# B.TECH. HR-21 COURSE STRUCTURE

## CSE-ARTIFICIAL INTELIGENCE & MACHINE LEARNING

(Applicable for the batch admitted from 2021-22 onwards)

### III – Semester (II – Year)

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Course code</th>
<th>Subject</th>
<th>Hours Per Week</th>
<th>Credits</th>
<th>Scheme of Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>L</td>
<td>T</td>
<td>P</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Int.</td>
<td>Ext.</td>
<td>Tot.</td>
</tr>
<tr>
<td>1</td>
<td>21BS3MT03</td>
<td>Mathematical &amp; Statistical Foundations</td>
<td>3</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>21PC3CS03</td>
<td>Object Oriented Programming using Java</td>
<td>3</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>21HS3EG03</td>
<td>English for Employability</td>
<td>1</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>21PC3CS01</td>
<td>Data Structures</td>
<td>3</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>21PC3CS02</td>
<td>Database Management Systems</td>
<td>3</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>21PC3CS04</td>
<td>Database Management Systems-Lab</td>
<td>-</td>
<td>-</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>21HS3EG04</td>
<td>English for Employability-Lab</td>
<td>-</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>21PC3CS06</td>
<td>Programming Using Java- Lab</td>
<td>-</td>
<td>-</td>
<td>3</td>
</tr>
<tr>
<td>9</td>
<td>21PC3CS05</td>
<td>Data Structures-Lab</td>
<td>-</td>
<td>-</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>21PR3IN01</td>
<td>Evaluation of Summer Internship-I</td>
<td>-</td>
<td>-</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>TOTAL</strong></td>
<td><strong>21.5</strong></td>
<td><strong>370</strong></td>
<td><strong>630</strong></td>
</tr>
</tbody>
</table>

### Non Credit Courses

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Course code</th>
<th>Subject</th>
<th>Hours Per Week</th>
<th>Credits</th>
<th>Scheme of Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>L</td>
<td>T</td>
<td>P</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Int.</td>
<td>Ext.</td>
<td>Tot.</td>
</tr>
<tr>
<td>11</td>
<td>21MC3HS02</td>
<td>Environmental Science</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### IV – Semester (II – Year)

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Course code</th>
<th>Subject</th>
<th>Hours Per Week</th>
<th>Credits</th>
<th>Scheme of Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>L</td>
<td>T</td>
<td>P</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Int.</td>
<td>Ext.</td>
<td>Tot.</td>
</tr>
<tr>
<td>1</td>
<td>21ES4MT04</td>
<td>Matrix Computations and it’s Applications</td>
<td>3</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>21ES4CS03</td>
<td>Python Programming</td>
<td>3</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>21PC4CS08</td>
<td>Operating Systems</td>
<td>3</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>21PC4CM01</td>
<td>Introduction to Artificial Intelligence</td>
<td>3</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>21PC4CD02</td>
<td>Data Mining &amp; Data Analytics</td>
<td>3</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>21PC4CS11</td>
<td>Operating Systems-Lab</td>
<td>-</td>
<td>-</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>21PC4CM02</td>
<td>Artificial Intelligence-Lab</td>
<td>-</td>
<td>-</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>21ES4CS04</td>
<td>Python Programming-Lab</td>
<td>-</td>
<td>-</td>
<td>3</td>
</tr>
<tr>
<td>9</td>
<td>21PC4CS23</td>
<td>Data Mining Lab</td>
<td>-</td>
<td>-</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>21PR4CD01</td>
<td>Doing Engineering-I</td>
<td>-</td>
<td>-</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>TOTAL</strong></td>
<td><strong>22.5</strong></td>
<td><strong>300</strong></td>
<td><strong>700</strong></td>
</tr>
</tbody>
</table>
MATHEMATICAL AND STATISTICAL FOUNDATION
(Common to CSC/ CSD/CSM/CSO)

Pre-requisite: Basic knowledge of set and relations theory, permutations, combinations, Venn diagrams, mean, median, mode and testing of hypothesis of different samples

Course Objectives: To provide the student with
- Different random variables and various probability distribution functions.
  - The discrete and continuous of the Bernoulli, binomial, Poisson and the Normal distributions.
  - Measures of Central tendency and curve fitting by the method of least squares.
  - Concept of point and interval estimations and concept of test of hypothesis and test of significance.
  - How to apply the five-step test procedure for test of hypothesis concerning a population mean when the sample size is small.

MODULE I
Unit 1: PROBABILITY

Unit 2: RANDOM VARIABLES

MODULE II
Unit 1: DISCRETE DISTRIBUTIONS
Uniform, Binomial and, Poisson distribution and statistical constants of these distributions using moment generating function

Unit 2: CONTINUOUS DISTRIBUTIONS
Uniform Distribution, Exponential Distribution, Normal Distribution and statistical constants of these distributions using moment generating function

MODULE III
Unit 1: BIVARIATE DISTRIBUTION
Joint Probability distributions - Joint Probability mass function, joint probability density function, Marginal Distribution, Covariance of two random variables.

Unit 2: CORRELATION and REGRESSION
Karl Pearson coefficient of correlation, Rank correlation, Regression coefficient, Lines of regression.
MODULE IV
Unit 1: SAMPLING DISTRIBUTION:
Random Sampling, Some Important Statistics, Sampling Distributions, Sampling Distribution of Means, variance and the Central Limit Theorem.

Unit 2: ESTIMATION:
Estimating the Mean, Standard Error of a Point Estimate, Prediction Intervals, Tolerance Limits, Estimating the Variance, Estimating a Proportion for single mean, Difference between Two Means, between Two Proportions for Two Samples

MODULE V
Unit 1: GREATEST COMMON DIVISORS AND PRIME FACTORIZATION:
Greatest common divisors, The Euclidean algorithm, The fundamental theorem of arithmetic, Factorization of integers and the Fermat numbers

Unit 2: CONGRUENCES:
Introduction to congruences, Linear congruences, The Chinese remainder theorem, Systems of linear congruences.

Text Books:

Reference Books:

MOOC courses:
1. Probability: http://nptel.ac.in/courses/111105041/
2. Probability and Statistics: http://nptel.ac.in/courses/111105035/
3. Probability: https://nptel.ac.in/courses/111/102/111102111/

E-Books:
Course Outcomes: After learning the contents of this paper the student must be able to

CO1: Solve the probability and the probability distribution function of random variables.
CO2: Evaluate random processes which occur in engineering applications governed by the Binomial, Poisson, Normal and Exponential distributions.
CO3: Apply Inferential Statistics to make predictions or judgments about the population from which the sample data is drawn.
CO4: Apply the number theory concepts to cryptography.

<table>
<thead>
<tr>
<th>Course Outcomes (COs)</th>
<th>Program Outcomes (POs)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PO 1</td>
</tr>
<tr>
<td>CO1</td>
<td>3</td>
</tr>
<tr>
<td>CO2</td>
<td>3</td>
</tr>
<tr>
<td>CO3</td>
<td>3</td>
</tr>
<tr>
<td>CO4</td>
<td>3</td>
</tr>
</tbody>
</table>

CO-PO/PSO Mapping Chart
(3/2/1 indicates strength of correlation)
3 – High; 2 – Medium; 1 – Low
Object Oriented Programming through Java
(Common to CSE/CSM/CDS/CSC/CSO/EEE)

Prerequisites: Knowledge of Programming

Course Objectives:
1. Understand fundamental concepts and constructs of Java
2. Implement Different object-oriented Concepts in Java.
3. Write error free program using exception handling and multithreading
4. Develop GUI application

Course Outcomes:
1. Solve the given problem using OOPS technique.
2. Explain the concept of Package and Interfaces.
3. Implement Multi threading and Exception handling in java
4. Develop GUI based application using applet, awt, Event handling and swing.

Module I
History and Evolution of java: Java’s lineage, Java and internet, Byte code, Java buzzwords, Evolution of java.
Object oriented programming - data, types, variables, scope and life time of variables, arrays, operators, expressions, control statements, type conversion and casting, simple java programs, concepts of classes, objects, , methods, constructor, this and static keywords , garbage collection, overloading methods, parameter passing, access control, command line arguments, auto boxing, exploring String and String Buffer class
Inheritance: Types of inheritance, member access rules, super and final keywords, the object class and its methods, method overriding, dynamic method dispatch, abstract classes and methods.

