

**B. Tech Computer Science and Engineering (Artificial Intelligence and Machine Learning)**  
**Scheme of Studies/Examination**  
**Semester III**

S. No.	Course No.	Subject	L:T:P	Hours/Week	Credits	Examination Schedule				Duration of Exam (Hrs.)
						Major Test	Minor Test	Practical	Total	
1	BS- CS-AIML-201A	Applied Statistical Analysis for AI	3:0:0	3	3	75	25	0	100	3
2	ES- CS-AIML-203A	Data Structure	3:0:0	3	3	75	25	0	100	3
3	PC- CS-AIML-205A	Object Oriented Programming	3:0:0	3	3	75	25	0	100	3
4	PC- CS-AIML-207A	Introduction to AI	3:0:0	3	3	75	25	0	100	3
5	ES- CS-AIML-209A	Programming Language	3:0:0	3	3	75	25	0	100	3
6	HM-902A	Business Intelligence and Entrepreneurship	3:0:0	3	3	75	25	0	100	3
7	PC- CS-AIML-213LA	Data Structure Lab	0:0:2	2	1	0	40	60	100	3
8	PC- CS-AIML-215LA	Object Oriented Programming Lab	0:0:2	2	1	0	40	60	100	3
9	PC- CS-AIML-217LA	<b>Python –Lab-I</b>	0:0:2	2	1	0	40	60	100	3
		<b>Total</b>		<b>24</b>	<b>21</b>	<b>450</b>	<b>270</b>	<b>180</b>	<b>900</b>	
11	SIM-201A*	Seminar on Summer Internship	2:0:0	2	0	0	50	0	50	

<b>BS-CS-AIML-201A</b>	<b>Applied Statistical Analysis for AI</b>					
<b>Lecture</b>	<b>Tutorial</b>	<b>Practical</b>	<b>Major Test</b>	<b>Minor Test</b>	<b>Total</b>	<b>Time</b>
<b>3</b>	<b>0</b>	<b>0</b>	<b>75</b>	<b>25</b>	<b>100</b>	<b>3</b>
<b>Purpose</b>	To gain a broad understanding of the statistical analysis in Artificial Intelligence.					
<b>Course Outcomes(CO)</b>						
<b>CO1</b>	To study the Statistical Analysis concepts with their relationships and process.					
<b>CO2</b>	To familiarize with describing data, transforming and summarizing.					
<b>CO3</b>	To understand testing hypothesis with real time applications.					
<b>CO4</b>	To apply the examining relationships to find the correlation and regression.					
<b>CO5</b>	To demonstrate and analyse using basic statistical techniques with different use cases.					
<b>CO6</b>	To understand the advanced techniques with applications of decision trees, neural networks.					

#### UNIT – I

Introduction, Meaning of Statistics, The Scientific Method, Basic Steps of the Research Process, Experimental Data and Survey Data, Populations and Samples, Census and Sampling Method, Parameter and Statistic, Independent and Dependent Variables, Examining Relationships, Introduction to SPSS Statistics.

#### UNIT – II

Introduction, Types of Data, Data Transformation, Summarizing Data: Graphical Methods, Summarizing Data: Measures of Central Tendency, Summarizing Data: Measures of Dispersion, Levels of Measurement, Random Variables and Probability Distributions, Discrete and Continuous Random Variable, Making Inferences about Populations from samples, Estimator and Estimate, Confidence Interval for Population Mean (Large Sample).

#### UNIT – III

Introduction, Null and Alternative Hypothesis, Type I and Type II Error, The Procedure of Hypothesis Testing; Hypothesis Testing of a Population Mean: Sample, a proportion(One Sample), Population Variance, Population Mean: Two Independent Samples(), Dependent Samples (Paired Samples), Two Population Proportion, Two Population Variances; Analysis of Variance (ANOVA).

#### UNIT – IV

Introduction, Types of Correlation, Karl Pearson Coefficient Correlation, Spearman's Rank Order Correlation, Partial Correlation, Residuals and Plots, Simple Linear Regression, Multiple Regression Model, Repeated Measures, Non-linear Regression, Polynomial Regression Models, Decision Trees, Neural Networks, Cluster Analysis, Factor Analysis.

#### **SUGGESTED BOOKS:**

1. Probability for Statistics and Machine Learning: Anirban DasGupta - 2011
2. An Introduction to Statistics with Python With Applications in the Life Sciences By Thomas Haslwanter, 2016
3. Applied Statistics: A handbook of techniques- Zenon Reynarowych, springer verlag
4. Joel Grus, "Data Science from Scratch: First Principles with Python", O'Reilly Media Media.
5. Jain V.K., "Data Sciences", Khanna Publishing House, Delhi.
6. Applied Statistics-principles and Examples-D.R.Cox and E.J.Snell.
7. Applied statistical methods, Irving W. Burr, Academic press.
8. Probability, Statistics and Random process, Dr.K.Murugesan & P.Gurusamy by Anuradha Agencies, Deepthi publications.
9. Jain V.K., "Big Data and Hadoop", Khanna Publishing House, Delhi.
10. Jiawei Han and Jian Pei, "Data Mining Concepts and Techniques", Third Edition, Morgan Kaufmann Publishers

ES-CS-AIML-203A	Data Structure						
Lecture	Tutorial	Practical	Credit	Major Test	Minor Test	Total	Time
3	0	0	3	75	25	100	3
<b>Purpose</b>	To introduce the principles and paradigms of Data Structures for design and implement the software systems logically and physically.						
<b>Course Outcomes (CO)</b>							
<b>CO 1</b>	To introduce the basic concepts of Data structure , basic data types ,searching and sorting based on array data types.						
<b>CO 2</b>	To introduce the structured data types like Stacks and Queue and its basic operations' implementation.						
<b>CO 3</b>	To introduce dynamic implementation of linked list.						
<b>CO 4</b>	To introduce the concepts of Tree and graph and implementation of traversal algorithms.						

### Unit-I

**Introduction to Data Structures**, Data Types, Built in and User Defined Data Structures, Applications of Data Structure, Algorithm Analysis, Worst, Best and Average Case Analysis, Notations of Space and Time Complexity, Basics of Recursion.

**Arrays**, One Dimensional Arrays, Two Dimensional Arrays and Multi-Dimensional Arrays, Sparse Matrices, Searching from array using Linear and Binary Searching Algorithm, Sorting of array using Selection, Insertion, Bubble, Radix Algorithm

### Unit-II

**Stacks**: Definition, Implementation of Stacks and Its Operations, Evaluation of Infix, prefix and Postfix Expression, Inter-conversion of Infix, Prefix and Post-Fix Expression, Implementation of Merge Sort and Quick Sort Algorithm.

**Queues**: Definition, Sequential Implementation of Linear Queues and Its Operations, Circular Queue and Its Implementation, Priority Queues and Its Implementation, Applications of queues.

### Unit-III

**Linked Lists**: Need of Dynamic Data Structures, Single Link List and Its Dynamic Implementation, Traversing, Insertion, Deletion Operations on Single Link Lists. Comparison between Static and Dynamic, Implementation of Linked List. Circular Link Lists and Doubly Link List, Dynamic Implementation of Primitive Operations on Doubly Linked Lists and Circular Link List. Dynamic Implementation of Stacks and Queues.

### Unit-IV

**Trees**: Definition, Basic Terminology, Binary Tree, External and Internal Nodes, Static and Dynamic Implementation of a Binary Tree, Primitive Operations on Binary Trees, Binary Tree Traversals: Pre-Order, In-Order and Post-Order Traversals. Representation of Infix, Post-Fix and Prefix Expressions using Trees.

**Introduction to Binary Search Trees**: B+ trees, AVL Trees, Threaded Binary trees, Balanced Multi-way search trees, Implementation of Heap Sort Algorithm.

**Graphs**: Basic Terminology, Definition of Undirected and Directed Graphs, Memory Representation of Graphs, Minimum-Spanning Trees, Warshal Algorithm, Graph Traversals Algorithms: Breadth First and Depth First.

