

**PROPOSED COURSE STRUCTURE–B.TECH (AI&ML)-R23
(Applicable from the 2023-24 admitted batches)**

B.Tech.– II Year I Semester

S.No.	Category	Title	L	T	P	Credits
1	BS&H	Discrete Mathematics & Graph Theory	3	0	0	3
2	BS&H	Universal human values – understanding harmony and Ethical human conduct	2	1	0	3
3	Engineering Science	Artificial Intelligence	3	0	0	3
4	Professional Core	Advanced Data Structures & Algorithm Analysis	3	0	0	3
5	Professional Core	Object Oriented Programming Through Java	3	0	0	3
6	Professional Core	Advanced Data Structures and Algorithm Analysis Lab	0	0	3	1.5
7	Professional Core	Object Oriented Programming Through Java Lab	0	0	3	1.5
8	Skill Enhancement Course	Python Programming	0	1	2	2
9	Audit Course	Environmental Science	2	0	0	-
Total			16	2	8	20

B.Tech.– II Year II Semester

S.No.	Category	Title	L	T	P	Credits
1	Management Course- I	Optimization Techniques	2	0	0	2
2	Engineering Science/ Basic Science	Probability & Statistics	3	0	0	3
3	Professional Core	Machine Learning	3	0	0	3
4	Professional Core	Database Management Systems	3	0	0	3
5	Professional Core	Digital Logic & Computer Organization	3	0	0	3
6	Professional Core	AI &ML Lab	0	0	3	1.5
7	Professional Core	Database Management Systems Lab	0	0	3	1.5
8	Skill Enhancement course	Full Stack Development - 1	0	1	2	2



NRI INSTITUTE OF TECHNOLOGY(AUTONOMOUS)

GUNTUR-522 438, ANDHRA PRADESH

DEPARTMENT OF AI&ML

9	BS&H	Design Thinking &Innovation	1	0	2	2
Total			15	1	12	21
Mandatory Community Service Project Internship of 08 weeks duration during summer Vacation						

II Year I Semester

L	T	P	C
3	1	0	3

ARTIFICIAL INTELLIGENCE

Pre-requisite:

1. Knowledge in Computer Programming
2. A course on “Mathematical Foundations of Computer Science”.
3. Background in linear algebra, data structures and algorithms, and probability

Course Objectives:

1. The student should be made to study the concepts of Artificial Intelligence.
2. The student should be made to learn the methods of solving problems using Artificial Intelligence.
3. The student should be made to introduce the concepts of Expert Systems.
4. To understand the applications of AI, namely game playing, theorem proving, and machine learning.
5. To learn different knowledge representation techniques

UNIT - I

Introduction: AI problems, foundation of AI and history of AI intelligent agents: Agents and Environments, the concept of rationality, the nature of environments, structure of agents, problem solving agents, problem formulation.

UNIT - II

Searching- Searching for solutions, uniformed search strategies – Breadth first search, depth first Search. Search with partial information (Heuristic search) Hill climbing, A* ,AO* Algorithms, Problem reduction, Game Playing-Adversial search, Games, mini-max algorithm, optimal decisions in multiplayer games, Problem in Game playing, Alpha-Beta pruning, Evaluation functions.

UNIT - III

Representation of Knowledge: Knowledge representation issues, predicate logic- logic programming, semantic nets- frames and inheritance, constraint propagation, representing knowledge using rules, rules based deduction systems. Reasoning under uncertainty, review of probability, Bayes’ probabilistic interferences and dempstershafer theory.

UNIT - IV

Logic concepts: First order logic. Inference in first order logic, propositional vs. first order inference, unification & lifts forward chaining, Backward chaining, Resolution, Learning from observation Inductive learning, Decision trees, Explanation based learning, Statistical Learning methods, Reinforcement Learning.

UNIT - V

Expert Systems: Architecture of expert systems, Roles of expert systems – Knowledge Acquisition Meta knowledge Heuristics. Typical expert systems – MYCIN, DART, XCON: Expert systems shells.

Textbooks:

1. S. Russel and P. Norvig, “Artificial Intelligence – A Modern Approach”, Second Edition, Pearson Education.
2. Kevin Night and Elaine Rich, Nair B., “Artificial Intelligence (SIE)”, Mc Graw Hill

Reference Books:

1. David Poole, Alan Mackworth, Randy Goebel, “Computational Intelligence: a logical approach”, Oxford University Press.
2. G. Luger, “Artificial Intelligence: Structures and Strategies for complex problemsolving”, Fourth Edition, Pearson Education.
3. J. Nilsson, “Artificial Intelligence: A new Synthesis”, Elsevier Publishers.
4. Artificial Intelligence, SarojKaushik, CENGAGE Learning.

Online Learning Resources:

1. <https://ai.google/>
2. https://swayam.gov.in/nd1_noc19_me71/preview

II-B.TECH I-SEMESTER

L	T	P	C
3	1	0	3

ADVANCED DATA STRUCTURES & ALGORITHM ANALYSIS

Course Objectives:

- To understand and apply asymptotic notations (Big-O, Big-Theta, Big-Omega) for analyzing the time and space complexities of algorithms.
- To implement and manipulate advanced data structures such as AVL trees, B-trees, heaps, and graphs, and apply them effectively to solve computational problems.
- To design and apply diverse algorithmic strategies including divide and conquer, greedy method, dynamic programming, and backtracking, for solving complex computational problems across various domains.
- To proficiently implement algorithms and data structures in programming languages like C++ or Java, ensuring correctness, efficiency, and adherence to specified computational constraints.

Course Outcomes:

Students will be able to:

- Analyze algorithms using asymptotic notations to evaluate their efficiency and scalability in diverse computational contexts. (L4-Analyze).
- Implement and manipulate advanced data structures effectively to solve computational problems. (L3-Apply).
- Design and apply diverse algorithmic strategies to solve complex problems across various domains. (L6-Create).
- Implement algorithms and data structures proficiently in programming contexts. (L3-Apply).
- Develop comprehensive computational problem-solving skills, integrating theoretical knowledge with practical application across diverse domains. (L6-Create).