Module II
Packages and Interfaces: Defining, Creating and Accessing a Package, understanding CLASSPATH, importing packages, Differences between classes and interfaces, defining an interface, implementing interface, applying interfaces, variables in interface and extending interfaces.
Exception handling: Concepts of exception handling, benefits of exception handling, exception hierarchy, usage of try, catch, throw, throws and finally, built in exceptions, creating own exceptions.

Module III
Multithreading: Differences between multi-threading and multi programming, thread life cycle, creating threads, thread priorities, synchronizing threads, inter thread communication.
I/O Streams: Types of Streams: Byte and character streams, File class, reading and writing files, reading and writing from console, Random access file, automatically close a file, serialization.

Module IV
Applets: Concepts of Applets, differences between applets and applications, life cycle of an applet, creating applets, passing parameters to applets.

Event Handling: Events, Event sources, Event classes, Event Listeners, Delegation event model, handling mouse and keyboard events, Adapter classes.

Module V

Text Books:

Reference Books:
1. An Introduction to JAVA Programming (Chapter 6) Author: Y.Daniel Liang, Publishers: Tata Mec-Hill.
3. Internet and Java Programming, R.KrishnaMurthu and S.Prabhu, New Age Publishers

Web Resources:
1. https://www.w3schools.com/java/

E-Books:
1. java 2 complete reference 9th ed Herbert Schildt.pdf

CO-PO/PSO Mapping:

<table>
<thead>
<tr>
<th>Course Outcomes (COs)</th>
<th>Program Outcomes (POs)</th>
<th>Program Specific Outcomes*</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PO 1</td>
<td>PO 2</td>
</tr>
<tr>
<td>CO1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO4</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>
Course Objective:
1. To introduce growing need of English Language Skills for Employability (ELSE).
2. To help the students understand the importance of inter-personal communication, team dynamics and role behaviour at the work place
3. To enhance their Behavioural skill and Entrepreneurship skills.

Module I - Wings of Fire
Unit 1: “Orientation” an excerpt from Wings of fire-An Autobiography of Abdul Kalam by Arun Tiwari.
Unit 2: Grammar
Vocabulary- Omission of Articles, Collective Nouns, Prepositions, Collocations.
Reading- Observation Passage, Survey Passage
Writing- Comprehension Passages based on graphs, charts & other illustrations.

Module II 5 Points Someone
Unit 1: The Gift an excerpt from 5 Points Someone by Chetan Bhagat
Unit 2: Grammar -
Vocabulary- Advanced Collocations, Proverbs, Idioms, One word Substitute
Reading – Complex passage, Reading Comparison,
Writing- Usage of Idioms and Proverbs in Passage

Module III – Wise Leaders Wanted & Shift Your Perspective: Connect to Your Noble Purpose
Unit 1: “Wise Leaders Wanted & Shift Your Perspective: Connect to Your Noble Purpose” an excerpt from From Smart to Wise: Acting and Leading with Wisdom Kaipa, Prasad, and Navi Radjou.
Unit 2: Grammar
Vocabulary- Technical vocabulary, Auxiliaries and Modals,
Reading- Technical Comprehension,
Writing- Creative Resume.

Module IV- Variation Under Nature
Unit 1: “Variation Under Nature” an excerpt from Origin of Species by Charles Darwin
Unit 2: Grammar
Vocabulary- Coherence-Cohesive devices, Figures of speech
Reading- Inferring Reading, Survey, Question, Read, Recite and Review (SQ3R)

Module V- Let's Build a Company: A Start-up Story Minus the Bullshit
Unit 1: Let's Build a Company: A Start-up Story Minus the Bullshit by Harpreet Grover and Vibhore Goyal
Unit 2: **Grammar**

- **Vocabulary** - Topic/Situation based Vocabulary, Tongue Twisters.
- **Reading** - Critical Reading of known/unknown passages

**Textbooks**

**References**
5. Murphy, R. *Intermediate English Grammar*.

**Course Outcomes:** After undergoing this course, the student will be able to;

**CO1:** Understand the concept and process of employability skills.

**CO2:** Demonstrate the employability skills in both verbal and non-verbal communications.

**CO3:** Apply skill identification strategies to bring out the results on social and industry demands.

**CO4:** Recognize right professional, Entrepreneurship skills and social ethical values.

<table>
<thead>
<tr>
<th>CO-PO Mapping Chart</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>(3/2/1 indicates strength of correlation) 3 – High; 2 – Medium; 1 – Low</strong></td>
</tr>
<tr>
<td><strong>Course Outcomes (COs)</strong></td>
</tr>
<tr>
<td><strong>PO 1</strong></td>
</tr>
<tr>
<td>CO1</td>
</tr>
<tr>
<td>CO2</td>
</tr>
<tr>
<td>CO3</td>
</tr>
<tr>
<td>CO4</td>
</tr>
</tbody>
</table>
DATASTRUCTURES

Prerequisite: Knowledge of C Programming

Course Objectives:
1: To make student knowledgeable in data structure concepts
2: To make students apply the data structure concepts in relevant applications
3: To make students apply the graph techniques in relevant applications
4: To make students determine the best sorting technique to process the data

Course Outcomes:
At the end of the course the students will be able to:
1: Identify the concepts of data structures in relevant applications
2: Apply the concept of data structures for the new situations
3: Apply the graph traversal techniques to solve map related problems
4: Analyze the best sorting technique pertaining to the area applied

Module 1:
Algorithm Analysis: Asymptotic analysis, simple justification techniques, Recursion, Illustrative examples, Analyzing Recursive algorithms, Designing Recursive algorithms.
Sorting: Bubble sort, selection sort, Insertion sort, Merge sort, Quick Sort, Randomized Quick sort. Comparison of sorting algorithms, Polynomials and Sparse Matrices.

Module 2:

Module 3:
Linked Lists – Singly Linked List, Implementation using arrays, Implementation using Queues, Circular Linked List- Doubly linked list- Implementation of doubly linked list, implementation using De queue.
Trees- General trees, Binary trees, Implementation using array and linked representation, tree traversals- in order, preorder, post order, Breadth-first traversal, Implementation, Application of tree traversals.

Module 4:
Priority Queue-ADT, Implementation, Heap, sorting with a priority Queue, Adaptable priority Queues Hash Tables and Skip Lists-Maps and Dictionaries, Hash tables, Sorted Maps, Skip lists, Sets, Multi sets and Multi maps
Graphs: Definitions, Terminologies, Matrix and Adjacency List Representation of Graphs, Traversal methods: Breadth First Search and Depth First Search.
Module 5:

Text Books:

Reference Books:
2. Data Structures using C by E. Balaguruswamy.
3. Data Structures, S. Lipscutz Schumaum; Outlines, TMH

CO-PO/PSO Mapping:

<table>
<thead>
<tr>
<th>CO-PO</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>PSO 1</th>
<th>PSO2</th>
<th>PSO3</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO-1</td>
<td>H</td>
<td>M</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO-2</td>
<td>M</td>
<td>M</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO-3</td>
<td>M</td>
<td>M</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO-4</td>
<td>M</td>
<td>M</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Database Management Systems

Prerequisite: Basics of computer programming language.

Course Objectives:
1. To enable students, define and describe basic concepts of database managements and applications.
2. To master the basics of SQL and design queries using SQL.
3. To provide students the theoretical concepts of data models and database design
4. To make students familiarize with relational model, relational algebra, transaction control and concurrency control.
5. To introduce storage structures and access techniques

Course Outcomes:
1. Describe fundamentals of DBMS, database design and normal forms.
2. Design SQL commands and queries for retrieval and management of data.
3. Summarize database storage structures and access techniques.
4. Understand basics of transaction processing and concurrency control.

Module I
Database System Applications: A Historical Perspective, File Systems versus a DBMS, the Data Model, Levels of Abstraction in a DBMS, Data Independence, Structure of a DBMS
Introduction to Database Design: Database Design and ER Diagrams, Entities, Attributes, and Entity Sets, Relationships and Relationship Sets, Additional Features of the ER Model, Conceptual Design with the ER Model

Module II
Relational Model: Integrity constraint over relations, enforcing integrity constraints, querying relational data, logical database design, introduction to views, destroying/altering tables and views.
Relational Algebra and Calculus: Selection and Projection, Set operations, Joins, Division, More examples on Algebra queries, Tuple relational Calculus, Domain Relational Calculus. Form of Basic SQL Query, Data Definition commands, Data manipulation commands.

Module III
Advanced SQL: SET operators, Data Time Functions, Aggregate Operators, Group by clause, Joins, Advanced Sub queries, Nested Queries, triggers and active data bases.