### SUGGESTED BOOKS:

1. Theory and Problems of Data Structures by Jr. Seymour Lipschetz, Schaum's outline, TMH.
2. Data Structures and Algorithms by PAI, TMH.
3. Fundamentals of Data structures by Ellis Horowitz and Sartaj Sahni, Pub, 1983, AW.
4. Data Structures and Algorithms by A.V. Aho, J.E. Hopcroft and T.D. Ullman, Original edition, Addison-Wesley, 1999, Low Priced Edition.
5. Data Structures and Program Design in C by Robert Kruse, PHI,

6. Shukla, Data Structures using C++, Wiley India

7. Introduction to Computers Science -An Algorithms Approach, Jean Paul Tremblay, Richard B. Bunt, 2002, T.M.H.

8. Data Structure and the Standard Template library – Willam J. Collins, 2003, T.M.H.

**Note: The Examiner will be given the question paper template and will have to set the question paper according to the template provided along with the syllabus.**

<b>PS-CS-AIML-205A</b>	<b>Object-Oriented Programming</b>						
<b>Lecture</b>	<b>Tutorial</b>	<b>Practical</b>	<b>Credit</b>	<b>Major Test</b>	<b>Minor Test</b>	<b>Total</b>	<b>Time</b>
<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>	<b>75</b>	<b>25</b>	<b>100</b>	<b>3</b>
<b>Purpose</b>	To introduce the principles and paradigms of Object Oriented Programming Language for design and implement the Object Oriented System.						
<b>Course Outcomes (CO)</b>							
<b>CO1</b>	To introduce the basic concepts of object oriented programming language and the its representation.						
<b>CO2</b>	To allocate dynamic memory, access private members of class and the behavior of inheritance and its implementation.						
<b>CO3</b>	To introduce polymorphism, interface design and overloading of operator.						
<b>CO4</b>	To handle backup system using file, general purpose template and handling of raised exception during programming.						

### Unit-I

Introduction to C++, C++ Standard Library, Illustrative Simple C++ Programs. Header Files, Namespaces, Application of object oriented programming.

Object Oriented Concepts, Introduction to Objects and Object Oriented Programming, Encapsulation, Polymorphism, Overloading, Inheritance, Abstract Classes, Accessifier (public/ protected/ private), Class Scope and Accessing Class Members, Controlling Access Function, Constant, Class Member, Structure and Class.

### Unit-II

Friend Function and Friend Classes, This Pointer, Dynamic Memory Allocation and Deallocation (New and Delete), Static Class Members, Constructors, parameter Constructors and Copy Constructors, Deconstructors, Introduction of inheritance, Types of Inheritance, Overriding Base Class Members in a Derived Class, Public, Protected and Private Inheritance, Effect of Constructors and Deconstructors of Base Class in Derived Classes.

### Unit-III

Polymorphism, Pointer to Derived class, Virtual Functions, Pure Virtual Function, Abstract Base Classes, Static and Dynamic Binding, Virtual Deconstructors.

Fundamentals of Operator Overloading, Rules for Operators Overloading, Implementation of Operator Overloading Like <<, >> Unary Operators, Binary Operators.

### Unit-IV

Text Streams and binary stream, Sequential and Random Access File, Stream Input/ Output Classes, Stream Manipulators.

Basics of C++ Exception Handling, Try, Throw, Catch, multiple catch, Re-throwing an Exception, Exception specifications.

Templates: Function Templates, Overloading Template Functions, Class Template, Class Templates and Non- Type Template arguments.

### **SUGGESTED BOOKS:**

1. The complete reference C ++ by Herbert shieldt Tata McGraw Hill.
2. Object Oriented Programming in Turbo C++ by Robert Lafore, 1994, The WAITE Group Press.
3. Shukla, Object Oriented Programming in c++, Wiley India.
4. C++ How to Program by H M Deitel and P J Deitel, 1998, Prentice Hall.
5. Programming with C++ By D Ravichandran, 2003, T.M.H.

**Note: The Examiner will be given the question paper template and will have to set the question paper according to the template provided along with the syllabus.**

<b>PC-CS-AIML-207A</b>	<b>Introduction to AI</b>					
<b>Lecture</b>	<b>Tutorial</b>	<b>Practical</b>	<b>Major Test</b>	<b>Minor Test</b>	<b>Total</b>	<b>Time</b>
<b>3</b>	<b>0</b>	<b>0</b>	<b>75</b>	<b>25</b>	<b>100</b>	<b>3</b>
<b>Purpose</b>	To gain a broad understanding of the discipline of Artificial Intelligence and its scope in various emerging areas.					
<b>Course Outcomes(CO)</b>						
<b>CO1</b>	Demonstrate fundamental understanding of Artificial Intelligence (AI) and its foundation					
<b>CO2</b>	Demonstrate basic concepts of problem solving, searching, inference, perception					
<b>CO3</b>	Demonstrate proficiency in applying AI techniques in various domains					
<b>CO4</b>	Apply basic principles of AI in solutions that require real world knowledge representation and learning					
<b>CO5</b>	Demonstrate the real life examples of Artificial Intelligence					
<b>CO6</b>	Demonstrate an ability to share in discussions of AI, its current scope and limitations, and societal implications					

### UNIT – I

**Scope of AI:** Introduction to Artificial Intelligence, History of Artificial Intelligence, Artificial Intelligence Languages, Multi Agent Systems, natural language processing, vision and speech processing, robotics, expert systems, Case study: Google Duplex, Dialogflow.

### UNIT – II

**Problem Solving, Searching and Planning:** Problem spaces and search, Heuristic and Informed search strategies, Minmax search, Alpha-beta pruning.

Search and optimization (gradient descent), Adversarial search, Planning and scheduling, Case study: Health Care System.

### UNIT – III

**Knowledge Engineering, Representation, Reasoning and finding Optimal Paths:** Knowledge and Knowledge based system, Knowledge and rationality, Logic and inference, Propositional and predicate logic, Ontologies, Bayesian Reasoning, Temporal reasoning, Knowledge Discovery: Data and Web Mining Case study: Medical diagnosis

### UNIT – IV

**Applications of AI in Various domains:** AI in Marketing, AI in Banking, AI in Finance, AI in Agriculture, AI in Health Care, AI in Gaming, AI in Space Exploration, AI in Autonomous vehicles, AI in Chatbots, AI in Creativity.

### **SUGGESTED BOOKS:**

1. E. Rich and K. Knight, "Artificial Intelligence", TMH, 2nd Ed., 1992.
2. N. J. Nilsson, "Principles of AI", Narosa Publ. House, 1990.
3. M. N. Hoda, "Foundation Course in Artificial Intelligence", Vikas Pub., 2004.
4. Artificial Intelligence' RB Mishra, PHI
5. Knowledge and Knowledge based System' Russell.
6. Artificial intelligence, Patrick Henry Winston:, 1992, Addition Wesley 3 Ed.
7. Stuart Russell and Peter Norvig, "Artificial Intelligence: A Modern Approach" , 3rd Edition, Prentice Hall.
8. P. H. Winston, "Artificial Intelligence", Pearson Education, 3rd Edition, 2002. Artificial Intelligence.
9. D. W. Patterson, "Introduction to AI and Expert Systems", PHI, 1992.
10. R. J. Schalkoff, "Artificial Intelligence – An Engineering Approach", McGraw Hill Int. Ed. Singapore, 1992.
11. M. Sasikumar, S. Ramani, "Rule Based Expert Systems", Narosa Publishing House, 1994. 5. Tim Johns, "Artificial Intelligence, Application Programming, Wiley Dreamtech, 2005.
12. Trivedi, M.C., "A Classical Approach to Artificial Intelligence", Khanna Publishing House, Delhi.
13. Saroj Kaushik, "Artificial Intelligence", Cengage Learning India, 2011.
14. David Poole and Alan Mackworth, "Artificial Intelligence: Foundations for Computational Agents", Cambridge University Press 2010.