UNIT-1

Performance Analysis: Algorithm, Performance Analysis-Space complexity, Time complexity, Asymptotic Notations- Big oh notation, Omega notation, Theta notation and Little oh notation.

Trees Part-I:

AVL Trees: Basics, Representation, Rotations, Insertion and Deletion operations.

Red Black Trees: Basics, Representation, Insertions, Deletions and Search Operations.

UNIT – II:

Trees Part-II:

B Trees: Basics, Representation, Insertions, Deletions and Search Operations.

B+ Trees: Representation, Operations, Applications.

Splay Trees: Representation, Operations, Applications.

Binary Heaps: Representation, Operations, Applications.

Graphs: Terminology, Representations, Basic Search and Traversals, Connected Components and Biconnected Components, applications.

UNIT – III:

Divide and Conquer, Greedy Method and Dynamic Programming:

Divide and Conquer: The General Method, Binary Search, Finding the maximum and minimum, Quick Sort, Merge Sort, Strassen's matrix multiplication.

Greedy method: General method, applications-Job sequencing with deadlines, knapsack problem, Minimum cost spanning trees, Single source shortest path problem.

Dynamic Programming: General Method, All pairs shortest paths, Single Source Shortest Paths– General Weights (Bellman Ford Algorithm), Optimal Binary Search Trees, 0/1 Knapsack, String Editing, Travelling Salesperson problem.

UNIT – IV:

Back tracking, Branch and Bound.

Backtracking: General Method, 8-Queens Problem, Sum of Subsets problem, Graph Coloring, 0/1 Knapsack Problem.

Branch and Bound: The General Method, 0/1 Knapsack Problem, Travelling Salesperson problem.

UNIT – V:

NP Hard and NP Complete Problems.

NP Hard and NP Complete Problems: Basic Concepts, Cook's theorem

NP Hard Graph Problems: Clique Decision Problem (CDP), Chromatic Number Decision Problem (CNDP), Traveling Salesperson Decision Problem (TSP).

NP Hard Scheduling Problems: Scheduling Identical Processors, Job Shop Scheduling.

Text books:

1. Fundamentals of Data Structures in C++, Horowitz, Ellis; Sahni, Sartaj; Mehta, Dinesh, 2nd Edition Universities Press.
2. Computer Algorithms in C++, Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, 2nd Edition University Press.

Reference Books:

1. Data Structures and program design in C, Robert Kruse, Pearson Education Asia
2. An introduction to Data Structures with applications, Trembley & Sorenson, McGraw Hill
3. The Art of Computer Programming, Vol.1: Fundamental Algorithms, Donald E Knuth, Addison-Wesley, 1997.
4. Data Structures using C & C++: Langsam, Augenstein & Tanenbaum, Pearson, 1995
5. Algorithms + Data Structures & Programs: N. Wirth, PHI
6. Fundamentals of Data Structures in C++: Horowitz Sahni & Mehta, Galgottia Publications.
7. Data structures in Java:, Thomas Standish, Pearson Education Asia.

Online Learning Resources:

1. https://www.tutorialspoint.com/advanced_data_structures/index.asp
2. <http://peterindia.net/Algorithms.html>
3. Abdul Bari, [Introduction to Algorithms \(youtube.com\)](https://www.youtube.com/watch?v=...)

II-B.TECH I-SEMESTER

L	T	P	C
3	1	0	3

OBJECT-ORIENTED PROGRAMMING THROUGH JAVA

Course Objectives

- Identify Java language components and understand their integration within applications.
- Learn the fundamentals of object-oriented programming in Java, including creation of classes, method invocation, and utilization of class libraries.
- Explore the concepts of inheritance and dynamic binding in Java, and gain proficiency in using exception handling for robust application development.
- Understand the design principles for developing threaded applications in Java.
- Gain proficiency in utilizing Java APIs for effective program development and integration.

Course Outcomes

- Apply Object-Oriented Programming Principles in Java (L3-Apply).
- Analyze Java Programming Constructs and Control Statements (L4-Analyze).
- Evaluate Advanced Java Features: Packages, Generics, and Exception Handling(L5-Evaluate).
- Apply Multithreading and Synchronization in Java (L3-Apply).
- Implement Java Database Connectivity (JDBC) and GUI Development(L3-Apply)

UNIT-I

Introduction to OOP and Java: Overview of OOP, Object oriented programming paradigms, Features of Object-Oriented Programming, Java Buzzwords, Java Virtual Machine (JVM) , Java Run-time environment(JRE).

Program Structure in Java: Introduction, Writing Simple Java Programs, Elements or Tokens, in Java Programs, Java Statements, Command Line Arguments, User Input to Programs, Escape Sequences Comments, Programming Style.

Data Types, Variables and Arrays: Primitive types, Floating-Point types, Characters, Booleans, Literals, Variables, Type Conversion and Casting, Automatic type promotion in expressions. **Arrays-** 1D arrays, Multidimensional arrays, Basics of Strings. **Operators-** Arithmetic operators, bitwise operators, Relational operators, Boolean Logical Operators, Assignment operator. The ? operator, operator precedence.

Control Statements: Selection statements, Iteration statements, Jump statements,

UNIT-II

Classes and Objects: Introduction, Class Declaration and Modifiers, Class Members, Declaration of Class Objects, Assigning One Object to Another, Access Control for Class

Members, Accessing Private Members of Class, Constructor Methods for Class,

Overloaded Constructor Methods, Nested Classes, Final Class and Methods, Passing Arguments by Value and by Reference, Keyword this.

Methods: Introduction, Defining Methods, Overloaded Methods, Overloaded Constructor Methods, Class Objects as Parameters in Methods, Access Control, Recursive Methods, Nesting of Methods, Overriding Methods, Attributes Final and Static.

Inheritance: Introduction, Process of Inheritance, Types of Inheritances, Universal Super Class-Object Class, Inhibiting Inheritance of Class Using Final, Access Control and Inheritance, Multilevel Inheritance, Application of Keyword Super, Constructor Method and Inheritance, Method Overriding, Dynamic Method Dispatch, Abstract Classes, Interfaces and Inheritance.