Module IV
Storage and Indexing: Data on External Storage, File Organization and Indexing, Cluster Indexes, Primary and Secondary Indexes, Index data Structures, Hash Based Indexing, Tree base Indexing, Comparison of File Organizations, Indexes and Performance Tuning
Tree Structured Indexing: Intuitions for tree Indexes, Indexed Sequential Access Methods
(ISAM), B+ Trees: A Dynamic Index Structure, Search, Insert, Delete, Duplicates and B+ Trees in Practice.

Module V
Transaction Management: The ACID Properties, Transactions and Schedules, Concurrent Execution of Transactions, Lock-Based Concurrency Control, Performance of Locking, Transaction Support in SQL, Introduction to Crash Recovery

Concurrency Control: 2PL, Serializability, and Recoverability, Introduction to Lock Management, Lock Conversions, Dealing with Deadlocks, Specialized Locking Techniques, Concurrency Control without Locking

Text Books:

Reference Books:

CO-PO/PSO Mapping:

<table>
<thead>
<tr>
<th>Course Outcomes (COs)</th>
<th>Program Outcomes (POs)</th>
<th>Program Specific Outcomes*</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PO 1</td>
<td>PO 2</td>
</tr>
<tr>
<td>CO1</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>CO2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>CO3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>CO4</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>
Database Management System lab

Prerequisite: Nil

Course Objectives:

1. Introduce ER data model, database design and normalization.
2. Learn SQL basics for data definition and data manipulation.

Course Outcomes:

1. Design database schema for a given application and apply normalization.
2. Perform operations on the database using SQL commands for data definition and data manipulation.
3. Develop solutions for database applications using procedures, cursors and triggers.
4. Implement joins on the tables.

List of Experiments:

The following programs are to be developed using My SQL / SQL / PL/SQL

**Roadway Travels:** "Roadway Travels" is in business since 1997 with several buses connecting different places in India. Its main office is located in Hyderabad.

The company wants to computerize its operations in the following areas:

- Reservations and Ticketing
- Cancellations

**Reservations & Cancellation:** Reservations are directly handled by booking office. Reservations can be made 30 days in advance and tickets issued to passenger. One Passenger/person can book many tickets (to his/her family). Cancellations are also directly handed at the booking office.

In the process of computerization of Roadway Travels you have to design and develop a Database which consists of the data of Buses, Passengers, Tickets, and Reservation and cancellation details. You should also develop query’s using SQL to retrieve the data from the database.

The above process involves many steps like 1. Analyzing the problem and identifying the Entities and Relationships, 2. E-R Model 3. Relational Model 4. Normalization 5. Creating the database 6. Querying. **Students are supposed to work on these steps week wise and finally create a complete “Database System” to Roadway Travels.** Examples are given at every experiment for guidance to students.
Experiment 1: E-R Model
Analyze the problem carefully and come up with the entities in it. Identify what data has to be persisted in the database. This contains the entities, attributes etc.

Identify the primary keys for all the entities. Identify the other keys like candidate keys, partial keys, if any.

Example:
**Entities:** 1. BUS 2. Ticket 3. Passenger
**Relationships:** 1. Reservation 2. Cancellation
**Primary Key Attributes:** 1. Ticket ID (Ticket Entity)
2. Passport ID (Passenger Entity)
3. Bus_NO (Bus Entity)

Apart from the above mentioned entities you can identify more. The above mentioned are few.

**Note:** The student is required to submit a document by writing the Entities and Keys to the lab teacher.

Experiment 2: Concept design with E-R Model
Relate the entities appropriately. Apply cardinalities for each relationship. Identify strong entities and weak entities (if any). Indicate the type of relationships (total / partial). Try to incorporate generalization, aggregation, specialization etc wherever required.

Example: E-R Diagram for bus

![E-R Diagram for bus](image)

**Note:** The student is required to submit a document by drawing the E-R Diagram to the lab teacher.
Experiment 3: Relational Model
Represent all the entities (Strong, Weak) in tabular fashion. Represent relationships in a tabular fashion. There are different ways of representing relationships as tables based on the cardinality. Represent attributes as columns in tables or as tables based on the requirement.

Different types of attributes (Composite, Multi-valued, and Derived) have different way of representation.

Example: The passenger tables look as below. This is an example. You can add more attributes based on your E-R model. This is not a normalized table.

<table>
<thead>
<tr>
<th>Name</th>
<th>Age</th>
<th>Sex</th>
<th>Address</th>
<th>Ticket_id</th>
<th>Passport ID</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: The student is required to submit a document by Represent relationships in a tabular fashion to the lab teacher.

Experiment 4: Normalization
Database normalization is a technique for designing relational database tables to minimize duplication of information and, in so doing, to safeguard the database against certain types of logical or structural problems, namely data anomalies. For example, when multiple instances of a given piece of information occur in a table, the possibility exists that these instances will not be kept consistent when the data within the table is updated, leading to a loss of data integrity. A table that is sufficiently normalized is less vulnerable to problems of this kind, because its structure reflects the basic assumptions for when multiple instances of the same information should be represented by a single instance only.

For the above table in the First normalization we can remove the multi valued attribute Ticket_id and place it in another table along with the primary key of passenger.

First Normal Form: The above table can be divided into two tables as shown below.

<table>
<thead>
<tr>
<th>Name</th>
<th>Age</th>
<th>Sex</th>
<th>Address</th>
<th>Passport ID</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Passport ID</th>
<th>Ticket_id</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

You can do the second and third normal forms if required. Anyhow Normalized tables are given at the end.
Experiment 5: Installation of Mysql / SQL and practicing DDL commands
Installation of MySql / SQL. In this session you will learn Creating databases, How to create tables, altering the database, dropping tables and databases if not required. You will also try truncate, rename commands etc.

Example for creation of a normalized “Passenger” table.
CREATE TABLE Passenger (  
    Passport_id INTEGER PRIMARY KEY,  
    Name VARCHAR (50) Not NULL,  
    Age Integer Not NULL,  
    Sex Char,  
    Address VARCHAR (50) Not NULL);  

Similarly create all other tables.

Note: Detailed creation of tables is given at the end.

Experiment 6: Practicing DML commands
DML commands are used to for managing data within schema objects. Some examples:
  - SELECT - retrieve data from the a database  
  - INSERT - insert data into a table  
  - UPDATE - updates existing data within a table  
  - DELETE - deletes all records from a table, the space for the records remain

Inserting values into “Bus” table:
Insert into Bus values (1234,‘hyderabad’, ‘tirupathi’);  
Insert into Bus values (2345,’hyderabad’, ’Banglore’);  
Insert into Bus values (23,’hyderabad’,’Kolkata’);  
Insert into Bus values (45,’Tirupathi,’Banglore’);  
Insert into Bus values (34,’hyderabad’,’Chennai’);  

Inserting values into “Passenger” table:
Insert into Passenger values (1, 45,’ramesh’, 45,’M’,’abc123’);  
Insert into Passenger values (2, 78,’geetha’, 36,’F’,’abc124’);  
Insert into Passenger values (45, 90,’ram’, 30,’M’,’abc12’);  
Insert into Passenger values (67, 89,’ravi’, 50,’M’,’abc14’);  
Insert into Passenger values (56, 22,’seetha’, 32,’F’,’abc55’);  

Few more Examples of DML commands:
Select * from Bus: (selects all the attributes and display)  
UPDATE BUS SET Bus No = 1 WHERE BUS NO=2;
Experiment 7: Querying
In this week you are going to practice queries (along with sub queries) using ANY, ALL, IN, Exists, NOT EXISTS, UNION, INTERSECT, Constraints etc.

Practice the following Queries:
1. Display unique PNR_no of all passengers.
2. Display all the names of male passengers.
3. Display the ticket numbers and names of all the passengers.
4. Find the ticket numbers of the passengers whose name start with ‘r’ and ends with ‘h’.
5. Find the names of passengers whose age is between 30 and 45.
6. Display all the passengers names beginning with ‘A’
7. Display the sorted list of passengers names

Experiment 8 and Experiment 9: Querying (continued…)
You are going to practice queries using Aggregate functions (COUNT, SUM, AVG, and MAX and MIN), GROUP BY, HAVING and Creation and dropping of Views.
1. Write a Query to display the Information present in the Passenger and cancellation tables. Hint: Use UNION Operator.
2. Display the number of days in a week on which the 9W01 bus is available.
3. Find number of tickets booked for each PNR_no using GROUP BY CLAUSE. Hint: Use GROUP BY on PNR_No.
4. Find the distinct PNR numbers that are present.
5. Find the number of tickets booked by a passenger where the number of seats is greater than 1. Hint: Use GROUP BY, WHERE and HAVING CLAUSES.
6. Find the total number of cancelled seats.

