AVR: AVR data types and assembler directives, AVR assembly language programs, AVR I/O Port Programming, Time delay loop, Simple C programs for general purpose I/O and bit addressability, Timers, Interrupts, serial port, Serial port Interfacing protocols, SPI, I2C, UART. C Language programming for peripherals.

UNIT – III

ARM Architecture: Difference between RISC and CISC, RISC Design philosophy, ARM Design Philosophy, History of ARM microprocessor, ARM processor family. The Acorn RISC Machine, ARM Core data flow model, Architectural inheritance, The ARM7TDMI programmer’s model: General purpose registers, CPSR, SPSR, ARM memory map, data format, load and store architecture, ARM states, Interrupts and Exceptions.

UNIT – IV

LPC2148 ARM CPU and embedded C Programming for ARM: Salient features, applications, block diagram, memory mapping. Functional features of Interrupt controller, RTC, USB, UART, I2C, SPI, SSP controllers, watch dog timers and other system control units. C programs for General purpose I/O, general purpose timer, PWM Modulator, UART, I2C Interface, SPI Interface, ADC, DAC

UNIT – V
LPC 2148–Peripherals: Pin Connect Block- Features, Register description with example. GPIO-Features, Applications, Pin description, Register description with examples. PLL-Features, block diagram, bit structure of PLLCON, PLLCFG, &amp; PLLSTAT,and PLLFEED.PLL frequency Calculation- procedure for determining PLL settings, examples for PLL Configuration Timers-Features, applications, Architecture of timer module, register description, Simple C programs for application using -GPIO, PLL, Timer.

Suggested Reading:

1	Dhananjay Gadre, “Programming and Customizing the AVR Microcontroller”, TMH, 1st Edition, 2001.
2	Andrew N. SLOSS , “ARM System Developer’s guide-Designing and optimizing system software , Elsevier Publications, 2016.
3	Atul P. Godse, “ARM controller”, First Edition, Technical Publications, 2020.

AI 414 PE	**OPERATIONS RESEARCH**					
Prerequisites			**L**	**T**	**P**	**C**
			3	**0**	**-**	**3**
Evaluation	**CIE**	**40 Marks**	**SEE**		**60 Marks**	

Course Objectives	
1	To give an overview of different Optimization Techniques useful for problem solving and decision making.

Course Outcomes **On completion of this course, the student will be able to**	
CO1	The objective of the course is to give an overview of different Optimization Techniques useful for problem solving and decision making.
CO2	Solve decision making situation problem using the concept of linear programming techniques.
CO3	Solve transport related problems of Industry.
CO4	Solve the problems related to assignment of jobs or projects to the employees in IT and Management related, which minimizes the total assignment cost.

UNIT – I

Introduction

Introduction to OR- Origin, Nature, definitions, Managerial applications and limitations of OR.

Linear Programming: Mathematical model, Formulation of LPP, assumptions underlying LPP, Solution by the Graph, Exceptional cases.

UNIT – II

Allocation Model - I

LPP - Simplex Method- Solution to LPP problems Maximisation and Minimisation cases Optimality conditions. Degeneracy.

Dual - Formulation**,** Relationship between Primal - Dual, Solution of dual, Economic interpretation of dual.

UNIT – III

Allocation Model - II

Transportation Problem (TP) - Mathematical model, IBFS using northwest corner rule, Row and Column Minimum methods, Matrix minimum method(LCM) and Vogel's approximation method, Unbalanced TP, Degeneracy, Optimality Test and Managerial applications

UNIT – IV

Allocation Model – III

Assignment Problem (AP): Mathematical model, Unbalanced AP, Restricted AP, method of obtaining solution- Hungarian method.

Travelling salesman problem

UNIT – V

Competitive Strategy Models

Game Theory- concepts, saddle point, Dominance, Zero-sum game, two, three and more Persons games, analytical method of solving two person zero sum games, graphical solutions for (m x 2) and (2 x n) games.

Suggested Reading:

1	J.K. Sharma, “Operations Research Theory and Applications 2009, 4th Ed. Macmillan.
2	S.D.Sharma, “Operations Research” , Publishing 2017, Latest Edition, Kedar Nath Ram Nath.
3	S.D.Sharma, “Operations Research” , Publishing 2017, Latest Edition, Kedar Nath Ram Nath.
4	Kasana, HS & Kumar, KD, “Introductory Operations Research theory and applications”, 2008, Springer.
5	Chakravarty, P, “Quantitative Methods for Management and Economics”, 2009, 1st Ed. HPH.

AI 415 PE	**MOBILE APPLICATION DEVELOPMENT**					
Prerequisites			**L**	**T**	**P**	**C**
			3	**0**	**-**	**3**
Evaluation	**CIE**	**40 Marks**	**SEE**		**60 Marks**	

Course Objectives	
1	To impart knowledge on Android OS design and Features.
2	To know the android application components, user interface.
3	To analyse persistent storages.