<b>ES-CS-AIML-209A</b>	<b>Programming Language</b>						
<b>Lecture</b>	<b>Tutorial</b>	<b>Practical</b>	<b>Credit</b>	<b>Major Test</b>	<b>Minor Test</b>	<b>Total</b>	<b>Time</b>
<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>	<b>75</b>	<b>25</b>	<b>100</b>	<b>3</b>
<b>Purpose</b>	To introduce the principles and paradigms of programming languages for design and implement the software intensive systems.						
<b>Course Outcomes (CO)</b>							
<b>CO 1</b>	To introduce the basic concepts of programming language, the general problems and methods related to syntax and semantics.						
<b>CO 2</b>	To introduce the structured data objects, subprograms and programmer defined data types.						
<b>CO 3</b>	To outline the sequence control and data control.						
<b>CO 4</b>	To introduce the concepts of storage management using programming languages.						

#### Unit-I

**Introduction, Syntax and Semantics:** A brief history, Characteristics of a good programming language, Programming language translators- compiler and interpreters, Elementary data types – data objects, variable and constants, data types. Specification and implementation of elementary data types, Declarations, type checking and type conversions, Assignment and initialization, Numeric data types, enumerations, Booleans and characters.

**Syntax and Semantics:** Introduction, general problem of describing syntax, Formal method of describing Syntax, attribute grammar dynamic semantic.

#### Unit-II

**Structured data objects, Subprograms and Programmer Defined Data Types:** Structured data objects and data types, specification and implementation of structured data types, Declaration and type checking of data structure, vector and arrays, records Character strings, variable size data structures, Union, pointer and programmer defined data objects, sets, files.

**Subprograms and Programmer Defined Data Types:** Evolution of data type concept abstraction, encapsulation and information hiding, Subprograms, type definitions, abstract data types, over loaded subprograms, generic subprograms.

#### Unit-III

**Sequence Control and Data Control:** Implicit and explicit sequence control, sequence control within expressions, sequence control within statement, Subprogram sequence control: simple call return, recursive subprograms, Exception and exception handlers, co routines, sequence control. Concurrency – subprogram level concurrency, synchronization through semaphores, monitors and message passing.

**Data Control:** Names and referencing environment, static and dynamic scope, block structure, Local data and local referencing environment, Shared data: dynamic and static scope, Parameter and parameter transmission schemes.

#### Unit-IV

**Storage Management and Programming Languages:** Major run time elements requiring storage, programmer and system controlled storage management and phases, Static storage management, Stack based storage management, Heap storage management, variable and fixed size elements.

**Programming Languages:** Introduction to procedural, non-procedural, structured, logical, functional and object oriented programming language, Comparison of C and C++ programming languages.

#### **SUGGESTED BOOKS:**

- Terrence W. Pratt, Marvin V. Zelkowitz, Programming Languages Design and Implementation, Pearson.
- Allen Tucker and Robert Noonan, Programming Languages–Principles and Paradigms, Tata McGraw-Hill, 2009.
- Ellis Horowitz, Fundamentals of Programming Languages, Galgotia Publications, 2010.
- C. Ghezzi, Programming Languages Concepts, Wiley Publications, 2010.

**Note: The Examiner will be given the question paper template and will have to set the question paper according to the template provided along with the syllabus.**

HM-902 A	Business Intelligence and Entrepreneurship						
Lecture	Tutorial	Practical	Credit	Major Test	Minor Test	Total	Time
3	0	0	3	75	25	100	3
<b>Purpose</b>	To make the students conversant with the basics concepts in management thereby leading to nurturing their managerial skills.						
<b>Course Outcomes (CO)</b>							
<b>CO1</b>	Students will be able understand who the entrepreneurs are and what competences needed to become an Entrepreneur.						
<b>CO2</b>	Students will be able understand insights into the management, opportunity search, identification of a Product; market feasibility studies; project finalization etc. required for small business enterprises.						
<b>CO3</b>	Students can be able to write a report and do oral presentation on the topics such as product identification, business idea, export marketing etc.						
<b>CO4</b>	Students will be able to know the different financial and other assistance available for the small industrial units.						

### UNIT – I

**Entrepreneurship :** Concept and Definitions; Entrepreneurship and Economic Development; Classification and Types of Entrepreneurs; Entrepreneurial Competencies; Factor Affecting Entrepreneurial Growth – Economic, Non-Economic Factors; EDP Programmes; Entrepreneurial Training; Traits/Qualities of an Entrepreneurs; Manager Vs. Entrepreneur, Entrepreneurial challenges.

### UNIT – II

**Opportunity / Identification and Product Selection:** Entrepreneurial Opportunity Search and Identification; Criteria to Select a Product; Conducting Feasibility Studies; Sources of business ideas, Marketing Plan : Conducting of Marketing Research, Industry Analysis, Competitor analysis, market segmentation and positioning, building a marketing plan, marketing mix, launching a new product; export marketing, Methods of Project Appraisal, Project Report Preparation; Specimen of Project Report; Project Planning and Scheduling using Networking Techniques of PERT / CPM.

### UNIT – III

**Small Enterprises and Enterprise Launching Formalities :** Definition of Small Scale; Rationale; Objective; Scope; SSI; Registration; NOC from Pollution Board; Machinery and Equipment Selection , Role of SSI in Economic Development of India; major problem faced by SSI,MSMEs – Definition and Significance in Indian Economy; MSME Schemes, Challenges and Difficulties in availing MSME Schemes.

### UNIT – IV

**Role of Support Institutions and Management of Small Business :** DIC; SIDO; SIDBI; Small Industries Development Corporation (SIDC); SISI; NSIC; NISBUD; State Financial Corporation SIC; Venture Capital : Concept, venture capital financing schemes offered by various financial institutions in India.

**Special Issues for Entrepreneurs:** Legal issues – Forming business entity, requirements for formation of a Private/Public Limited Company, Entrepreneurship and Intellectual Property Rights: IPR and their importance. (Patent, Copy Right, Trademarks) , Case Studies-At least one in whole course.

#### Note:

- Case studies of Entrepreneurs – successful, failed, turnaround ventures should be discussed in the class.
- Exercises / activities should be conducted on ‘generating business ideas’ and identifying problems and opportunities.
- Interactive sessions with Entrepreneurs, authorities of financial institutions, Government officials should be organized

#### SUGGESTED BOOKS:

1. “Entrepreneurship development small business enterprises”, Pearson, Poornima M Charantimath,2013.
2. Roy Rajiv, “Entrepreneurship”, Oxford University Press, 2011.
3. “Innovation and Entrepreneurship”,Harper business- Drucker.F, Peter, 2006.
4. “Entrepreneurship”, Tata Mc-graw Hill Publishing Co.ltd new Delhi- Robert D. Hisrich, Mathew J. Manimala, Michael P Peters and Dean A. Shepherd, 8th Edition, 2012.
5. Enterpreneurship Development- S.Chand and Co.,Delhi- S.S.Khanka 1999



6. Small-Scale Industries and Entrepreneurship. Himalaya Publishing House, Delhi –Vasant Desai 2003.
7. Entrepreneurship Management -Cynthia, Kaulgud, Aruna, Vikas Publishing House, Delhi, 2003.

**Note: The Examiner will be given the question paper template and will have to set the question paper according to the template provided along with the syllabus.**

PC-CS-AIML-213LA	Data Structure Lab						
Lecture	Tutorial	Practical	Credit	Minor Test	Practical	Total	Time
0	0	2	1	40	60	100	2
<b>Purpose</b>	To introduce the principles and paradigms of Data Structures for design and implement the software systems logically and physically.						
<b>Course Outcomes (CO)</b>							
<b>CO1</b>	Implement linear and non linear data structures using linked list.						
<b>CO2</b>	Apply various data structures such as stack, queue and tree to solve the problems.						
<b>CO3</b>	Implement various searching and sorting techniques.						
<b>CO4</b>	Choose appropriate data structure while designing the applications and analyze the complexity of the algorithms.						