Experiment 10: Triggers
In this session you are going to work on Triggers. Creation of insert trigger, delete trigger, update trigger. Practice triggers using the above database.
Eg:
```
CREATE OR REPLACE TRIGGER SUN_TRI
AFTER INSERT OR UPDATE OR DELETE ON EMP
DECLARE
  DY VARCHAR2 (200);
BEGIN
  DY:=TO_CHAR(SYSDATE,'DY');
  IF DY='SUN' THEN
    RAISE_APPLICATION_ERROR (-20005,'TODAY IS SUNDAY TRANSACTION NOT ALLOWED
  END IF;
END;
```

Experiment 11: Procedures
In this session you are going to learn Creation of stored procedure, Execution of procedure and modification of procedure. Practice procedures using the above database.
Eg: CREATE PROCEDURE myProc()
```
BEGIN
  SELECT COUNT (Tickets) FROM Ticket WHERE age>=40;End;
```
Experiment 12: Cursors
In this session you need to do the following: Declare a cursor that defines a result set.
Open the cursor to establish the result set. Fetch the data into local variables as needed from the
cursor, one row at a time. Close the cursor when done.

Eg:
DECLARE
CURSOR A IS
SELECT * FROM EMP;
B A%ROWTYPE;
BEGIN
OPEN A;
LOOP
FETCH A INTO B;
EXIT WHEN A%NOTFOUND;
DBMS_OUTPUT.PUT_LINE('EMP NO IS ' || B.EMPNO);
DBMS_OUTPUT.PUT_LINE('ENAME IS ' || B.ENAME);
DBMS_OUTPUT.PUT_LINE('SAL IS ' || B.SAL);
DBMS_OUTPUT.PUT_LINE('MGR NO IS ' || B.MGR);
DBMS_OUTPUT.PUT_LINE('COMM IS ' || B.COMM);
DBMS_OUTPUT.PUT_LINE('HIREDATE IS ' || B.HIREDATE);
DBMS_OUTPUT.PUT_LINE('DEPTNO IS ' || B.DEPTNO);
DBMS_OUTPUT.PUT_LINE('EMP JOB IS ' || B.JOB);
DBMS_OUTPUT.PUT_LINE('*************************');
END LOOP;
CLOSE A;
END;

Text Books:
   3rd Edition

Reference Books:
1. Database Systems design, Implementation, and Management, Peter Rob & Carlos Coronel
   7th Edition.
2. Fundamentals of Database Systems, Elmasri Navrate, Pearson Education
3. Introduction to Database Systems, C. J. Date, Pearson Education

CO-PO/PSO Mapping Chart

<table>
<thead>
<tr>
<th>Course Outcomes (COs)</th>
<th>Program Outcomes (POs)</th>
<th>Program Outcomes*</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PO 1</td>
<td>PO 2</td>
</tr>
<tr>
<td>CO1</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>CO2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>CO3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>CO4</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>
English For Employability Lab
(Common to CSE/EEE/ECE/CSM/CSD/CSC/CSO/MECH)

Prerequisites:
1. Basic knowledge of interpersonal and communication skills.
2. Enriching the speaking and writing ability
3. Basic grammar rules of (LSRW)
4. Basic Spoken English skills

Course Objectives:
1. To improve the communication skills, body language, facial expression and gesture.
2. To be able to understand the concept of employability skills (Quantum dexterity) and enhancing ones’ behavior in the personal, professional and social forum.
3. To evaluate the LSRW (listening, speaking, reading and writing) through assessment.
4. To learn the basic grammar for improving spoken and written communication.
5. To become problem solver, analyze and apply critical and analytical skills.
6. To identify the Employability skills, assigning tasks (Group Discussion, JAM, Role play etc.,) for day today evaluation.

Activity 1- Narration (Historical places, events, Picture narration, Memorable incidents of life)
Self-Introduction, Daily Routine, Likes & Dislikes, Vocabulary, Triangular Activity (Person based- S-P), Imperatives & JAM
Targeted Skills- Listening- Speaking- Audio-Video clips

Activity 2: Quantum of Dexterity (QOD)-- Ability (Personal, Behavioural & Professional)
Request/Permission/Order, Survival kit, Career Objective Professional, Hidden Talents (Personal), Character Traits (Behavioural)
Targeted Skills- Reading-Writing – Concluding an open-ended Story, Creative Writing.

Activity 3: Critical & Analytical Skills
SWOC- (Social & Cultural, Political, Economic, Legal Impact, Technical, Nuances of Pronunciation, Voice Modulation, Neutralizing Mother Tongue Interference, Tongue Twisters for practice,
Targeted Skills- Writing SWOC, Self-Introduction, Exposure to a structured talk.

Activity 4: Flick Flow/Extempore
Targeted Skills- Speaking Skills

Activity 5: On Job Training
Formal & Informal communication, Resume E-mail Etiquette, Telephonic & Interview Etiquette, Situation based- Santa’s Bag, topic/case study-based Group Discussion (Job Consultancy/Role Play)
Targeted Skills- Listening-Writing-Speaking

Course Outcomes: After undergoing this course, the student will be able to;

CO1: Understand the explicit and implicit of importance of employability skills.

CO2: Demonstrate life skills like team work, learning skills, problem solving, attitude, adaptability and flexibility.

CO3: Apply critical and analytical skills to bring out the solution on problem/case study.

CO4: Recognize the need of appropriate words, Phrases & functional grammar and apply them in both spoken and written communication.

<table>
<thead>
<tr>
<th>CO-PO Mapping Chart</th>
</tr>
</thead>
<tbody>
<tr>
<td>(3/2/1 indicates strength of correlation) 3 – High; 2 – Medium; 1 – Low</td>
</tr>
<tr>
<td>Course Outcomes (COs)</td>
</tr>
<tr>
<td>CO1</td>
</tr>
<tr>
<td>CO2</td>
</tr>
<tr>
<td>CO3</td>
</tr>
<tr>
<td>CO4</td>
</tr>
</tbody>
</table>

3 – High; 2 – Medium; 1 – Low
Programming using Java Lab

Course Objectives:
1. To introduce object oriented concept for solving the real world problem
2. To introduce Exception Handling for detecting run time errors
3. To understand the utilization of multithreading concept
4. To build GUI Application using Swing, AWT, Applet

Course Outcomes:
1. Execute the given problem using java language.
2. Construct the program using oops concept.
3. Demonstrate the usage of multithreading and exception handling mechanisms
4. Implement simple GUI interfaces for a computer program to interact with users, and the event-based GUI handling principles.

List of Experiments:
1. A. Develop a java Program to find the roots of Quadratic equation.
   B. Develop a java Program to find the Fibonacci sequence.
2. A. Develop a Java program that prompts the user for an integer and then prints out all prime numbers up to that integer.
   B. Develop a java Program to multiply 2 given matrices.
3. A. Develop a java Program for sorting a given list of names in ascending order.
   B. Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named print Area (). Provide three classes named Rectangle, Triangle, and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method print Area () that prints the area of the given shape.
4. Develop a Java program to create an exception by name Minor Exception, which will be raised when the Age of the voter is less than 18 years.
5. Develop a Java program that implements a multithread application that has three threads. First thread generates random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number.
6. Develop a Java program that correctly implements producer consumer problem using the concept of inter thread communication.
7. A. Develop a java program that copies the content of one file to another.
   B. Develop a java Program to accept data from keyboard & develop it into a file.
8. A. Develop an applet in Java that receives an integer in one text field, and computes its factorial value and returns it in another text field, when the button named “Compute” is clicked.
   B. Develop a java program for passing parameters to applets.
9. Develop a Java program to demonstrate Mouse Listener, Mouse Motion Listener
10. Develop a Java program that simulates a traffic light. The program lets the user select one of three lights: red, yellow, or green with radio buttons. On selecting a button, an appropriate
message with “Stop” or “Ready” or “Go” should appear above the buttons in selected color. Initially, there is no message shown.