UNIT-III

Packages and Java Library: Introduction, Defining Package, Importing Packages and Classes into Programs, Path and Class Path, Access Control, Packages in Java SE, Java.lang Package and its Classes, Class Object, Enumeration, class Math, Wrapper Classes, Autoboxing and Auto-unboxing, Java util Classes and Interfaces,

Exception Handling: Introduction, Hierarchy of Standard Exception Classes, Keywords throws and throw, try, catch, and finally Blocks, Multiple Catch Clauses, Class Throwable, Unchecked Exceptions, Checked Exceptions.

UNIT-IV

I/O, Generics, String Handling: I/O Basics – Reading and Writing Console I/O – Reading and Writing Files, Object Serialization.

String Handling: Introduction, Interface Char Sequence, Class String, Methods for Extracting Characters from Strings, Comparison, Modifying, Searching; Class String Buffer.

Multithreaded Programming: The Java Thread Model-Thread Priorities, Synchronization, Messaging, The Thread Class and the Runnable interface. The Main Thread, **creating a Thread**-Implementing Runnable, Extending Thread, **Creating Multiple Threads, Thread Priorities, Synchronization-Basics.**

UNIT-V

Java Database Connectivity: Introduction, JDBC Architecture, Installing MySQL and MySQL Connector/J, JDBC Environment Setup, Establishing JDBC Database Connections, Result Set Interface.

Java FX GUI: Java FX Scene Builder, Java FX App Window Structure, displaying text and image, event handling, laying out nodes in scene graph, mouse events.

Text Books:

- 1) JAVA one step ahead, Anitha Seth, B.L.Juneja, Oxford.
- 2) Joy with JAVA, Fundamentals of Object-Oriented Programming, Debasis Samanta, Monalisa Sarma, Cambridge, 2023.
- 3) JAVA 9 for Programmers, Paul Deitel, Harvey Deitel, 4th Edition, Pearson.
- 4) Thinking in Java by Bruce Eckel
- 5) The complete Reference Java, 11th edition, Herbert Schildt, TMH

Reference Books:

- 1) Introduction to Java programming, 7th Edition, Y Daniel Liang, Pearson.
- 2) Learning Java, Pat Niemeyer Jonathan Knudsen Publisher: O'Reilly.

Online Resources:

- 1) <https://nptel.ac.in/courses/106/105/106105191/>
- 2)

https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_012880464547618816347_shared/overview

II Year I Semester

L	T	P	C
0	0	3	1.5

ADVANCED DATA STRUCTURES & ALGORITHM ANALYSIS LAB

Course Objectives

- Develop practical proficiency in constructing and managing advanced data structures including AVL trees, B-Trees, and Heap Trees.
- Apply a variety of algorithm design techniques to solve computational problems such as graph traversals, sorting techniques, and optimization problems.
- Implement and analyze operations on data structures like AVL trees and B-Trees, including insertion, deletion, and traversal operations.
- Explore and implement advanced algorithms such as dynamic programming for solving optimization problems like the 0/1 Knapsack Problem and the Travelling Salesperson Problem.

Course Outcomes:

- CO1: Apply advanced data structures, including AVL trees, B-Trees, and Heap Trees, to optimize data management and retrieval(L-3).
- CO2: Analyze algorithms for graph traversal, sorting, and optimization problems to ensure efficient problem-solving in various computational scenarios. (L-4).
- CO3: Apply dynamic programming, greedy algorithms, and backtracking techniques to solve complex problems such as the 0/1 Knapsack Problem, Travelling Salesperson Problem, and N-Queens Problem(L-3).
- CO4: Analyze the performance of different algorithmic approaches for finding minimum cost spanning trees and shortest paths, comparing their efficiency in diverse graph representations(L-4).
- CO5: Apply skills in constructing and manipulating data structures and algorithms, addressing real-world computational challenges (L-3).

Implement the following using C/C++/Java:

1. Write a program to perform the following:
 - i) Creating a AVL Tree of integers.
 - ii) Traversing the above binary tree in preorder, inorder and postorder.
2. Write a program to perform the following:
 - i) Creating a B-Tree of integers.
 - ii) Traversing the above binary tree in preorder, inorder and postorder.
3. Construct Min and Max Heap using arrays, delete any element and display the content of the Heap.

4. Implement BFT and DFT for given graph, when graph is represented by:
 - i) Adjacency Matrix.
 - ii) Adjacency Lists.
5. Write a program for finding the biconnected components in a given graph.
6. Implement Quick sort and Merge sort and observe the execution time for various input sizes (Average, Worst and Best cases).
7. Write a program to find the closest pair of points using a divide and conquer strategy. Use the random number generator to generate a large number of points in a unit square as input to the algorithm.
8. Use dynamic programming to find the optimal binary search tree for a given set of numbers together with their probabilities. Remember that the numbers may be generated in any order, so, a presorting step is also required.
9. Write a program for finding Single Source Shortest Paths using the Greedy method.
10. Implement Job Sequencing with deadlines using Greedy strategy.
11. Implement N-Queens Problem Using Backtracking.
12. Implement Travelling Sales Person problem using Branch and Bound approach.

TEXT BOOKS:

1. Mark Allen Weiss, Data Structures and Algorithm Analysis in C++, 3rd Edition, Pearson Education, 2007.

REFERENCES:

1. Sartaj Sahni, Data Structures Algorithms and Applications in C++, 2nd Edition, Universities Press, 2007.
2. Ellis Horowitz, Sartaj Sahni, Rajasekharan, Fundamentals of Algorithms, 2nd Edition, Universities Press, 2009.
3. Aho V Alfred, Hapcroft E John, Ullman D Jeffry, Data Structures and Algorithms, 1st Edition, Pearson Education, 2002.
4. Adam Drozdek, Thomson, Data Structures and Algorithms in JAVA, 3rd Edition, Cengage Learning, 2008.
5. Horowitz, Sahni, Mehta, Fundamentals of Data Structures in C++, 2nd Edition, Universities Press, 2007.