### LIST OF PRACTICALS:

1. Write a program for Binary search methods.
2. Write a program for insertion sort, selection sort and bubble sort.
3. Write a program to implement Stack and its operation.
4. Write a program for quick sort.
5. Write a program for merge sort.
6. Write a program to implement Queue and its operation.
7. Write a program to implement Circular Queue and its operation.
8. Write a program to implement singly linked list for the following operations: Create, Display, searching, traversing and deletion.
9. Write a program to implement doubly linked list for the following operations: Create, Display, inserting, counting, searching, traversing and deletion.
10. Write a program to implement circular linked list for the following operations: Create, Display, inserting, counting, searching, traversing and deletion.
11. Write a program to implement insertion, deletion and traversing in B tree

**NOTE:** A student has to perform at least ten experiments. Seven experiments should be performed from the above list. Remaining three experiments may either be performed from the above list or designed & set by the concerned institution as per the scope of the syllabus.

PC-CS-AIML-215LA	Object Oriented Programming Lab						
Lecture	Tutorial	Practical	Credit	Minor Test	Practical	Total	Time
0	0	2	1	40	60	100	2
<b>Purpose</b>	To introduce the principles and paradigms of Object Oriented Programming Language for design and implement the Object Oriented System.						
<b>Course Outcomes (CO)</b>							
CO1	To familiarize with the class and objects						
CO2	To implement the concept of constructors						
CO3	To familiarize the concept of operator overloading						
CO4	To implement the concepts of Inheritance						

### LIST OF PRACTICALS

1. Raising a number  $n$  to a power  $p$  is the same as multiplying  $n$  by itself  $p$  times. Write a function called `power`

( ) that takes a double value for  $n$  and an int value for  $p$ , and returns the result as double value. Use a default argument of 2 for  $p$ , so that if this argument is omitted, the number will be squared. Write a main ( ) function that gets values from the user to test this function.

2. A point on the two dimensional plane can be represented by two numbers: an X coordinate and a Y coordinate. For example, (4,5) represents a point 4 units to the right of the origin along the X axis and 5 units up the Y axis. The sum of two points can be defined as a new point whose X coordinate is the sum of the X coordinates of the points and whose Y coordinate is the sum of their Y coordinates. Write a program that uses a structure called `point` to model a point. Define three points, and have the user input values to two of them. Then set the third point equal to the sum of the other two, and display the value of the new point. Interaction with the program might look like this:

Enter coordinates for P1: 3 4

Enter coordinates for P2: 5 7

Coordinates of P1 + P2 are : 8, 11

3. Create the equivalent of a four function calculator. The program should request the user to enter a number, an operator, and another number. It should then carry out the specified arithmetical operation: adding, subtracting, multiplying, or dividing the two numbers. (It should use a switch statement to select the operation). Finally it should display the result. When it finishes the calculation, the program should ask if the user wants to do another calculation. The response can be 'Y' or 'N'. Some sample interaction with the program might look like this.

Enter first number, operator, and second number: 10/ 3

Answer = 3.333333

Do another (Y/ N)? Y

Enter first number, operator, second number 12 + 100

Answer = 112

Do another (Y/ N) ? N

4. A phone number, such as (212) 767-8900, can be thought of as having three parts: the area code (212), the exchange (767) and the number (8900). Write a program that uses a structure to store these three parts of a phone number separately. Call the structure `phone`. Create two structure variables of type `phone`. Initialize one, and have the user input a number for the other one. Then display both numbers. The interchange might look like this:

- Enter your area code, exchange, and number: 415 555 1212
- My number is (212) 767-8900
- Your number is (415) 555-1212

5. Create two classes DM and DB which store the value of distances. DM stores distances in metres and centimeters and DB in feet and inches. Write a program that can read values for the class objects and add one object of DM with another object of DB. Use a friend function to carry out the addition operation. The object that stores the results maybe a DM object or DB objects, depending on the units in which the results are required. The display should be in the format of feet and inches or metres and cenitmetres depending on the object on display.

6. Create a class rational which represents a numerical value by two double values- NUMERATOR and DENOMINATOR. Include the following public member Functions:

- constructor with no arguments (default).
- constructor with two arguments.
- void reduce( ) that reduces the rational number by eliminating the highest common factor between the numerator and denominator.
- Overload + operator to add two rational number.
- Overload >> operator to enable input through cin.
- Overload << operator to enable output through cout.

Write a main ( ) to test all the functions in the class.

7. Consider the following class definition

```
class father {
protected : int age;
public;
father (int x) {age = x;}
virtual void iam ( )
{ cout << "I AM THE FATHER, my age is : "<< age<< endl;}
```

Derive the two classes son and daughter from the above class and for each, define iam ( ) to write our similar but appropriate messages. You should also define suitable constructors for these classes. Now, write a main ( ) that creates objects of the three classes and then calls iam ( ) for them. Declare pointer to father. Successively, assign addresses of objects of the two derived classes to this pointer and in each case, call iam ( ) through the pointer to demonstrate polymorphism in action.

8. Write a program that creates a binary file by reading the data for the students from the terminal. The data of each student consist of roll no., name ( a string of 30 or lesser no. of characters) and marks.

9. A hospital wants to create a database regarding its indoor patients. The information to store include

- a) Name of the patient
- b) Date of admission
- c) Disease
- d) Date of discharge

Create a structure to store the date (year, month and date as its members). Create a base class to store the above information. The member function should include functions to enter information and display a list of all the patients in the database. Create a derived class to store the age of the patients. List the information about all the to store the age of the patients. List the information about all the pediatric patients (less than twelve years in age).

10. Make a class **Employee** with a name and salary. Make a class **Manager** inherit from **Employee**. Add an instance variable, named department, of type string. Supply a method to **to String** that prints the manager's name, department and salary. Make a class **Executive** inherits from **Manager**. Supply a method **to String** that prints the string "**Executive**" followed by the information stored in the **Manager** superclass object. Supply a test program that tests these classes and methods.

11. Imagine a tollbooth with a class called toll Booth. The two data items are a type unsigned int to hold the total number of cars, and a type double to hold the total amount of money collected. A constructor initializes both these to 0. A member function called payingCar ( ) increments the car total and adds 0.50 to the cash total. Another function, called nopayCar ( ), increments the car total but adds nothing to the cash total. Finally, a member function called displays the two totals. Include a program to test this class. This program should allow the user to push one key to count a paying car, and another to count a nonpaying car. Pushing the ESC kay should cause the program to print out the total cars and total cash and then exit.

**12.** Write a function called `reversit ( )` that reverses a string (an array of char). Use a for loop that swaps the first and last characters, then the second and next to last characters and so on. The string should be passed to `reversit ( )` as an argument. Write a program to exercise `reversit ( )`. The program should get a string from the user, call `reversit ( )`, and print out the result. Use an input method that allows embedded blanks. Test the program with Napoleon's famous phrase, "Able was I ere I saw Elba".

**13.** Create some objects of the string class, and put them in a Deque-some at the head of the Deque and some at the tail. Display the contents of the Deque using the `forEach ( )` function and a user written display function. Then search the Deque for a particular string, using the `first That ( )` function and display any strings that match. Finally remove all the items from the Deque using the `getLeft ( )` function and display each item. Notice the order in which the items are displayed: Using `getLeft ( )`, those inserted on the left (head) of the Deque are removed in "last in first out" order while those put on the right side are removed in "first in first out" order. The opposite would be true if `getRight ( )` were used.

**14.** Assume that a bank maintains two kinds of accounts for customers, one called as savings account and the other as current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed.

Create a class `account` that stores customer name, account number and type of account. From this derive the classes

`cur_acct` and `sav_acct` to make them more specific to their requirements. Include necessary member functions in order to achieve the following tasks:

- a) Accept deposit from a customer and update the balance.
- b) Display the balance.
- c) Compute and deposit interest.
- d) Permit withdrawal and update the balance.
- e) Check for the minimum balance, impose penalty, necessary and update the balance.
- f) Do not use any constructors. Use member functions to initialize the class members.