11. Develop a java Program to design a calculator for implementing basic functions like +, *, -, / using grid layout.

12. Develop an applet that moves the character up, down, left and right when the appropriate arrows are pressed.

CO-PO/PSO Mapping:

<table>
<thead>
<tr>
<th>Course Outcomes (COs)</th>
<th>Program Outcomes (POs)</th>
<th>Program Specific Outcomes*</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P O 1</td>
<td>P O 2</td>
</tr>
<tr>
<td>CO1</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>CO2</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>CO3</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>CO4</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>
DATA STRUCTURES LAB

Pre-requisite: Programming in C and Problem Solving

Course Objectives:

1. Understand various data representation techniques in the real world.
2. Implement linear and non-linear data structures.
3. Analyze various algorithms based on their time and space complexity.
4. Develop real-time applications using suitable data structure.
5. Identify suitable data structure to solve various computing problems.

Course Outcomes:

1. Identify the appropriate data structures and algorithms for solving real world problems
2. Implement various kinds of searching and sorting techniques.
3. Implement data structures such as stacks, queues, search trees, and hash tables to solve various computing problems
4. Choose appropriate traversal method to find shortest distance

List of Experiments:

1. Develop a program to read a linear list of items and store it in an array.
   a. Copy the contents from one array to another array
   b. Copy the contents from one array to another array in reverse order
   c. Delete the duplicate elements from an array.

2. Develop a program to implement various sorting techniques:
   a. Insertion sort
   b. Selection Sort
   c. Bubble Sort
   d. Merge Sort
   e. Quick Sort

3. a. Develop a program to create a stack and perform various operations on it.
   b. Develop a program to create a queue and perform various operations on it.

4. Develop programs for the following:
   a. Uses Stack operations to convert infix expression into postfix expression.
   b. Uses Stack operations for evaluating the postfix expression.

5. Design a program to create a singly linked list for the following operations
   a. Insert a Node at Beginning, at Ending and at a given Position
   b. Delete a Node at Beginning, at Ending and at a given Position
   c. Search, Count the Number of Nodes and Display
6. Design a program to create a doubly linked list for the following operations
   a. Insert a Node at Beginning, at Ending and at a given Position
   b. Delete a Node at Beginning, at Ending and at a given Position
   c. Search, Count the Number of Nodes and Display

7. Design a program to create a binary tree and perform various traversals.
   a. In order
   b. Pre Order
   c. Post Order

8. Develop a program to Perform Linear Search and Binary Search on a list stored in an array.

9. Design a program to create a Hash table for the following operations
   a. Insert
   b. Delete
   c. Search

10. Develop programs to implement the following graph traversal algorithms:
    a. Depth first search.
    b. Breadth first search.

Text/Reference Books:
2. Data structures using C - A.S. Tanenbaum, Y. L. pansam, and MJ Augestien, PHI Pearson Education

CO-PO/PSO Mapping Chart:

<table>
<thead>
<tr>
<th>Course Outcome s (COs)</th>
<th>Program Outcomes (POs)</th>
<th>Program Specific Outcomes*</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PO 1</td>
<td>PO 2</td>
</tr>
<tr>
<td>CO1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(3/2/1 indicates strength of correlation)
3–High; 2–Medium; 1-Low
ENVIRONMENTAL SCIENCE

(Common to CSE/EEE/ECE/MECH/CSM/CDS/CSC/CSO)

Course Objectives:
1. Understanding the importance of ecological balance for sustainable development.
2. To educate students about natural resources and their exploitation
3. Understanding the concepts of green chemistry and its applications.

Module I ECOSYSTEMS AND ECOLOGY

UNIT-1: Ecosystem
Definition, Scope, and Importance of ecosystem. Classification, structure, and function of an ecosystem, Food chains, food webs, and ecological pyramids.

UNIT-2: Ecology
Flow of energy, Biogeochemical cycles, Bioaccumulation, Biomagnification, ecosystem value, services and carrying capacity, Field visits.

Module II NATURAL RESOURCES

UNIT-1: Classification of Resources
Living and Non-Living resources, water resources: use and over utilization of surface and ground water, floods and droughts, Dams: benefits and problems. Mineral resources: use and exploitation, environmental effects of extracting and using mineral resources,

UNIT-2: Land and Energy resources
Land resources: Forest resources, Energy resources: growing energy needs, renewable and non-renewable energy sources, use of alternate energy source, case studies.

Module III BIODIVERSITY AND BIOTIC RESOURCES

UNIT-1: Biodiversity
Introduction, Definition, genetic, species and ecosystem diversity. Value of biodiversity; consumptive use, productive use, social, ethical, aesthetic and optional values. India as a mega diversity nation, Hot spots of biodiversity. Field visit

UNIT-2: Biotic Resources
Module IV ENVIRONMENTAL POLLUTION AND SOLID WASTE

UNIT-1: Environmental pollution


UNIT-2: Global Environmental Issues and Global Efforts


Module V Green Chemistry & Hazardous chemicals

UNIT-1: Green Chemistry
Introduction & Principles, green solutions for chemical energy storage, green chemistry solutions will be discussed within the fields of Chemical production: choice of feedstock, solvents, catalysts, synthesis routes including microwave and ultrasonic assisted synthesis.

UNIT -2: Hazardous Chemicals
Classification of hazardous chemicals, transportation of hazardous chemicals, Hazchem code, Storage and handling of hazardous substances, Emergency preparedness (on site & offsite), Safety audit, Concept of fire and explosion, Major accidents involving hazardous substances.

Text Books:

1. Textbook of Environmental Studies for Undergraduate Courses by Erach Bharucha for University Grants Commission.
2. Environmental Studies by R. Rajagopalan, Oxford University Press.

Course Outcomes:

CO-1: Understand the importance of ecosystem and ecological balance in conservation of biodiversity.

CO-2: Understand the concepts of natural resources and its exploitation.

CO-3: Explain the control of pollution for sustainable environment.

CO-4: Explain the concepts of green chemistry and its applications.
## CO-PO/PSO Mapping Chart

(3/2/1 indicates strength of correlation)

3 – High; 2 – Medium; 1 - Low

<table>
<thead>
<tr>
<th>Course Outcomes (COs)</th>
<th>Program Outcomes (POs)</th>
<th>Program Specific Outcomes *</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P O 1</td>
<td>P O 2</td>
</tr>
<tr>
<td>CO1</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>CO2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>CO3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>CO4</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>
Pre-requisite: Basic knowledge of matrix theory.

Course Objectives: To provide the student with

- Vector spaces.
- Eigen values and Eigen vectors.
- Solution of linear system by Jacobi, Gauss-Seidel, LU decomposition Gaussian elimination and Least square methods.
- Householder transformation, QR factorization and Sensitivity analysis.
- Singular Value Decomposition.
- Moore-Penrose inverse.
- Reduction to Heisenberg and tridiagonal forms and bi diagonal form, Power, inverse power and Rayleigh quotient iterations.
- Sensitivity analysis of singular values and singular vectors, conjugate gradient method.

MODULE I
VECTOR SPACES
Unit 1:
Introduction, Vector spaces, sub-spaces, spanning sets, basis and dimension,
Unit 2:
Linear transformations, matrix of linear transformations, eigen values and eigen vectors.

MODULE II
LINEAR SYSTEM
Unit 1:
Matrix norm, Sensitivity analysis and condition numbers, Linear systems, Jacobi, Gauss-Seidel and successive over relaxation methods.
Unit 2:
LU decompositions, Gaussian elimination with partial pivoting, Banded systems, positive definite systems.
MODULE III
Unit 1:  
Cholesky decomposition – sensitivity analysis, Gram- Schmid orthonormal process, Householder transformation, QR factorization, stability of QR factorization.
Unit 2:  
Solution of linear least squares problems, normal equations, singular value decomposition (SVD).

MODULE IV
Unit 1:  
Unit 2:  
Reduction to Heisenberg and tridiagonal forms; Power, inverse power and Rayleigh quotient iterations.

MODULE V
Unit 1:  
Explicit and implicit QR algorithms for symmetric and non-symmetric matrices.
Unit 2:  
Reduction to bi diagonal form, Sensitivity analysis of singular values and singular vectors, conjugate gradient method.

Text Books:

Reference Books:

MOOC courses:
1. https://nptel.ac.in/courses/111/102/111102152/
E- Books:

1. Matrix Computations by Gene H. Golub, Charles F. Van Loan
   [https://1lib.in/book/2190556/88c37f](https://1lib.in/book/2190556/88c37f)
2. Numerical linear algebra and applications by Biswa Nath Datta
   [https://1lib.in/book/5259166/c11b06](https://1lib.in/book/5259166/c11b06)
   [https://1lib.in/book/942630/262edc](https://1lib.in/book/942630/262edc)
4. Numerical Linear Algebra by L N Trefethen, David Bau III
   [https://1lib.in/book/2464030/a9a674](https://1lib.in/book/2464030/a9a674)

Course Outcomes: After learning the contents of this paper the student must be able to

CO1: Determine Eigen values and Eigen vectors of a matrix which arise in several engineering applications.