WEB REFERENCES:

1. www.nptel.iitm.ac.in/video.php?subjectid=106102064
2. <http://cse01-iiith.vlabs.ac.in/>
3. <http://peterindia.net/Algorithms.html>

II Year I Semester

L	T	P	C
0	0	3	1.5

OBJECT ORIENTED PROGRAMMING THROUGH JAVA LAB

Course Objectives:

The aim of this course is to

- Practice object oriented programming in the Java programming language
- Implement Classes, Objects, Methods, Inheritance, Exception, Runtime Polymorphism, User defined Exception handling mechanism
- Illustrate inheritance, Exception handling mechanism, JDBC connectivity
- Construct Threads, Event Handling, implement packages, Java FX GUI

Experiments covering the Topics:

- Object Oriented Programming fundamentals- data types, control structures
- Classes, methods, objects, Inheritance, polymorphism,
- Exception handling, Threads, Packages, Interfaces
- Files, I/O streams, Java FX GUI

Sample Experiments:

Exercise – 1:

- a) Write a JAVA program to display default value of all primitive data type of JAVA
- b) Write a java program that display the roots of a quadratic equation $ax^2+bx=0$. Calculate the discriminate D and basing on value of D, describe the nature of root.

Exercise - 2

- a) Write a JAVA program to search for an element in a given list of elements using binary search mechanism.
- b) Write a JAVA program to sort for an element in a given list of elements using bubble sort
- c) Write a JAVA program using String Buffer to delete, remove character.

Exercise - 3

- a) Write a JAVA program to implement class mechanism. Create a class, methods and invoke them inside main method.
- b) Write a JAVA program implement method overloading.
- c) Write a JAVA program to implement constructor.
- d) Write a JAVA program to implement constructor overloading.

Exercise - 4

- a) Write a JAVA program to implement Single Inheritance
- b) Write a JAVA program to implement multi level Inheritance
- c) Write a JAVA program for abstract class to find areas of different shapes

Exercise - 5

- a) Write a JAVA program give example for “super” keyword.

- b) Write a JAVA program to implement Interface. What kind of Inheritance can be achieved?
- c) Write a JAVA program that implements Runtime polymorphism

Exercise - 6

- a) Write a JAVA program that describes exception handling mechanism
- b) Write a JAVA program Illustrating Multiple catch clauses
- c) Write a JAVA program for creation of Java Built-in Exceptions
- d) Write a JAVA program for creation of User Defined Exception

Exercise - 7

- a) Write a JAVA program that creates threads by extending Thread class. First thread display "Good Morning "every 1 sec, the second thread displays "Hello "every 2 seconds and the third display "Welcome" every 3 seconds,(Repeat the same by implementing Runnable)
- b) Write a program illustrating is Alive and join ()
- c) Write a Program illustrating Daemon Threads.
- d) Write a JAVA program Producer Consumer Problem

Exercise – 8

- a) Write a JAVA program that import and use the user defined packages
- b) Without writing any code, build a GUI that display text in label and image in an ImageView (use JavaFX)
- c) Build a Tip Calculator app using several JavaFX components and learn how to respond to user interactions with the GUI

Exercise – 9

- a) Write a java program that connects to a database using JDBC
- b) Write a java program to connect to a database using JDBC and insert values into it.
- c) Write a java program to connect to a database using JDBC and delete values from it



II Year I Semester

L	T	P	C
0	1	2	2

PYTHON PROGRAMMING (SKILL ENHANCEMENT COURSE)

Course Objectives:

The main objectives of the course are to

- Introduce core programming concepts of Python programming language.
- Demonstrate about Python data structures like Lists, Tuples, Sets and dictionaries
- Implement Functions, Modules and Regular Expressions in Python Programming and to create practical and contemporary applications using these

UNIT-I:

History of Python Programming Language, Thrust Areas of Python, Installing Anaconda Python Distribution, Installing and Using Jupyter Notebook.

Parts of Python Programming Language: Identifiers, Keywords, Statements and Expressions, Variables, Operators, Precedence and Associativity, Data Types, Indentation, Comments, Reading Input, Print Output, Type Conversions, the type () Function and Is Operator, Dynamic and Strongly Typed Language.

Control Flow Statements: if statement, if-else statement, if...elif...else, Nested if statement, while Loop, for Loop, continue and break Statements, Catching Exceptions Using try and except Statement.

Sample Experiments:

1. Write a program to find the largest element among three Numbers.
2. Write a Program to display all prime numbers within an interval
3. Write a program to swap two numbers without using a temporary variable.
4. Demonstrate the following Operators in Python with suitable examples.
 - i) Arithmetic Operators
 - ii) Relational Operators
 - iii) Assignment Operators
 - iv) Logical Operators
 - v) Bit wise Operators
 - vi) Ternary Operator
 - vii) Membership Operators
 - viii) Identity Operators
5. Write a program to add and multiply complex numbers
6. Write a program to print multiplication table of a given number.

UNIT-II:

Functions: Built-In Functions, Commonly Used Modules, Function Definition and Calling the function, return Statement and void Function, Scope and Lifetime of Variables, Default Parameters, Keyword Arguments, *args and **kwargs, Command

Line Arguments.

Strings: Creating and Storing Strings, Basic String Operations, Accessing Characters in String by Index Number, String Slicing and Joining, String Methods, Formatting Strings.

Lists: Creating Lists, Basic List Operations, Indexing and Slicing in Lists, Built-In Functions Used on Lists, List Methods, del Statement.

Sample Experiments:

1. Write a program to define a function with multiple return values.
2. Write a program to define a function using default arguments.
3. Write a program to find the length of the string without using any library functions.
4. Write a program to check if the substring is present in a given string or not.
5. Write a program to perform the given operations on a list:
 - i. addition
 - ii. insertion
 - iii. slicing
6. Write a program to perform any 5 built-in functions by taking any list.

UNIT-III:

Dictionaries: Creating Dictionary, Accessing and Modifying key:value Pairs in Dictionaries, Built-In Functions Used on Dictionaries, Dictionary Methods, del Statement.

Tuples and Sets: Creating Tuples, Basic Tuple Operations, tuple() Function, Indexing and Slicing in Tuples, Built-In Functions Used on Tuples, Relation between Tuples and Lists, Relation between Tuples and Dictionaries, Using zip() Function, Sets, Set Methods, Frozenset.