**15.** Create a base class called `shape`. Use this class to store two double type values that could be used to compute the area of figures. Derive two specific classes called `triangle` and `rectangle` from the base `shape`. Add to the base class, a member function `get_data ( )` to initialize baseclass data members and another member function `display_area ( )` to compute and display the area of figures. Make `display_area ( )` as a virtual function and redefine this function in the derived classes to suit their requirements. Using these three classes, design a program that will accept dimensions of a triangle or a rectangle interactively and display the area.

Remember the two values given as input will be treated as lengths of two sides in the case of rectangles and as base and height in the case of triangles and used as follows:

Area of rectangle =  $x * y$

Area of triangle =  $\frac{1}{2} * x * y$

**NOTE:** A student has to perform at least ten experiments. Seven experiments should be performed from the above list. Remaining three experiments may either be performed from the above list or designed & set by the concerned institution as per the scope of the syllabus.

PC-CS-AIML-217LA	Python Lab-I					
Lecture	Tutorial	Practical	Minor Test	Practical	Total	Time
0	0	2	40	60	100	2
<b>Purpose</b>	To gain a broad understanding of the basics of Python Programming Language.					
<b>Course Outcomes</b>						
<b>CO1</b>	To understand the basic concepts of programming in Python.					
<b>CO2</b>	To implement data types, logical and mathematical operators.					
<b>CO3</b>	To implement list, tuples, dictionaries, arrays, strings					
<b>CO4</b>	To understand and implement the fundamentals of functions, recursion					
<b>CO5</b>	To learn and apply the concepts of looping and conditional statements					
<b>CO6</b>	To learn and implement the fundamentals of searching and sorting					

### LIST OF PRACTICALS:

1. Write a program to demonstrate basic data types in python.
2. Write a program to implement input, output operations and logical , mathematical operations.
3. Write a program for checking whether the given number is an even number or not.
4. Write a program to demonstrate string, list, tuple and dictionaries in python.
5. Write a program using a for loop that loops over a sequence.
6. Write a program to perform logical and mathematical operations.
7. Write a program to use split and join methods in the string and trace a birthday of a person with dictionary data structure.
8. Write a program to implement function and recursion.
9. Write a program to perform linear search and binary search.
10. Write a program to perform uninformed and informed search.
11. Write a program to perform Insertion sort and binary sort.
12. Write a Python program to demonstrate usage of basic regular expression.
13. Write a program to construct Bayesian Network.

NOTE: A student has to perform at least ten experiments. Seven experiments should be performed from the above list. Remaining three experiments may either be performed from the above list or designed & set by the concerned institution as per the scope of the syllabus.

**B. Tech Computer Science and Engineering (Artificial Intelligence and Machine Learning)**  
**Scheme of Studies/Examination**  
**Semester IV**

S. No.	Course No.	Subject	L:T:P	Hours/Week	Credits	Examination Schedule				Duration of Exam (Hrs.)
						Major Test	Minor Test	Practical	Total	
1	BS- CS- AIML- 202M	Mathematics for Machine Learning	3:0:0	3	3	75	25	0	100	3
2	PC- CS- AIML- 204A	Intelligent Systems	3:0:0	3	3	75	25	0	100	3
3	PC- CS- AIML- 206A	Database Management System	3:0:0	3	3	75	25	0	100	3
4	PC- CS- AIML- 208A	Internet and Web Technology	3:0:0	3	3	75	25	0	100	3
5	PC- CS- AIML- 210A	Operating System	3:0:0	3	3	75	25	0	100	3
6	PC- CS- AIML- 212A	Software Engineering	3:0:0	3	3	75	25	0	100	3
7	PC- CS- AIML- 216A	Database Management System Lab	0:0:2	2	1	0	40	60	100	3
8	PC- CS- AIML- 218A	Internet and Web Technology Lab	0:0:2	2	1	0	40	60	100	3
9	PC- CS- AIML-220A	Python –Lab-II	0:0:2	2	1	0	40	60	100	3
		<b>Total</b>		<b>24</b>	<b>21</b>	<b>450</b>	<b>270</b>	<b>180</b>	<b>900</b>	
10	MC-901A	Environmental Sciences	3:0:0	3	0		100	0	100	3

BS-CS-AIML 202M	Mathematics for Machine Learning					
Lecture	Tutorial	Practical	Major Test	Minor Test	Total	Time
3	0	0	75	25	100	3
<b>Purpose</b>	To understand and learn the core concepts of the prerequisite mathematics for applications in data science and machine learning					
<b>Course Outcomes(CO)</b>						
<b>CO1</b>	To understand the basic concepts of data science & machine learning Concepts and their application in modern context					
<b>CO2</b>	To apply the basic statistical concepts for solving various problems					
<b>CO3</b>	To distinguish between various probability distributions and apply the concepts for the solution of related problems					
<b>CO4</b>	To learn the essential tools of matrices and linear algebra including linear transformations, eigen values, diagonalisation, orthogonalization and factorization					
<b>CO5</b>	To learn mathematical modelling, types of matrixes					
<b>CO6</b>	To Implement mathematical concepts using real-world data					

### UNIT – I

**Overview of Data Science & Machine Learning:** Introduction and history of Data Science, Introduction and history of Machine Learning, Overlap between Data Science, Machine Learning and Artificial Intelligence, Applications of Data Science & Machine Learning in the modern context, Types of data, Basic Statistical Concepts: Scale of Measurements (Nominal, Ordinal, Ratio and Interval), Measures of Location, Measures of Variability/Spread, Measures of Shape.

**Case Studies: Bollywood Dataset, coronary heart disease dataset.**

### UNIT – II

**Probability Theory:** Principle of counting, definitions of probability theory, independent events, mutually exclusive events, collectively exhaustive events, conditional probability, Bayes Theorem, Discrete probability distribution (Discrete Uniform Distribution, Poisson Distribution, Bernoulli Distribution and Binomial Distribution), covariance, correlation, Continuous probability distribution, normal distribution, Central Limit Theorem, Binomial Distribution, Continuous Uniform Distribution, Exponential Distribution, P-Value, T-Value, Confidence Interval, t distribution and chi square distribution

### UNIT – III

**Linear Algebra:** Introduction to linear algebra, notations and definitions, Elementary transformations, Elementary matrices, inverse using elementary transformations, Rank of a matrix, Normal form of a matrix, Linear dependence and independence of vectors, Consistency of linear system of equations, Eigen Values and Eigen vectors, Properties of Eigen values, Cayley Hamilton theorem, Linear Transformation, Orthogonal transformation

### UNIT – IV

**Mathematical modelling:** Similar matrices, Diagonalisation of a matrix Operations on matrices - additions, subtraction, multiplication, scalar multiplication, vector multiplication, Orthogonal Matrix, Singularity of Matrix, Matrix factorization, decomposition such as LU, QR and SVD, Conceptualizing a mathematical model/curve from first principles, concept of boundary conditions

### SUGGESTED BOOKS:

1. Probability for Statistics and Machine Learning: Anirban DasGupta - 2011



2. Probability for Machine Learning: Discover How To Harness, Jason Brownlee – 2019.
3. Machine Learning using Python: Manaranjan pradhan, U Dinesh Kumar-2020, Wiley.
4. Machine Learning using Python, PRADHAN, Manaranjan, 1st and Reprint 2019, Wiley.
5. Machine Learning (in Python and R), ) MUELLER,John Paul ,Wiley India
6. 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
- 7.Python for Probability, Statistics, and Machine Learning, José Unpingco – 2019
- 8.Mathematics for Machine Learning, Marc peter Deisenroth, A. Aldo Faisal and Cheng Soon Ong. Published by Cambridge University press.
9. Jeeva Jose, “Machine Learning”, Khanna Publishing House, Delhi.
- 10.Ian Goodfellow, Yoshua Bengio and Aaron Courville, "Deep Learning", MIT Press  
<http://www.deeplearningbook.org>

PC-CS-AIML-204A	Intelligent Systems					
Lecture	Tutorial	Practical	Major Test	Minor Test	Total	Time
3	0	0	75	25	100	3
<b>Purpose</b>	To impart understanding of the main abstractions and reasoning for intelligent systems.					
<b>Course Outcomes(CO)</b>						
<b>CO1</b>	Understand the basic terminologies in artificial intelligence to develop intelligent systems					
<b>CO2</b>	Apply the random search and heuristic search for intelligent systems.					
<b>CO3</b>	Understand the abstractions and reasoning for intelligent systems					
<b>CO4</b>	Apply the rule based methods in intelligent systems					
<b>CO5</b>	Identify the characteristics and architectures of algorithms of multi agent systems					
<b>CO6</b>	Identify different application areas of Intelligent Systems					

### UNIT-I

**Introduction:** Overview of AI Problems, AI problems as NP, NP-Complete, NP-Hard, Strong and weak, neat and scruffy, symbolic and sub-symbolic, knowledge base and data driven AI.