CO2: Apply various Matrix Techniques to Industrial Data Analysis.

CO3: Determine pseudo inverse of a matrices by using Singular Value Decomposition.

CO4: Understand the QR factorization and Sensitivity analysis of linear system of equations.

<table>
<thead>
<tr>
<th>Course Outcomes (COs)</th>
<th>Program Outcomes (POs)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PO 1</td>
</tr>
<tr>
<td>CO1</td>
<td>3</td>
</tr>
<tr>
<td>CO2</td>
<td>3</td>
</tr>
<tr>
<td>CO3</td>
<td>3</td>
</tr>
<tr>
<td>CO4</td>
<td>3</td>
</tr>
</tbody>
</table>

CO-PO/PSO Mapping Chart
(3/2/1 indicates strength of correlation)
3 – High; 2 – Medium; 1 – Low
Subject Code: 21ES4CS03

Python Programming
(Common To All Branches)

Pre-requisite: Programming for Problem solving Using C

Course Objectives:
1. Acquire programming skills in core Python.
2. Acquire Object-oriented programming skills in Python.
3. Develop the skill of designing graphical-user interfaces (GUI) in Python.
4. Develop the ability to write database applications in Python.

Course Outcomes:
1. Understand operators, precedence of operators, associatively while evaluating expressions in program statements.
2. Distinguish the capabilities of procedural as well as object-oriented programming in Python and demonstrate the same in real world scenario.
3. Demonstrate indexing and slicing mechanisms for extracting a portion of data in a sequence.
4. Extend the knowledge of Python programming to build successful career in software development.

Module I
Python Basics, Objects- Python Objects, Standard Types, Other Built-in Types, Internal Types, Standard Type Operators, Standard Type Built-in Functions, Categorizing the Standard Types, Unsupported Types
Numbers - Introduction to Numbers, Integers, Floating Point Real Numbers, Complex Numbers, Operators, Sequences - Strings, Lists, and Tuples, Mapping and Set Types

Module II
Conditional Statements: Selection/Conditional Branching Statements: if, if-else, nested if, if-elif-else statement(s).
Iterative Statements – while and for loop, Nested loops, break and continue statement, pass Statement, else Statement used with loops.

Module III
Functions: Communicating with functions, Variable Scope and lifetime, return statement, Types of arguments, Lambda functions, and Recursive functions
Classes and Objects – Defining Classes, Creating Objects, Data Abstraction and Hiding through Classes, Class Method and self-Argument, Class variables and Object variables, __init__() and __del__() method.

Module IV
Exceptions: Exceptions in Python, Detecting and Handling Exceptions, Context Management, , Raising Exceptions, Related Modules
Regular Expressions: Introduction, Special Symbols and Characters, Res and Python
Module V

**Multithreaded Programming:** Introduction, Threads and Processes, Python, Threads, and the Global Interpreter Lock, Thread Module, Threading Module, Related Modules

**GUI Programming:** Introduction, Tkinter and Python Programming, Brief Tour of Other GUIs, Related Modules and Other GUIs

**Text Books:**

**Reference Books:**

**CO-PO/PSO Mapping**

<table>
<thead>
<tr>
<th>CO-PO/PSO Mapping Chart</th>
</tr>
</thead>
<tbody>
<tr>
<td>(3/2/1 indicates strength of correlation)</td>
</tr>
<tr>
<td>3 – High; 2 – Medium; 1 - Low</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Course Outcomes (COs)</th>
<th>Program Outcomes (POs)</th>
<th>Program Specific Outcomes*</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PO 1</td>
<td>PO 2</td>
</tr>
<tr>
<td>CO1</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>CO2</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>CO3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3 – High; 2 – Medium; 1 - Low
OPERATING SYSTEMS

Course Objective
1. To understand the components, operations of the operating system
2. To interpret the scheduling policies and memory management issues
3. To understand the process concurrency and synchronization
4. To understand the concept of file management

Course Outcomes:
At the end of the course student will be able to
1. Understand the structure of Operating System and its architecture
2. Apply the scheduling strategies for real time implementations
3. Illustrate synchronization problems, deadlock and its techniques
4. Infer Security and Protection Mechanism

Module I
Operating System Introduction: Operating Systems objectives and functions, Computer System Architecture, OS Structure, OS Operations,

Module II

Module III
Memory Management and Virtual Memory - Logical & Physical Address Space, Swapping, Contiguous Allocation,

Module IV
Mass Storage Structure – Overview of Mass Storage Structure, Disk Structure, Disk Attachment, Disk Scheduling, Disk Management, Swap space Management
Module V

**Deadlocks** - System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection and Recovery from Deadlock.


**TEXT BOOKS:**


**REFERENCE BOOKS:**


**CO-PO &PSO Mapping:**

<table>
<thead>
<tr>
<th>Course Name - Course Outcomes / Program Outcomes</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>PSO 1</th>
<th>PSO 2</th>
<th>PSO 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO1</td>
<td>M</td>
<td>L</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO2</td>
<td>H</td>
<td>H</td>
<td>M</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO3</td>
<td>M</td>
<td>H</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO4</td>
<td>M</td>
<td>L</td>
<td>L</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
INTRODUCTION TO ARTIFICIAL INTELLIGENCE

Course Objectives

1. To provide a strong foundation of fundamental concepts in artificial intelligence.
2. To provide a basic exposition to the goals and methods of Artificial intelligence.
3. To provide the exposition to uncertainty management, decision making and learning methods.
4. To provide different knowledge representation, reasoning, and learning techniques.

Course Outcomes

1. Discuss basic concepts of Artificial Intelligence, AI principles, AI Task domains and application.
2. Apply searching techniques, constraint satisfaction problem and game playing techniques which involve perception, reasoning and learning.
3. Explain working of uncertainty management, decision making and learning methods.
4. Apply different knowledge representation, reasoning, and learning techniques to real world problems.

MODULE-I
Introduction: Artificial Intelligence, AI Problems, AI Techniques, the Level of the Model, Criteria for Success. Problem Space and Search, Defining the Problem as a State Space Search, Problem Characteristics; Tic-Tac-Toe Problem, Production Systems.

Basic Search Techniques: Solving Problems by searching; Issues in The Design of Search Programs; Uniform search strategies; Breadth first search, depth first search, depth limited search, bidirectional search, Best First search, comparing search strategies in terms of complexity.

MODULE-II
Special Search Techniques: Heuristic Search, greedy best first search, A* search Problem Reduction, AO*Algorithm; Hill climbing search, Simulated Annealing search; Genetic Algorithm; Constraint Satisfaction Problems; Adversarial search, Games, Optimal decisions and strategies in games, Minimax search, Alpha, beta pruning.


MODULE-III

Structured Representations of Knowledge: Semantic Nets, Partitioned Semantic Nets, Frames, Conceptual Dependency, Conceptual Graphs, Scripts, CYC.
MODULE-IV


Statistical Reasoning: Bayes Theorem, Certainty Factors and Rule-Based Systems, Bayesian Probabilistic Inference, Bayesian Networks, Dempster-Shafer Theory.


MODULE-IV


Learning: Types of learning, general learning model, Learning by induction; generalization, specialization, example of inductive learner.

Text Book:
1. Artificial Intelligence, George F Luger, Pearson Education Publications

References:
1. Introduction To Artificial Intelligence & Expert Systems, Patterson, PHI
3. Artificial Intelligence: A modern Approach, Russell and Norvig, Printice Hall

<table>
<thead>
<tr>
<th>CO-PO/PSO Mapping Chart</th>
</tr>
</thead>
<tbody>
<tr>
<td>(3/2/1 indicates strength of correlation) 3 – High; 2 – Medium; 1 – Low</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CO/PO</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>PSO 1</th>
<th>PSO2</th>
<th>PSO3</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO-1</td>
<td></td>
<td></td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO-2</td>
<td></td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO-3</td>
<td></td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO-4</td>
<td></td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Data Mining & Data Analytics

Course Outcomes:
1. Understand Data Mining Principles.
2. Understand the impact of data analytics for business decisions and strategy.
3. Apply the Association, Clustering rules for mining the data.
4. Design various classification techniques.