Sample Experiments:

1. Write a program to create tuples (name, age, address, college) for at least two members and concatenate the tuples and print the concatenated tuples.
2. Write a program to count the number of vowels in a string (No control flow allowed).
3. Write a program to check if a given key exists in a dictionary or not.
4. Write a program to add a new key-value pair to an existing dictionary.
5. Write a program to sum all the items in a given dictionary.

UNIT-IV:

Files: Types of Files, Creating and Reading Text Data, File Methods to Read and Write Data, Reading and Writing Binary Files, Pickle Module, Reading and Writing CSV Files, Python os and os.path Modules.



Object-Oriented Programming: Classes and Objects, Creating Classes in Python, Creating Objects in Python, Constructor Method, Classes with Multiple Objects, Class Attributes Vs Data Attributes, Encapsulation, Inheritance, Polymorphism.

Sample Experiments:

1. Write a program to sort words in a file and put them in another file. The output file should have only lower-case words, so any upper-case words from source must be lowered.
2. Python program to print each line of a file in reverse order.
3. Python program to compute the number of characters, words and lines in a file.
4. Write a program to create, display, append, insert and reverse the order of the items in the array.
5. Write a program to add, transpose and multiply two matrices.
6. Write a Python program to create a class that represents a shape. Include methods to

calculate its area and perimeter. Implement subclasses for different shapes like circle, triangle, and square.

UNIT-V:

Introduction to Data Science: Functional Programming, JSON and XML in Python, NumPy with Python, Pandas.

Sample Experiments:

1. Python program to check whether a JSON string contains complex object or not.
2. Python Program to demonstrate NumPy arrays creation using array () function.
3. Python program to demonstrate use of ndim, shape, size, dtype.
4. Python program to demonstrate basic slicing, integer and Boolean indexing.
5. Python program to find min, max, sum, cumulative sum of array
6. Create a dictionary with at least five keys and each key represent value as a list where this list contains at least ten values and convert this dictionary as a pandas data frame and explore the data through the data frame as follows:
 - a) Apply head () function to the pandas data frame
 - b) Perform various data selection operations on Data Frame
7. Select any two columns from the above data frame, and observe the change in one attribute with respect to other attribute with scatter and plot operations in matplotlib

Reference Books:

1. Gowrishankar S, Veena A., Introduction to Python Programming, CRC Press.
2. Python Programming, S Sridhar, J Indumathi, V M Hariharan, 2ndEdition, Pearson, 2024



NRI INSTITUTE OF TECHNOLOGY(AUTONOMOUS)
GUNTUR-522 438, ANDHRA PRADESH
DEPARTMENT OF AI&ML

3. Introduction to Programming Using Python, Y. Daniel Liang, Pearson.

Online Learning Resources/Virtual Labs:

1. <https://www.coursera.org/learn/python-for-applied-data-science-ai>
2. <https://www.coursera.org/learn/python?specialization=python#syllabus>



NRI INSTITUTE OF TECHNOLOGY(AUTONOMOUS)
GUNTUR-522 438, ANDHRA PRADESH
DEPARTMENT OF AI&ML

L	T	P	C
3	0	0	3

II Year II Semester

MACHINE LEARNING

Course Objectives:

The objectives of the course is to

- Define machine learning and its different types (supervised and unsupervised) and understand their applications.
- Apply supervised learning algorithms including decision trees and k-nearest neighbours (k-NN).
- Implement unsupervised learning techniques, such as K-means clustering.

UNIT-I: Introduction to Machine Learning: Evolution of Machine Learning, Paradigms for ML, Learning by Rote, Learning by Induction, Reinforcement Learning, Types of Data, Matching, Stages in Machine Learning, Data Acquisition, Feature Engineering, Data Representation, Model Selection, Model Learning, Model Evaluation, Model Prediction, Search and Learning, Data Sets.

UNIT-II: Nearest Neighbor-Based Models: Introduction to Proximity Measures, Distance Measures, Non-Metric Similarity Functions, Proximity Between Binary Patterns, Different Classification Algorithms Based on the Distance Measures ,K-Nearest Neighbor Classifier, Radius Distance Nearest Neighbor Algorithm, KNN Regression, Performance of Classifiers, Performance of Regression Algorithms.

UNIT-III: Models Based on Decision Trees: Decision Trees for Classification, Impurity Measures, Properties, Regression Based on Decision Trees, Bias–Variance Trade-off, Random Forests for Classification and Regression.

The Bayes Classifier: Introduction to the Bayes Classifier, Bayes’ Rule and Inference, The Bayes Classifier and its Optimality, Multi-Class Classification | Class Conditional Independence and Naive Bayes Classifier (NBC)

UNIT-IV: Linear Discriminants for Machine Learning: Introduction to Linear Discriminants, Linear Discriminants for Classification, Perceptron Classifier, Perceptron Learning Algorithm, Support Vector Machines, Linearly Non-Separable Case, Non-linear SVM, Kernel Trick, Logistic Regression, Linear Regression, Multi-Layer Perceptrons (MLPs), Backpropagation for Training an MLP.

UNIT-V: Clustering : Introduction to Clustering, Partitioning of Data, Matrix Factorization | Clustering of Patterns, Divisive Clustering, Agglomerative Clustering,



Partitional Clustering, K-Means Clustering, Soft Partitioning, Soft Clustering, Fuzzy C-Means Clustering, Rough Clustering, Rough K-Means Clustering Algorithm, Expectation Maximization-Based Clustering, Spectral Clustering.

Text Books:

1. "Machine Learning Theory and Practice", M N Murthy, V S Ananthanarayana, Universities Press (India), 2024

Reference Books:

1. "Machine Learning", Tom M. Mitchell, McGraw-Hill Publication, 2017
2. "Machine Learning in Action", Peter Harrington, DreamTech
3. "Introduction to Data Mining", Pang-Ning Tan, Michel Stenbach, Vipin Kumar, 7th Edition, 2019.