### UNIT-II

**Heuristic Search:** Best First Search, Hill Climbing, Beam Search, Tabu Search.

**Randomized Search:** Simulated Annealing, Genetic Algorithm, Ant Colony Optimization, Particle Swarm Optimization, Basics of probability theory and probability distributions, information theory, Bayesian learning, Gaussian Mixture models and the EM algorithm, Factor analysis, Principal components analysis, Independent Component Analysis.

### UNIT-III

**Intelligent Systems:** Knowledge acquisition, Computational intelligence, Rule-based systems, Forward-chaining (a data-driven strategy), Conflict resolution, Backward chaining (a goal-driven strategy), Sources of uncertainty, Bayesian updating, Certainty theory.

### UNIT-IV

**Possibility theory:** fuzzy sets and fuzzy logic, Object-oriented systems, Data abstraction, Inheritance, Encapsulation, Unified Modeling Language (UML), Dynamic (or late) binding.

**Key Application Areas:** Expert System, Decision Support Systems, **Deep Learning:** Speech and vision, natural Language processing, Information Retrieval, Semantic Web.

### SUGGESTED BOOKS:

1. Artificial Intelligence' RB Mishra, PHI
2. Introduction to Artificial Intelligence, Charnaik, Pearson.
3. Artificial Intelligence by Elaine Rich, Kevin Knight and Shivashankar B Nair, Tata McGraw Hill.
4. Introduction to Artificial Intelligence and Expert Systems by Dan W Patterson, Pearson Education.
5. Artificial Intelligence : Building Intelligent Systems, KULKARNI, Parag , REPRINT, PHI.
6. Crina Grosan, Ajith Abraham, "Intelligent Systems: A Modern Approach ",Springer-Verlag, 2011
7. Bogdan M. Wilamowski, J. David Irwin, "The Industrial Electronics Handbook. Second Edition: Intelligent Systems", CRC Press, 2011
8. Abraham-Kandel, Gideon-Langholz, "Hybrid-Architectures for Intelligent Systems", CRC-Press, 1992
9. Augmented Human, PAPAGIANNIS, Helen , 1st print, SPD.
10. Ian Goodfellow, Yoshua Bengio and Aaron Courville, "Deep Learning", MIT Press,  
<http://www.deeplearningbook.org>

<b>PC-CS-AIML-208A</b>	<b>Internet &amp; Web technology</b>					
<b>Lecture</b>	<b>Tutorial</b>	<b>Practical</b>	<b>Major Test</b>	<b>Minor Test</b>	<b>Total</b>	<b>Time</b>
<b>3</b>	<b>0</b>	<b>0</b>	<b>75</b>	<b>25</b>	<b>100</b>	<b>3</b>
<b>Purpose</b>	To gain a broad understanding of the discipline of Web engineering and its application to the development and management of Web Applications.					
<b>Course Outcomes</b>						
<b>CO1</b>	Learn the basic concepts of information and web architecture.					
<b>CO2</b>	Learn about the skills that will enable to design and build high level web enabled applications.					
<b>CO3</b>	Understand the applicability of Java Script as per current software industry standards.					
<b>CO4</b>	Acquaint the latest programming language for the implementation of object based and procedure based applications using Python.					

### Unit-I

**Information Architecture:** The role of Information Architect, Collaboration and communication, Organizing information, organizational challenges, Organizing web sites and Intranets, Creating cohesive organization systems, designing navigation systems, types of navigation systems, Integrated navigation elements, designing elegant navigation systems, Searching systems, Searching your web site, designing the search interface, Indexing the right stuff, To search or not to search grouping content, conceptual design, High level Architecture Blueprint. Architectural Page Mockups, Design Sketches.

### Unit-II

**Introduction to XHTML and HTML5:** Origins and Evolution of HTML and XHTML, Basic Syntax, Standard XHTML Document Structure, Basic Text Markup, Images, Hypertext Links, Lists, Tables, Forms, HTML5, Syntactic Differences between HTML and XHTML.

**Cascading Style Sheets:** Introduction, Levels of Style Sheets, Style Specification Formats, Selector Forms, Property Value Forms, Font Properties, List Properties, Color, Alignment of Text, Box Model, Background Images, Conflict Resolution.

### Unit -III

**Java Script:** Overview of JavaScript, Object Orientation and JavaScript, General Syntactic Characteristics, Primitives, Operations, and Expressions, Screen Output and Keyboard Input, Control Statements, Object Creation and Modification, Arrays, Functions, Constructors, Pattern Matching Using Regular Expressions, Errors in Scripts

### Unit -IV

**Python:** Introduction to Python, Data Types and Expressions, Control Statements, Strings and Text Files, Lists and Dictionaries, Design with Functions, Design with Classes

**Suggested Books:**

1. By Peter Morville, Louis Rosenfeld, "Information Architecture on the World Wide Web", O'Reilly Media, 2006.
2. Robert W. Sebesta, "Programming The World Wide Web", Eight Edition, Pearson India, 2015.
3. Kenneth A. Lambert, "The Fundamentals of Python: First Programs", 2011, Cengage Learning.
4. Thomas A Powell, "HTML The Complete Reference", Tata McGraw Hill Publications.

PC-CS- AIML- 210A	Operating System							
	Lecture	Tutorial	Practical	Credit	Major Test	Minor Test	Total	Time
	3	0	0	3	75	25	100	3
<b>Purpose</b>	To familiarize the students with the basics of Operating Systems.							
<b>Course Outcomes (CO)</b>								
<b>CO1</b>	To understand the structure and functions of Operating system.							
<b>CO2</b>	To learn about processes, threads and scheduling algorithms.							
<b>CO3</b>	To understand the principle of concurrency and the concept of deadlocks.							
<b>CO4</b>	To understand various memory management scheme and to study I/O management and file systems.							

### Unit-I

**Introduction:** Introduction to OS. Operating system functions, Different types of O.S.: batch process, multi-programmed, time-sharing, real-time, distributed, parallel.

**System Structure:** Computer system operation, I/O structure, storage structure, storage hierarchy, different types of protections, operating system structure (simple, layered, virtual machine), O/S services, system calls.

### Unit-II

**CPU scheduling:** scheduling criteria, preemptive and non-preemptive scheduling, scheduling algorithms, algorithm evaluation, multi-processor scheduling.

**Threads:** overview, benefits of threads, user and kernel threads.

**Process Management:** Concept of processes, process states, process control, co-operating processes, inter-process communication.

**Process Synchronization:** background, critical section problem, critical region, synchronization hardware, Classical problems of synchronization, semaphores.

### Unit-III

**Deadlocks:** Concept of deadlock, deadlock characterization, deadlock prevention, deadlock avoidance, deadlock detection, recovery from deadlock.

**Memory Management:** background, logical vs. physical address space, contiguous memory allocation, paging, segmentation, segmentation with paging. Concept of fragmentation.

**Virtual Memory:** background, demand paging, concept of page replacement, page replacement algorithms, allocation of frames, thrashing.