Course Outcomes On completion of this course, the student will be able to	
CO1	Understand the basics of Android devices and Platform.
CO2	Acquire knowledge on basic building blocks of Android programming required for App development.
CO3	Understand persistence Data storage mechanism in Android
CO4	Understand advanced application concepts like Animations and Google Maps services.

UNIT – I

Introduction: Introduction to mobile application development, **Android platform**: Android platform features and architecture, versions, comparison added features in each version,ART (Android Runtime), ADB (Android Debug Bridge).

Development environment/IDE: Android studio and its working environment, gradle build system, emulator setup.

Application anatomy: Application framework basics: resources, layout, values, asset XML representation and generate R.Javafile, Android manifest file, creating a simple application

UNIT – II

ANDROID UI DESIGN

GUI for Android: Introduction to activities, activities life-cycle, **Intent: intent** object, intent filters, adding categories, linking activities, **Views and View Groups**: Basic views, picker views, adapter views, Menu, App Bar etc, basics of screen design; different layouts. App widgets.

Lollipop Material design: new themes, new widgets, Card layouts. Recycle View **Fragments**: Introduction to activities, activities life-cycle.

UNIT – III

DATA PERSISTENCE

Different Data persistence schemes: Shared preferences, File Handling, Managing data using Files – Using application specific folders and files, creating files, reading data from files, listing contents of a directory Shared Preferences – Creating shared preferences, saving and retrieving data using Shared Preference.

Content providers: User content provider, Android in build content providers

UNIT – IV

BACKGROUND RUNNING PROCESS, NETWORKING AND TELEPHONY SERVICES

Services: introduction to services – local service, remote service and binding the service, the communication between service and activity, Intent Service.

Multithreading: Handlers, Async Task

Broadcast receivers: Local Broadcast Manager, Dynamic broadcast receiver, System Broadcast. Pending Intent, Notifications

Telephony Manager: Sending SMS and making calls.

UNIT – V

ADVANCED APPLICATIONS

Location based services: Displaying Maps, Obtaining the Maps API Key , Displaying the Zoom Control, Changing Views, Navigating to a specificLocation Getting the Location that was Touched Geocoding and Reverse Geocoding Getting location data and Monitoring a Location.

Mobile Application Development for iOS

Introduction, iOS Navigation and Interface Design, Persistent Data in iOS, Tables in iOS:Navigation and Information Display.

Suggested Reading:

1	Dawn Griffiths, David Griffiths, *“Head First Android Development” ,OReilly 2015.*
2	J.F. DiMarzio’s, “ Practical Android 4 Games Application Development”, Apress, 2011
3	Professional Android 4 Application Development, Reto Meier, Wiley India, 2012.

AI 451PC	**ARTIFICAL INTELLIGENCE LAB**					
Prerequisites			**L**	**T**	**P**	**C**
			0	**0**	**2**	**1**
Evaluation	**CIE**	**25 Marks**	**SEE**		**50 Marks**	

Course Objectives	
1	Develop the ability to design and implement solutions for both informed and uninformed search problems in Artificial Intelligence (AI)
2	Acquire proficiency in using Prolog to express and reason about knowledge in first-order logic
3	Utilize the Natural Language Toolkit (NLTK) and advanced techniques to implement Natural Language Processing (NLP) solutions
4	Select and apply relevant Python libraries to synthesize information and construct supervised learning models
5	Develop a comprehensive case study in a multidisciplinary domain, showcasing the integration of AI techniques to solve complex problems.

Course Outcomes **On completion of this course, the student will be able to**	
CO1	Design and develop solutions for informed and uninformed search problems in Al
CO2	Demonstrate reasoning in first order logic using Prolog.
CO3	Utilize advanced package like NLTK for implementing natural language processing
CO4	Demonstrate and enrich knowledge to select and apply python libraries to synthesize information and develop supervised learning models
CO5	Develop a case study in multidisciplinary areas to demonstrate use of AI

List of Programs

1. Write a program to implement Uninformed search techniques:
a. BFS
b. DFS
2. Write a program to implement Informed search techniques
a. Greedy Best first search
b. A algorithm
3. Study of Prolog its facts, and rules.
a. Write simple facts for the statements and querying it.
b. Write a program for Family-tree.
4. Write a program to train and validate the following classifiers for given data (scikit-learn):
a. Decision Tree
b. Multi-layer Feed Forward neural network
5. Text processing using NLTK
a. Remove stop words
b. implement stemming
c. POS (Parts of Speech) tagging
6. In addition to the above programs, students should be encouraged to study implementations of one of the following
Game bot (Tic Tac toe, 7 puzzle)
Expert system (Simple Medical Diagnosis)
Text classification
Chat bot

AI 452PC	**COMPUTER ORGANIZATION AND MICROPROCESSOR LAB**					
Prerequisites			**L**	**T**	**P**	**C**
			0	**0**	**2**	**1**
Evaluation	**CIE**	**25 Marks**	**SEE**		**50 Marks**	