L	T	P	C
3	0	0	3

II Year II Semester

DATABASE MANAGEMENT SYSTEMS

Course Objectives:

The main objectives of the course is to

- Introduce database management systems and to give a good formal foundation on the relational model of data and usage of Relational Algebra
- Introduce the concepts of basic SQL as a universal Database language
- Demonstrate the principles behind systematic database design approaches by covering conceptual design, logical design through normalization
- Provide an overview of physical design of a database system, by discussing Database indexing techniques and storage techniques

UNIT I:

Introduction: Database system, Characteristics (Database Vs File System), Database Users, Advantages of Database systems, Database applications. Brief introduction of different Data Models; Concepts of Schema, Instance and data independence; Three tier schema architecture for data independence; Database system structure, environment, Centralized and Client Server architecture for the database.

Entity Relationship Model: Introduction, Representation of entities, attributes, entity set, relationship, relationship set, constraints, sub classes, super class, inheritance, specialization, generalization using ER Diagrams.

UNIT II:

Relational Model: Introduction to relational model, concepts of domain, attribute, tuple, relation, importance of null values, constraints (Domain, Key constraints, integrity constraints) and their importance, Relational Algebra, Relational Calculus. BASIC SQL:Simple Database schema, data types, table definitions (create, alter), different DML operations (insert, delete, update).

UNIT III:

SQL:Basic SQL querying (select and project) using where clause, arithmetic & logical operations, SQL functions(Date and Time, Numeric, String conversion).Creating tables with relationship, implementation of key and integrity constraints, nested queries, sub queries, grouping, aggregation, ordering, implementation of different types of joins, view(updatable and non-updatable), relational set operations.



NRI INSTITUTE OF TECHNOLOGY(AUTONOMOUS)
GUNTUR-522 438, ANDHRA PRADESH
DEPARTMENT OF AI&ML

UNIT IV:

Schema Refinement (Normalization): Purpose of Normalization or schema refinement, concept of functional dependency, normal forms based on functional dependency Lossless join and dependency preserving decomposition, (1NF, 2NF and 3 NF), concept of surrogate key, Boyce-Codd normal form (BCNF), MVD, Fourth normal form (4NF), Fifth Normal Form (5NF).

UNIT V:

Transaction Concept: Transaction State, ACID properties, Concurrent Executions, Serializability, Recoverability, Implementation of Isolation, Testing for Serializability, lock based, time stamp based, optimistic, concurrency protocols, Deadlocks, Failure Classification, Storage, Recovery and Atomicity, Recovery algorithm.

Introduction to Indexing Techniques: B+ Trees, operations on B+ Trees, Hash Based Indexing:

Text Books:

- 1) Database Management Systems, 3rd edition, Raghurama Krishnan, Johannes Gehrke, TMH (For Chapters 2, 3, 4)
- 2) Database System Concepts, 5th edition, Silberschatz, Korth, Sudarsan, TMH (For Chapter 1 and Chapter 5)

Reference Books:

- 1) Introduction to Database Systems, 8th edition, C J Date, Pearson.
- 2) Database Management System, 6th edition, Ramez Elmasri, Shamkant B. Navathe, Pearson
- 3) Database Principles Fundamentals of Design Implementation and Management, Corlos Coronel, Steven Morris, Peter Robb, Cengage Learning.

Web-Resources:

- 1) <https://nptel.ac.in/courses/106/105/106105175/>
- 2) https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_012758066672820_22456_shared/overview



L	T	P	C
3	0	0	3

II Year II Semester

DIGITAL LOGIC & COMPUTER ORGANIZATION

Course Objectives:

The main objectives of the course is to

- provide students with a comprehensive understanding of digital logic design principles and computer organization fundamentals
- Describe memory hierarchy concepts
- Explain input/output (I/O) systems and their interaction with the CPU, memory, and peripheral devices

UNIT – I:

Data Representation: Binary Numbers, Fixed Point Representation. Floating Point Representation. Number base conversions, Octal and Hexadecimal Numbers, components, Signed binary numbers, Binary codes

Digital Logic Circuits-I: Basic Logic Functions, Logic gates, universal logic gates, Minimization of Logic expressions. K-Map Simplification, Combinational Circuits, Decoders, Multiplexers

UNIT – II:

Digital Logic Circuits-II: Sequential Circuits, Flip-Flops, Binary counters, Registers, Shift Registers, Ripple counters

Basic Structure of Computers: Computer Types, Functional units, Basic operational concepts, Bus structures, Software, Performance, multiprocessors and multi computers, Computer Generations, Von- Neumann Architecture

UNIT – III:

Computer Arithmetic : Addition and Subtraction of Signed Numbers, Design of Fast Adders, Multiplication of Positive Numbers, Signed-operand Multiplication, Fast Multiplication, Integer Division, Floating-Point Numbers and Operations

Processor Organization: Fundamental Concepts, Execution of a Complete Instruction, Multiple-Bus Organization, Hardwired Control and Multi programmed Control

UNIT – IV:

The Memory Organization: Basic Concepts, Semiconductor RAM Memories, Read-Only Memories, Speed, Size and Cost, Cache Memories, Performance Considerations, Virtual Memories, Memory Management Requirements, Secondary Storage



NRI INSTITUTE OF TECHNOLOGY(AUTONOMOUS)

GUNTUR-522 438, ANDHRA PRADESH

DEPARTMENT OF AI&ML

UNIT – V:

Input/Output Organization: Accessing I/O Devices, Interrupts, Processor Examples, Direct Memory Access, Buses, Interface Circuits, Standard I/O Interfaces.

Textbooks:

1. Computer Organization, Carl Hamacher, Zvonko Vranesic, Safwat Zaky, 6th edition, McGraw Hill, 2023.
2. Digital Design, 6th Edition, M. Morris Mano, Pearson Education, 2018.
3. Computer Organization and Architecture, William Stallings, 11th Edition, Pearson, 2022.

Reference Books:

1. Computer Systems Architecture, M. Morris Mano, 3rd Edition, Pearson, 2017.
2. Computer Organization and Design, David A. Paterson, John L. Hennessy, Elsevier, 2004.
3. Fundamentals of Logic Design, Roth, 5th Edition, Thomson, 2003.