Module 1:
Introduction to Data warehouse, Difference between operational database systems and data warehouses. Data warehouse Characteristics, Data warehouse Architecture.
Introduction, What is Data Mining, Definition, KDD, Challenges, Data Mining Tasks, Data Preprocessing, Data Cleaning, Missing data, Dimensionality Reduction, Feature Subset Selection, Discretization and Binarization, Data Transformation; Measures of Similarity and Dissimilarity – Basics.

Module 2:

Module 3:
Regression – Concepts, Blue property assumptions, Least Square Estimation, Variable Rationalization, and Model Building etc.
Logistic Regression: Model Theory, Model fit Statistics, Model Construction, Analytics applications to various Business Domains etc.

Module 4:
Association Rule Mining & Classification: Mining Frequent Patterns– Associations and correlations – Mining Methods– Mining Various kinds of Association Rules– Correlation Analysis.
Classification: Classification and Prediction – Basic concepts–Decision tree induction– Bayesian classification, Lazy learner.

Module 5:
Hierarchical Methods– Density–Based Methods, Grid–Based Methods, Outlier Analysis.

TEXT BOOKS:
1. Data Mining – Concepts and Techniques – Jiawei Han & Micheline Kamber, 3rd Edition Elsevier.
2. Data Mining Techniques – Arun K Pujari

REFERENCE BOOK:
1. Introduction to Data Mining, Tan, Steinbach and Kumar, Addision Wisley, 2006.
2. Data Mining Analysis and Concepts, M. Zaki and W. Meira
OPERATING SYSTEM LAB

Course Objective
1. To do the implementation of operating system concepts
2. To write the code to Implement and modify various concepts in operating systems

Course Outcomes:
1. Implement Inter process communication between two processes
2. Design and solve Synchronization problems
3. Implement scheduling, deadlock and file management
4. Simulate and implement operating system concepts such as memory management

List of Programs:
1. Simulate the following CPU scheduling algorithms
   a) Round Robin b) SJF c) FCFS d) Priority
2. Simulate the following CPU scheduling algorithms
   a) FCFS b) Priority
3. Simulate all file allocation strategies
   a) Sequential b) Indexed c) Linked
4. Simulate MVT and MFT
5. Simulate all File Organization Techniques
   a) Single level directory b) Two level
6. Simulate all File Organization Techniques
   a) Hierarchical b) DAG
7. Simulate Bankers Algorithm for Dead Lock Avoidance
8. Simulate Bankers Algorithm for Dead Lock Prevention
9. Simulate all page replacement algorithms
   a) FIFO b) LRU c) LFU Etc.
10. Simulate Paging Technique of memory management.

Text Books:

CO-PO &PSO Mapping

<table>
<thead>
<tr>
<th>CO-PO</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>PSO 1</th>
<th>PSO 2</th>
<th>PSO 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO-1</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO-2</td>
<td>H</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO-3</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO-4</td>
<td>M</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
ARTIFICIAL INTELLIGENCE LAB

Course Objectives
1. To study the applications of AI and agent-based approach to AI.
2. To study first-order predicate calculus, logical reasoning and problem-solving techniques.
3. To study and discuss various techniques and algorithms of AI used in general problem solving, optimization problems, constraint satisfaction problems.
4. To familiarize students with various sub-areas of AI, such as expert systems.

Course Outcomes
1. Apply artificial intelligence and its characteristics into its application areas.
2. Formulate real-world problems as state space problems, optimization problems or constraint satisfaction problems.
3. Apply appropriate algorithms and AI techniques to solve complex problems.
4. Design an expert system by using appropriate tools and techniques.

LIST OF PROGRAMS (can be implemented using Python or Lisp):
1. Implementation of DFS for water jug problem.
5. Implementation of Hill Climbing Algorithm.
6. Implementation of Hill-climbing to solve 8-Puzzle Problem.
10. Implementation Expert System with backward chaining.

CO-PO/PSO Mapping Chart
(3/2/1 indicates strength of correlation) 3 – High; 2 – Medium; 1 – Low

<table>
<thead>
<tr>
<th>CO/PO</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>PSO1</th>
<th>PSO2</th>
<th>PSO3</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO-1</td>
<td></td>
<td></td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO-2</td>
<td></td>
<td></td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO-3</td>
<td></td>
<td></td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO-4</td>
<td></td>
<td></td>
<td>3</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Text Book:
1. Practical Workbook Artificial Intelligence and Soft Computing for beginners, Anindita Das Bhattacharjee, Shroff Publisher-X Team Publisher

References:
Python Programming Lab
(Common To All Branches)

Prerequisite: Basic Programming Skills.

Course Objectives:
To learn the fundamentals of computers.

1. Acquire programming skills in core Python.
2. Acquire Object-oriented programming skills in Python.
3. Develop the skill of designing graphical-user interfaces (GUI) in Python.
4. Develop the ability to write database applications in Python.

Course Outcomes:

1. Acquire programming skills in core Python.
2. Acquire Object-oriented programming skills in Python.
3. Develop the skill of designing graphical-user interfaces (GUI) in Python.
4. Develop the ability to write database applications in Python.

List of Experiments:

Experiment – 1: OPERATORS
a. Read a list of numbers and write a program to check whether a particular element is present or not using membership operators.
b. Read your name and age and write a program to display the year in which you will turn 100 years old.

Experiment – 2: CONTROL STRUCTURES
a. Read your email id and write a program to display the no of vowels, consonants, digits and white spaces in it using if…elif…else statement.
b. Write a Program to find the sum of a Series 1/1! + 2/2! + 3/3! + 4/4! +……+ n/n!.
   (Input :n = 5, Output : 2.70833)

Experiment – 3: LIST
a. Read a list of numbers and print the numbers divisible by x but not by y (Assume x = 4 and y = 5).
b. Read a list of numbers and print the sum of odd integers and even integers from the list.(Ex: [23, 10, 15, 14, 63], odd numbers sum = 101, even numbers sum = 24)

Experiment – 4: TUPLE
a. Given a list of tuples. Write a program to find tuples which have all elements divisible by K from a list of tuples. test_list = [(6, 24, 12), (60, 12, 6), (12, 18, 21)], K = 6, Output : [(6, 24, 12), (60, 12, 6)]
Experiment – 5: SET
a. Write a program to generate and print a dictionary that contains a number (between 1 and n) in the form (x, x*x).
b. Write a program to perform union, intersection and difference using Set A and Set B.

Experiment – 6: DICTIONARY
a. Write a program to do the following operations:
   i. Create a empty dictionary with dict( ) method
   ii. Add elements one at a time
   iii. Update existing key’s value
   iv. Access an element using a key and also get( ) method
   v. Deleting a key value using del( ) method
b. Write a program to create a dictionary and apply the following methods:
   i. pop( ) method
   ii. pop item( ) method
   iii. clear( ) method

Experiment – 7: STRINGS
a. Given a string, write a program to check if the string is symmetrical and palindrome or not. A string is said to be symmetrical if both the halves of the string are the same and a string is said to be a palindrome string if one half of the string is the reverse of the other half or if a string appears same when read forward or backward.
b. Write a program to read a string and count the number of vowel letters and print all letters except 'e' and 's'.

Experiment – 8: USER DEFINED FUNCTIONS
a. Write a function merge_dict(dict1, dict2) to merge two Python dictionaries.
b. Given a list of n elements, write a linear_search() function to search a given element x in a list.

Experiment – 9: BUILT-IN FUNCTIONS
a. Write a program to demonstrate the working of built-in statistical functions mean ( ), mode ( ), median ( ) by importing statistics library.
b. Write a program to demonstrate the working of built-in trigonometric functions sin ( ), cos ( ), tan ( ), hypot ( ), degrees ( ), radians ( ) by importing math module.

Experiment – 10: CLASS AND OBJECTS
a. Write a program to create a Bank Account class. Your class should support the following methods for
   i) Deposit
   ii) Withdraw
   iii) Get Balance
   iv) Pin Change
b. Write a program to create an employee class and store the employee name, id, age, and salary using the constructor. Display the employee details by invoking employee_info () method and also using dictionary (__dict__).
### Reference Books:


### CO-PO/PSO Mapping Chart:

<table>
<thead>
<tr>
<th>Course Outcomes (COs)</th>
<th>Program Outcomes (POs)</th>
<th>Program Specific Outcomes*</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PO 1</td>
<td>PO 2</td>
</tr>
<tr>
<td>CO1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>CO2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>CO3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

CO-PO/PSO Mapping Chart  
(3/2/1 indicates strength of correlation)  
3 – High; 2 – Medium; 1 - Low
DATA MINING LAB

Course Objective

- To obtain practical experience using data mining techniques on real world data sets.
- Emphasize hands-on experience working with all real data sets.