### Unit-IV

**File Systems:** file concept, file organization and access methods, allocation methods, directory structure, free-space management

**I/O Management:** I/O hardware, polling, interrupts, DMA, kernel I/O subsystem (scheduling, buffering, caching, spooling and device reservation)

**Disk Management:** disk structure, disk scheduling (FCFS, SSTF, SCAN,C-SCAN) , disk reliability, disk Performance parameters

**Protection and Security:**

Goals of protection and security, security attacks, authentication, program threats, system threats, threat monitoring.

**Case studies:** UNIX file system, Windows file system

**Suggested Books:**

- Operating System Concepts”, Abraham Silberschatz, Peter Baer Galvin, and Greg Gagne, Wiley
- Operating systems: a concept based approach”, Dhananjay M. Dhamdhare, McGraw Hill .
- Operating Systems : Internals and Design Principles, William Stallings, Pearson
- Operating Systems Design and Implementation” ,(Prentice Hall Software Series) Andrew S Tanenbaum and Albert S Woodhull.
- Taub and Schilling, Principles of Communication Systems, TMH.
- Mithal G K, Radio Engineering, Khanna Pub.
- Sirnon Haykin, Communication Systems, John Wiley.

**Note: The Examiner will be given the question paper template and will have to set the question paper according to the template provided along with the syllabus.**

<b>PC-CS-AIML-212A</b>	<b>Software Engineering</b>						
<b>Lecture</b>	<b>Tutorial</b>	<b>Practical</b>	<b>Credit</b>	<b>Major Test</b>	<b>Minor Test</b>	<b>Total</b>	<b>Time</b>
<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>	<b>75</b>	<b>25</b>	<b>100</b>	<b>3</b>
<b>Purpose</b>	To gain a broad understanding of the discipline of software engineering and its application to the development and management of software process.						
<b>Course Outcomes(CO)</b>							
<b>CO1</b>	To understand the basic concepts of Software Engineering.						
<b>CO2</b>	To understand the fundamental concept of requirements engineering and Analysis Modelling.						
<b>CO3</b>	To understand the different design techniques and their implementation.						
<b>CO4</b>	To learn about software testing and maintenance measures.						

### Unit-I

**Introduction:** Introduction to Software Engineering, Software Characteristics, Software Crisis, The Evolving role of Software, Software Development Life Cycle (SDLC) Models: Water Fall Model, Prototype Model, Spiral Model, Evolutionary Development Models, Iterative Enhancement Models, RAD, V Model.

### Unit-II

**Software Requirement Specification:** Requirement Engineering Process: Elicitation, Analysis, Documentation, Review and Management of User Needs, Feasibility Study, Data Flow Diagrams, Decision Tables, SRS Document, IEEE Standard for SRS.

**Software Quality:** Software Quality, Concept of Software Quality Assurance (SQA), SEI-CMM Model. Introduction to Software Risk Management and Software Configuration Management

### Unit-III

**Software Design:** Basic Concept of Software Design, Modularization, Design Structure Charts, Pseudo Codes, Flow Charts, Coupling and Cohesion.

**Design Strategies:** Function Oriented Design, Object Oriented Design, Top-Down and Bottom-Up Design.

**Software Measurement and Metrics:** Various Size Oriented Measures: Halstead's Software Science, Function Point (FP) Based Measures, COCOMO, Cyclomatic Complexity Measures: Control Flow Graphs.

### Unit-IV

**Software Construction:** Software construction fundamentals, minimizing complexity, Top-Down and Bottom-Up programming, structured programming, Compliance with Design and Coding Standards.

**Testing:** Testing Objectives, Unit Testing, Integration Testing, system testing, Acceptance Testing, Regression Testing, Structural Testing, Functional Testing, debugging.

**Maintenance:** key issues, Types of software Maintenance, Cost of Maintenance, Software Re-Engineering.

### Suggested Books:

1. R. S. Pressman, Software Engineering: A Practitioners Approach, McGraw Hill.
2. K. K. Aggarwal and Yogesh Singh, Software Engineering, New Age International Publishers.\
3. Pankaj Jalote, Software Engineering, Wiley India.
4. Rajib Mall, Fundamentals of Software Engineering, PHI Publication.
5. Ian Sommerville, Software Engineering, Addison Wesley.

PC-CS-AIML-216A	Database Management Systems Lab							
	L	T	P	Credit	Practical	Minor Test	Total	Time
	0	0	2	1	60	40	100	2
<b>Purpose</b>	<b>To implement practically the various concepts of DBMS</b>							
	<b>Course Outcomes</b>							
<b>CO1</b>	To understand & Implement basic DDL commands.							
<b>CO2</b>	To learn & Implement DML and DCL commands.							
<b>CO3</b>	To understand the SQL queries using SQL operators.							
<b>CO4</b>	To understand the concept of relational algebra and implement using examples.							

### LIST OF PRACTICALS

1. Create a database and write the programs to carry out the following operation:
  - Add , Delete and modify a record in the database
  - Generate queries
  - Data operations
  - List all the records of database in ascending order.
2. To perform various integrity constraints on relational database.
3. Create a database and perform the following operations:-
  1. Arithmetic and Relational operations
  2. Group by & having clauses
  3. Like predicate for pattern matching in database
4. Create a view to display details of employees working on more than one project.
5. Create a view to display details of employees not working on any project.
6. Using two tables create a view which shall perform natural join, equi join, outer joins.
7. Write a procedure to give incentive to employees working on all projects. If no such employee found give app. Message.
8. Write a procedure for computing amount telephone bill on the basis of following conditions.
  1. telephone rent Rs. 205 including first 105 free units.
  2. if extra units >0 but <500 then rate is 80 paise per unit.
  3. if extra units >500 then rate is Rs. 1.20 per unit.
 For this purpose create a table with name, Phone No., No. of units consumed, bill amount of a customer.
9. Write a procedure for computing income tax of employee on the basis of following conditions:-
  1. if gross pay ≤ 40,000 then I.T rate is 0%.
  2. if gross pay >40,000 but <60000 then I.T rate is 10%.
  3. if gross pay >60,000 but <1,00,000 then I.T rate is 20%.
  4. if gross pay >1,00,000 then I.T rate is 30%.
 For this purpose create a table with name, ssn, gross salary and income tax of the employee.
10. Write trigger for before and after insertion, deletion and updation process.

**NOTE:** A student has to perform at least ten experiments. Seven experiments should be performed from the above list. Remaining three experiments may either be performed from the above list or designed & set by the concerned institution as per the scope of the syllabus.

PC-CS-AIML- 218A	<b>Internet and Web Technology Lab</b>						
<b>L</b>	<b>T</b>	<b>P</b>	<b>Credit</b>	<b>Practical</b>	<b>Minor Test</b>	<b>Total</b>	<b>Time</b>
<b>0</b>	<b>0</b>	<b>2</b>	<b>1</b>	<b>60</b>	<b>40</b>	<b>100</b>	<b>2</b>
<b>Purpose</b>	To introduce the concepts of HTML5, JavaScript and Python.						
<b>Course Outcomes (CO)</b>							
<b>CO1</b>	Design webpages using HTML, JavaScript and CSS.						
<b>CO2</b>	Design and test simple function/program to implement Searching and sorting techniques using Python.						
<b>CO3</b>	Develop program in Java Script for pattern matching using regular expressions and errors in scripts.						
<b>CO4</b>	Design client-server based web applications.						