Online Learning Resources:

<https://nptel.ac.in/courses/106/103/106103068/>



II Year II Semester

L	T	P	C
0	0	3	1.5

AI & ML LAB

Course Objectives:

1. The student should be made to study the concepts of Artificial Intelligence.
2. The student should be made to learn the methods of solving problems using Artificial Intelligence.
3. The student should be made to introduce the concepts of Expert Systems and machine learning.
4. To learn about computing central tendency measures and Data preprocessing techniques
5. To learn about classification and regression algorithms
6. To apply different clustering algorithms for a problem.

Software Required for ML: Python/R/Weka

List of Experiments

1. Pandas Library
 - a) Write a python program to implement Pandas Series with labels.
 - b) Create a Pandas Series from a dictionary.
 - c) Creating a Pandas Data Frame.
 - d) Write a program which makes use of the following Pandas methods
 - i) describe ()
 - ii) head ()
 - iii) tail ()
 - iv) info ()
2. Pandas Library: Visualization
 - a) Write a program which use pandas inbuilt visualization to plot following graphs:
 - i. Bar plots
 - ii. Histograms
 - iii. Line plots
 - iv. Scatter plots
3. Write a Program to Implement Breadth First Search using Python.
4. Write a program to implement Best First Searching Algorithm
5. Write a Program to Implement Depth First Search using Python.
6. Write a program to implement the Heuristic Search
7. Write a python program to implement A* and AO* algorithm. (Ex: find the shortest path)
8. Apply the following Pre-processing techniques for a given dataset.
 - a. Attribute selection
 - b. Handling Missing Values
 - c. Discretization
 - d. Elimination of Outliers
9. Apply KNN algorithm for classification and regression
10. Demonstrate decision tree algorithm for a classification problem and perform parameter tuning for better results
11. Apply Random Forest algorithm for classification and regression
12. Demonstrate Naïve Bayes Classification algorithm.



NRI INSTITUTE OF TECHNOLOGY(AUTONOMOUS)
GUNTUR-522 438, ANDHRA PRADESH
DEPARTMENT OF AI&ML

13. Apply Support Vector algorithm for classification
14. Implement the K-means algorithm and apply it to the data you selected. Evaluate performance by measuring the sum of the Euclidean distance of each example from its class center. Test the performance of the algorithm as a function of the parameters K.



II Year II Semester

L	T	P	C
0	0	3	1.5

DATABASE MANAGEMENT SYSTEMS LAB

Course Objectives:

This Course will enable students to

- Populate and query a database using SQL DDL/DML Commands
- Declare and enforce integrity constraints on a database
- Writing Queries using advanced concepts of SQL
- Programming PL/SQL including procedures, functions, cursors and triggers

Experiments covering the topics:

- DDL, DML, DCL commands
- Queries, nested queries, built-in functions,
- PL/SQL programming- control structures
- Procedures, Functions, Cursors, Triggers,
- Database connectivity- ODBC/JDBC

Sample Experiments:

1. Creation, altering and dropping of tables and inserting rows into a table (use constraints while creating tables) examples using SELECT command.
2. Queries (along with sub Queries) using ANY, ALL, IN, EXISTS, NOTEXISTS, UNION, INTERSET, Constraints. Example:- Select the roll number and name of the student who secured fourth rank in the class.
3. Queries using Aggregate functions (COUNT, SUM, AVG, MAX and MIN), GROUP BY, HAVING and Creation and dropping of Views.
4. Queries using Conversion functions (to_char, to_number and to_date), string functions (Concatenation, lpad, rpad, ltrim, rtrim, lower, upper, initcap, length, substr and instr), date functions (Sysdate, next_day, add_months, last_day, months_between, least, greatest, trunc, round, to_char, to_date)
5.
 - i. Create a simple PL/SQL program which includes declaration section, executable section and exception –Handling section (Ex. Student marks can be selected from the table and printed for those who secured first class and an exception can be raised if no records were found)
 - ii. Insert data into student table and use COMMIT, ROLLBACK and SAVEPOINT in PL/SQL block.



6. Develop a program that includes the features NESTED IF, CASE and CASE expression. The program can be extended using the NULLIF and COALESCE functions.
7. Program development using WHILE LOOPS, numeric FOR LOOPS, nested loops using ERROR Handling, BUILT –IN Exceptions, USE defined Exceptions, RAISE-APPLICATION ERROR.
8. Programs development using creation of procedures, passing parameters IN and OUT of PROCEDURES.
9. Program development using creation of stored functions, invoke functions in SQL Statements and write complex functions.
10. Develop programs using features parameters in a CURSOR, FOR UPDATE CURSOR, WHERE CURRENT of clause and CURSOR variables.
11. Develop Programs using BEFORE and AFTER Triggers, Row and Statement Triggers and INSTEAD OF Triggers
12. Create a table and perform the search operation on table using indexing and non- indexing techniques.
13. Write a Java program that connects to a database using JDBC
14. Write a Java program to connect to a database using JDBC and insert values into it
15. Write a Java program to connect to a database using JDBC and delete values from it

Text Books/Suggested Reading:

1. Oracle: The Complete Reference by Oracle Press
2. Nilesh Shah, "Database Systems Using Oracle", PHI, 2007
3. Rick F Vander Lans, "Introduction to SQL", Fourth Edition, Pearson Education, 2007



II Year II Semester

L	T	P	C
0	1	2	2

FULL STACK DEVELOPMENT – 1 (SKILL ENHANCEMENT COURSE)

Course Objectives:

The main objectives of the course are to

- Make use of HTML elements and their attributes for designing static web pages
- Build a web page by applying appropriate CSS styles to HTML elements
- Experiment with JavaScript to develop dynamic web pages and validate forms

Experiments covering the Topics:

- Lists, Links and Images
- HTML Tables, Forms and Frames
- HTML 5 and Cascading Style Sheets, Types of CSS
- Selector forms
- CSS with Color, Background, Font, Text and CSS Box Model
- Applying JavaScript - internal and external, I/O, Type Conversion
- JavaScript Conditional Statements and Loops, Pre-defined and User-defined Objects
- JavaScript Functions and Events
- Node.js

Sample Experiments:




1. Lists, Links and Images

- a. Write a HTML program, to explain the working of lists.