Course Objectives	
1	Attain expertise in logic design, covering gates, flip-flops, and state-of-the-art IC chips like multiplexers, decoders, encoders, counters, and shift-registers
2	Develop skills to design various adders, including BCD adder, adder/subtracter composite unit, carry-look ahead adder, and counters
3	Gain proficiency in designing complex components like ALU and 4-bit processors using Verilog.
4	Acquire hands-on experience in programming using 8085 Microprocessor, covering instruction set, addressing modes, and interfacing with devices
5	Acquire hands-on experience in programming using 8085 Microprocessor, covering instruction set, addressing modes, and interfacing with devices

Course Outcomes On completion of this course, the student will be able to	
CO1	Acquire hands-on experience in programming using 8085 Microprocessor, covering instruction set, addressing modes, and interfacing with devices
CO2	Showcase the ability to design and implement adders, counters, and complex arithmetic units, ensuring accurate and reliable circuit performance.
CO3	Exhibit proficiency in creating intricate components like ALU and 4-bit processors using Verilog, expanding capabilities in digital circuitry design.
CO4	Display adeptness in programming 8085 Microprocessor, utilizing diverse instructions and addressing modes for various tasks.
CO5	Demonstrate skills in programming 8051 Microcontroller for practical applications, including interfacing with converters, motors, and displays.

List of Experiments

PART A: Programs using VERILOG

1. Review of the different logic design ckts., a) Gates b) Flip/Flop(RS, JK, D, T),
2. Familiarity with state of art IC-chips, e.g. a) Multiplexer , b) Decoder, c) Encoder,
d) Counter, e)Shift-Register, f)adder Truth Table verification and clarification from Data-book.
3. Design a BCD adder.
4. Design an Adder/Subtracter composite unit
5. Design a carry-look ahead Adder
6. Design a ripple counter and carry-look ahead counter.
7. Design ALU and 4-bit processor

PART B: 8085 Programming using Microprocessor Trainer Kit

8. Simple programming examples using 8085 instruction set. To understand the use of
various instructions and addressing modes.
9. Interfacing and programming of 8255
10. Interfacing and programming of 8254.
11. Interfacing and programming of 8279.

PART C: 8051 Programming

12. Simple programming examples using 8051 Microcontroller
13. A/D and D/A converter interface
14. Stepper motor interface
15. Display Interface

AI 453PC	**R WORKSHOP**					
Prerequisites			**L**	**T**	**P**	**C**
			0	**0**	**2**	**1**
Evaluation	**CIE**	**25 Marks**	**SEE**		**50 Marks**	

Course Objectives	
1	Develop proficiency in R programming fundamentals, from installation and comments to variables, data types, and basic output.
2	Gain command over R's flow control constructs, including if-else statements, loops, and functions
3	Learn to work with diverse data structures such as strings, vectors, matrices, lists, arrays, data frames, and factors in R
4	Acquire skills to create various data visualizations in R, including bar plots, histograms, pie charts, box plots, and strip charts
5	Learn how to read/write data, perform data analysis tasks, calculate statistics, and work with univariate, bivariate, and multivariate graphs

Course Outcomes **On completion of this course, the student will be able to**	
CO1	Attain the ability to comfortably navigate and utilize R, covering essential concepts and programming constructs
CO2	Demonstrate the capability to write efficient code using R's control structures and functions for effective problem-solving
CO3	Show competence in representing and manipulating data using various R data structures, preparing for more complex analyses.
CO4	Showcase proficiency in creating informative data visualizations, allowing effective communication of insights from various types of data
CO5	Develop the expertise to manipulate, analyze, and visualize data, enabling effective decision-making in diverse scenarios

1. Installation of R – studio
2. Getting started with R
   a. R comments
   b. R variables and constants
   c. R data types
   d. Print output
   e. R numbers
3. R flow control
   a. R Boolean expression
   b. R if … else
   c. R while loop
   d. R for loop
   e. R break and next
   f. R repeat loop
   g. R function
4. R Data structures
   a. R Strings
   b. R Vectors

   c. R Matrix
   d. R List
   e. R Arrays
   f. R Data Frame
   g. R Factor

5. R Data Visualization
   a. R Bar plot
   b. R Histogram
   c. R Pie chart
   d. R Box plot
   e. R Strip chart
   f. R Plot Function
   g. R Save Plot
   h. Color in R
6. R Data Manipulation
   a. R Read and Write CSV
   b. R Read and Write xlsx
   c. R Dataset
   d. R min () and max ()
   e. R mean, median and mode
   f. R Percentile
7. R Program for univariate graphs
   a. Categorical
   b. Quantitative
8. R Program for bivariate graphsarts
   a. Categorical Vs Categorical
   b. Quantitative Vs Quantitative
   c. Categorical Vs Quantitative
9. R Program for multivariate graphs
   a. Grouping
10. R Program for maps
    a. Dot density maps
    b. Choropleth maps
11. R program for time dependent graphs
    a. Time series
    b. Dumbbell charts
    c. Slope graphs
    d. Area charts.

****