### LIST OF PRACTICALS

1. Create your own page with your favorite hobbies using HTML, JavaScript and CSS.
2. Create a frameset in HTML that is divided into three sections. The frameset should have three zones.
  - a. The Topmost section of the frameset should take up about just 15% of the browser window. Name this frame title.
  - b. The middle section should be 75% of the browser window. Name this frame title.
  - c. The lower section should be 10% of the browser window. Name this frame menu.
3. Create pages for each section. For the lowermost section, create page that loads the content into the middle section. The topmost section should contain a page describing the web page itself.
4. Create a web page, which displays the map of your country Link, each city /state on the image map, such that the respective HTML page of the city/state is displayed when the user selects an area.
5. Add the tickertape applet to your page by customizing it for the following settings:
  - a. Increase the count by one.
  - b. Accordingly update the message count.
  - c. Change the text color to (237,192,171)
  - d. Experiment with changing the scrolling speed.
  - e. Customize the message text as per your page requirement.
6. Incorporate a quest book into the Diary Food Webpage and use Java Script to build validations into the form.
7. Use Cascading Style sheets (CSS) to modify the following:
  - a. Change background.
  - b. Change font type, face and color.
  - c. Align Text.
  - d. Remove underlines from hyperlinks.
8. Write the program for using JavaScript by using for – loops (through a block of code a number of times), for/in - loops (through the properties of an object), while - loops (through a block of code while a specified condition is true), do/while - loops (through a block of code while a specified condition is true).
9. Write a program in Java Script for the following:
  - a. Copying, passing, and comparing by value
  - b. Copying, passing, and comparing by reference
  - c. References themselves are passed by value
10. Write program in Java Script for pattern matching using regular expressions and errors in scripts.
11. Write a Python function/program that accepts the lengths of three sides of a triangle as inputs. The program output should indicate whether or not the triangle is an equilateral triangle.
12. Write the Python functions for linear search, binary search, selection sort, Bubble Sort, Insertion Sort and converting Fibonacci to a linear algorithm.
13. Write program in Python using Lists and dictionaries, Control statements and Strings and text files.

PC-CS-AIML-220A	Python Lab-II					
Lecture	Tutorial	Practical	Minor Test	Practical	Total	Time
0	0	2	40	60	100	2
<b>Purpose</b>	To gain a broad understanding of the discipline of machine Learning and its implementation using different libraries.					
<b>Course Outcomes</b>						
<b>CO1</b>	To understand the basic concepts of Python libraries					
<b>CO2</b>	To learn and apply concepts of data manipulation in machine Learning .					
<b>CO3</b>	To learn and apply descriptive analysis concepts.					
<b>CO4</b>	To understand the fundamentals of knowledge representation.					
<b>CO5</b>	To learn and apply concepts of distribution and hypothesis.					
<b>CO6</b>	To understand and implement various data visualization concepts.					

#### LIST OF PRACTICALS:

1. Write a program to implement of Basic Python Libraries-numpy, scipy.
2. Write a program to implement of Basic Python Libraries-matplotlib, pandas, Scikitlearn.
3. Write a program to create samples from population.
4. Write a program to evaluate Mean, Median, Mode of dataset.
5. Write a program to implement Central Limit Theorem in dataset.
6. Write a program to implement Measure of Spread in dataset.
7. Write a program to implement program to differentiate between descriptive and inferential statistics.
8. Write a program to implement pmf, pdf and cdf.
9. Write a program to implement different visualization techniques on sample dataset.
10. Write a program to implement different hypothesis test on sample dataset.

NOTE: A student has to perform at least ten experiments. Seven experiments should be performed from the above list. Remaining three experiments may either be performed from the above list or designed & set by the concerned institution as per the scope of the syllabus.



MC-901A	Environmental Sciences						
Lecture	Tutorial	Practical	Credit	Major Test	Minor Test	Total	Time
3	0	0	0	75	25	100	3
<b>Purpose</b>	To learn the multidisciplinary nature, scope and importance of Environmental sciences.						
<b>Course Outcomes (CO)</b>							
<b>CO1</b>	The students will be able to learn the importance of natural resources.						
<b>CO2</b>	To learn the theoretical and practical aspects of eco system.						
<b>CO3</b>	Will be able to learn the basic concepts of conservation of biodiversity.						
<b>CO4</b>	The students will be able to understand the basic concept of sustainable development.						

### Unit-I

The multidisciplinary nature of environmental studies, Definition, Scope and Importance, Need for public awareness, Natural Resources: Renewable and Non-Renewable Resources: Natural resources and associated problems.

- Forest Resources: Use and over-exploitation, deforestation, case studies. Timber extraction, mining, dams and their effects on forests and tribal people.
- Water Resources: Use and over-utilization of surface and ground water, floods, drought, conflicts over water, dams-benefits and problems.
- Mineral Resources: Use and exploitation, environmental effects of extracting and using mineral resources, case studies.
- Food Resources: World Food Problems, changes caused by agriculture and overgrazing, effects of modern agriculture, fertilizer-pesticide problems, water logging, salinity, case studies.
- Energy Resources: Growing energy needs, renewable and non-renewable energy sources, use of alternate energy sources. Case studies.
- Land Resources: Land as a resource, land, degradation, man induced landslides, soil erosion and desertification.

Role of an individual in conservation of natural resources, Equitable use of resources for sustainable lifestyle.

### Unit-II

**Ecosystem-Concept of an ecosystem:** Structure and function of an ecosystem, Producers, consumers and decomposers, Energy flow in the ecosystem, Ecological Succession, Food Chains, food webs and ecological pyramids. Introduction, types, characteristic features, structure and function of the following ecosystem: (a) Forest Ecosystem, (b) Grassland Ecosystem, (c) Desert Ecosystem and (d) Aquatic Ecosystems (ponds, streams, lakes, rivers, oceans, estuaries)

Field Work: Visit to a local area to document Environment assets-river/forest/grassland/hill/mountain, Visit to a local polluted site-Urban /Rural Industrial/Agricultural, Study of common plants, insects and birds, Study of simple ecosystems-pond, river, hill, slopes etc. (Field work equal to 5 lecture hours).

### Unit-III

**Biodiversity and its conservation:** Introduction, Definition: genetic, species and ecosystem diversity. Biogeographical classification of India. Value of biodiversity: consumptive use, productive use, social, ethical, aesthetic and option values. Biodiversity of global, National and local levels. India as a mega-diversity nation Hot spot of Biodiversity, Threats to biodiversity: Habitat loss, poaching of wild life, man-wildlife conflicts, Endangered and endemic species of India, Conservation of Biodiversity- In situ and Ex-Situ conservation of biodiversity.

**Environmental Pollution Definition:** Cause, effects and control measures of (a) Air Pollution (b) Water Pollution (c) Soil Pollution (d) Marine Pollution (e) Noise Pollution (f) Thermal Pollution (g) Nuclear Hazards

Solid waste management- cause, effects and control measures of urban and industrial wastes, Role of an individual in prevention of pollution, Pollution case studies, Disaster management: floods, earthquake, cyclone and landslides

### Unit-IV

**Social Issues and the Environment:** From unsustainable to sustainable development, Urban problems related to energy, Water conservation, rain water harvesting, watershed management. Resettlement and rehabilitation of people: Its problems and concerns, Case Studies: Environmental ethics-issues and possible solutions. Climate change, global warming, acid rain, ozone layer depletion, nuclear accidents and holocaust. Case studies: Wasteland Reclamation, Consumerism and waste products, Environment Protection Act, Air (Prevention and Control of Pollution) Act, Water (Prevention and Control of Pollution) Act, Wildlife Protection Act, Forest Conservation Act, Issues involved in enforcement of environmental legislation, Public Awareness, Human population and the Environment, Population growth, variation among nations, Population explosion-Family Welfare Programme, Environment and human health. Human rights, Value Education, HIV/AIDS, Women and Child Welfare, Role of Information Technology in Environment and Human Health, Case Studies, Drugs and their effects; Useful and harmful drugs, Use and abuse of drugs, Stimulant and depression drugs, Concept of drug de-addiction, Legal position on drugs and laws related to drugs.

**SUGGESTED BOOKS:**

- Environmental Studies- Deswal and Deswal. Dhanpat Rai and Co.
- Environmental Science and Engineering Anandan, P. and Kumaravelan, R. 2009. Scitech Publications (India) Pvt. Ltd., India
- Environmental Studies. Daniels Ranjit R. J. and Krishnaswamy. 2013. Wiley India.
- Environmental Science- Botkin and Keller. 2012. Wiley , India

**Note: The Examiner will be given the question paper template to set the question paper.**