Note: It should have an ordered list, unordered list, nested lists and ordered list in an unordered list and definition lists.

- b. Write a HTML program, to explain the working of hyperlinks using <a> tag and href, target Attributes.
- c. Create a HTML document that has your image and your friend's image with a specific height and width. Also when clicked on the images it should navigate to their respective profiles.
- d. Write a HTML program, in such a way that, rather than placing large images on a page, the preferred technique is to use thumbnails by setting the height and width parameters to something like to 100*100 pixels. Each thumbnail image is also a link to a full sized version of the image. Create an image gallery using this technique

2. HTML Tables, Forms and Frames

- a. Write a HTML program, to explain the working of tables. (use tags: <table>, <tr>, <th>, <td> and attributes: border, rowspan, colspan)
- b. Write a HTML program, to explain the working of tables by preparing a timetable. (Note: Use <caption> tag to set the caption to the table & also use cell spacing, cell padding, border, rowspan, colspan etc.).
- c. Write a HTML program, to explain the working of forms by designing Registration form. (Note: Include text field, password field, number field, date of birth field, checkboxes, radio buttons, list boxes using <select>&<option> tags, <text area> and two buttons ie: submit and reset. Use tables to provide a better view).
- d. Write a HTML program, to explain the working of frames, such that page is to be divided into 3 parts on either direction. (Note: first frame , second frame , third frame  hyperlink. And also make sure of using “no frame” attribute such that frames to be fixed).

3. HTML 5 and Cascading Style Sheets, Types of CSS

- a. Write a HTML program, that makes use of <article>, <aside>, <figure>, <figcaption>, <footer>, <header>, <main>, <nav>, <section>, <div>, tags.
- b. Write a HTML program, to embed audio and video into HTML web page.
- c. Write a program to apply different types (or levels of styles or style specification formats)
 - inline, internal, external styles to HTML elements. (identify selector, property and value).

4. Selector forms

- a. Write a program to apply different types of selector forms
 - i. Simple selector (element, id, class, group, universal)
 - ii. Combinator selector (descendant, child, adjacent sibling, general sibling)
 - iii. Pseudo-class selector
 - iv. Pseudo-element selector
 - v. Attribute selector

5. CSS with Color, Background, Font, Text and CSS Box Model

- a. Write a program to demonstrate the various ways you can reference a color in CSS.
- b. Write a CSS rule that places a background image halfway down the page, tilting it horizontally. The image should remain in place when the user scrolls up or down.
- c. Write a program using the following terms related to CSS font and text:

- i. font-size ii. font-weight iii. font-style
- iv. text-decoration v. text-transformation vi. text-alignment
- d. Write a program, to explain the importance of CSS Box model using
 - i. Content ii. Border iii. Margin iv. padding

6. Applying JavaScript - internal and external, I/O, Type Conversion

- a. Write a program to embed internal and external JavaScript in a web page.
- b. Write a program to explain the different ways for displaying output.
- c. Write a program to explain the different ways for taking input.
- d. Create a webpage which uses prompt dialogue box to ask a voter for his name and age. Display the information in table format along with either the voter can vote or not

7. Java Script Pre-defined and User-defined Objects

- a. Write a program using document object properties and methods.
- b. Write a program using window object properties and methods.
- c. Write a program using array object properties and methods.
- d. Write a program using math object properties and methods.
- e. Write a program using string object properties and methods.
- f. Write a program using regex object properties and methods.
- g. Write a program using date object properties and methods.
- h. Write a program to explain user-defined object by using properties, methods, accessors, constructors and display.

8. Java Script Conditional Statements and Loops

- a. Write a program which asks the user to enter three integers, obtains the numbers from the user and outputs HTML text that displays the larger number followed by the words "LARGER NUMBER" in an information message dialog. If the numbers are equal, output HTML text as "EQUAL NUMBERS".
- b. Write a program to display week days using switch case.
- c. Write a program to print 1 to 10 numbers using for, while and do-while loops.
- d. Write a program to print data in object using for-in, for-each and for-of loops
- e. Develop a program to determine whether a given number is an 'ARMSTRONG NUMBER' or not. [Eg: 153 is an Armstrong number, since sum of the cube of the digits is equal to the number i.e., $13 + 53 + 33 = 153$]
- f. Write a program to display the denomination of the amount deposited in the bank in terms of 100's, 50's, 20's, 10's, 5's, 2's & 1's. (Eg: If deposited amount is Rs.163, the output should be 1-100's, 1-50's, 1- 10's, 1-2's & 1-1's)

9. Java Script Functions and Events

- a. Design a appropriate function should be called to display
 - i. Factorial of that number
 - ii. Fibonacci series up to that number
 - iii. Prime numbers up to that number
 - iv. Is it palindrome or not

Design a HTML having a text box and four buttons named Factorial, Fibonacci, Prime, and Palindrome. When a button is pressed an appropriate function should be called to display

- i. Factorial of that number
 - ii. Fibonacci series up to that number
 - iii. Prime numbers up to that number
 - iv. Is it palindrome or not
- b. Write a program to validate the following fields in a registration page
- c. Write a program to validate the following fields in a registration page
 - i. Name (start with alphabet and followed by alphanumeric and the length should not be less than 6 characters)
 - ii. Mobile (only numbers and length 10 digits)
 - iii. E-mail (should contain format like xxxxxxx@xxxxxx.xxx)

Text Books:

1. Programming the World Wide Web, 7th Edition, Robert W Sebesta, Pearson, 2013.
2. Web Programming with HTML5, CSS and JavaScript, John Dean, Jones & Bartlett Learning, 2019 (Chapters 1-11).
3. Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node, Vasan Subramanian, 2nd edition, APress, O'Reilly.

Web Links:

1. <https://www.w3schools.com/html>
2. <https://www.w3schools.com/css>
3. <https://www.w3schools.com/js/>
4. <https://www.w3schools.com/nodejs>