Course Outcomes:

1: Build a data warehouse and query it using open-source tools.
2: Evaluate various data mining tasks using a data mining toolkit (such as weka) and visualize the results.
3: Determine the working of algorithms for data mining tasks such as association rule mining, classification, and clustering.
4: Implement classical models and algorithms in data mining

List of Programs:

Credit Risk Assessment

Description:
The business of banks is making loans. Assessing the credit worthiness of an applicant is of crucial importance. You have to develop a system to help a loan officer decide whether the credit of a customer is good, or bad. A bank’s business rules regarding loans must consider two opposing factors. On the one hand, a bank wants to make as many loans as possible. Interest on these loans is the bank’s profit source. On the other hand, a bank cannot afford to make too many bad loans. Too many bad loans could lead to the collapse of the bank. The bank’s loan policy must involve a compromise: not too strict, and not too lenient. To do the assignment, you first and foremost need some knowledge about the world of credit. You can acquire such knowledge in a number of ways.

i. Knowledge Engineering. Find a loan officer who is willing to talk. Interview her and try to represent her knowledge in the form of production rules.
ii. Find some training manuals for loan officers or perhaps a suitable textbook on finance. Translate this knowledge from text form to production rule form.
iii. Common sense. Imagine yourself as a loan officer and make up reasonable rules which can be used to judge the credit worthiness of a loan applicant.
iv. Case histories. Find records of actual cases where competent loan officers correctly judged when, and when not to, approve a loan application.

The German Credit Data:
Actual historical credit data is not always easy to come by because of confidentiality rules. Here is one such dataset, consisting of 1000 actual cases collected in Germany. credit dataset (original) Excel spreadsheet version of the German credit data.
In spite of the fact that the data is German, you should probably make use of it for this
A few notes on the German dataset

- DM stands for Deutsche Mark, the unit of currency, worth about 90 cents Canadian (but looks and acts like a quarter).
- owns telephone. German phone rates are much higher than in Canada so fewer people own telephones.
- foreign worker. There are millions of these in Germany (many from Turkey). It is very hard to get German citizenship if you were not born of German parents.
- There are 20 attributes used in judging a loan applicant. The goal is the classify the applicant into one of two categories, good or bad.

Subtasks: (Turn in your answers to the following tasks)

1. List all the categorical (or nominal) attributes and the real-valued attributes separately.
2. What attributes do you think might be crucial in making the credit assessment? Come up with some simple rules in plain English using your selected attributes.
3. One type of model that you can create is a Decision Tree - train a Decision Tree using the complete dataset as the training data. Report the model obtained after training.
4. Suppose you use your above model trained on the complete dataset, and classify credit good/bad for each of the examples in the dataset. What % of examples can you classify correctly? (This is also called testing on the training set) Why do you think you cannot get 100% training accuracy?
5. Is testing on the training set as you did above a good idea? Why or Why not?
6. One approach for solving the problem encountered in the previous question is using cross-validation? Describe what is cross-validation briefly. Train a Decision Tree again using cross-validation and report your results. Does your accuracy increase/decrease? Why?
7. Check to see if the data shows a bias against “foreign workers” (attribute 20), or “personal-status” (attribute 9). One way to do this (perhaps rather simple minded) is to remove these attributes from the dataset and see if the decision tree created in those cases is significantly different from the full dataset case which you have already done. To remove an attribute you can use the preprocess tab in Weka’s GUI Explorer. Did removing these attributes have any significant effect? Discuss.
8. Another question might be, do you really need to input so many attributes to get good results? Maybe only a few would do. For example, you could try just having attributes 2, 3, 5, 7, 10, 17 (and 21, the class attribute (naturally)). Try out some combinations. (You had removed two attributes in problem 7. Remember to reload the arff data file to get all the attributes initially before you start selecting the ones you want.)
9. Sometimes, the cost of rejecting an applicant who actually has a good credit (case 1) might be higher than accepting an applicant who has bad credit (case 2). Instead of counting the misclassifications equally in both cases, give a higher cost to the first case (say cost 5) and lower cost to the second case. You can do this by using a cost matrix in Weka. Train your Decision Tree again and report the Decision Tree and cross-validation results. Are they significantly different from results obtained in
problem 6 (using equal cost)?
10. Do you think it is a good idea to prefer simple decision trees instead of having long complex decision trees? How does the complexity of a Decision Tree relate to the bias of the model?
11. You can make your Decision Trees simpler by pruning the nodes. One approach is to use Reduced Error Pruning - Explain this idea briefly. Try reduced error pruning for training your Decision Trees using cross-validation (you can do this in Weka) and report the Decision Tree you obtain? Also, report your accuracy using the pruned model. Does your accuracy increase?
12. (Extra Credit): How can you convert a Decision Trees into “if-then- else rules”. Make up your own small Decision Tree consisting of 2-3. levels and convert it into a set of rules. There also exist different classifiers that output the model in the form of rules - one such classifier in Weka is rules. PART, train this model and report the set of rules obtained. Sometimes just one attribute can be good enough in making the decision, yes, just one! Can you predict what attribute that might be in this dataset? OneR classifier uses a single attribute to make decisions (it chooses the attribute based on minimum error). Report the rule obtained by training a oneR classifier. Rank the performance of j48, PART and oneR.

Note: The above experiments can be implemented using weka tool.

References:
1. Mentor lecture on Decision Trees
2. Andrew Moore’s Data Mining Tutorials (See tutorials on Decision Trees and Cross Validation)
3. Decision Trees (Source: Tan, MSU)
4. Tom Mitchell’s book slides (See slides on Concept Learning and Decision Trees)
5. Introduction to Weka

CO-PO & PSO Mapping:

<table>
<thead>
<tr>
<th>CO-PO</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>PSO1</th>
<th>PSO2</th>
<th>PSO3</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO-1</td>
<td>L</td>
<td></td>
<td></td>
<td>L</td>
<td>H</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO-2</td>
<td>L</td>
<td></td>
<td></td>
<td>L</td>
<td>H</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO-3</td>
<td>L</td>
<td></td>
<td></td>
<td>L</td>
<td>H</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO-4</td>
<td>L</td>
<td></td>
<td></td>
<td>L</td>
<td>H</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
DOING ENGINEERING-I

Pre-requisite: Basic Knowledge on Computers and C language

Course Objectives:
1) To Provide knowledge of different Smart System applications.
2) To provide knowledge of Arduino boards, Arduino as IDE, programming language, platform and basic components.
3) Develop skills to design and implement various smart system application.
4) To know the basics of micro-controllers and sensors very quickly and can start building prototype.

Course Outcomes:
1) Able to Design Smart systems applications.
2) Understand about any new IDE, compiler, and MCU chip in Arduino compatible boards or similar types.
3) Understand Arduino programming with embedded C.
4) Learn how to prototype circuits with a breadboard.

I-Microprocessor and Arduino
Introduction to Microprocessor, Micro controller, source mode and sink mode.
Arduino pin diagram, Types of Arduino boards and its pin description.

II-LED, LCD, Buzzer & Interfacing
Various types of Actuators, LED introduction, Buzzer introduction, Introduction to LCD 16x 2, Types of LCD’s, LCD Pin diagram, Interfacing LED, Buzzer & LCD to Arduino.
LCD: Displaying name on LCD, Setting cursor position in LCD, Display and NO display in LCD, Scrolling Display towards left and right.
LED: LED ON & OFF, TWO LED interfacing (Alternate LED ON & OFF), 1234 and 4321 LED pattern, LCD display with LED status.
Buzzer: Interfacing Buzzer with LED and LCD.

III-Motor driver and Interfacing to Arduino
L293D Pin diagram, Introduction to L293D Motor driver, Interfacing DC motors to Arduino, Designing robot.

IV-Sensors and Programming
Introduction to soil moister sensor, gas sensor, Ultra Sonic sensor, Temperature sensor, LDR sensor, IR sensor and PIR sensor and its pin configurations.
1. Interfacing Soil moister sensor & gas sensor to Arduino
2. Interfacing LED, LCD and gas sensor
3. Interfacing Ultrasonic sensor to Arduino and displaying distance on screen
4. Interfacing Ultrasonic sensor to LED, LCD
5. Interfacing Temperature sensor to Arduino and Interfacing with LED.
6. Interfacing LDR sensor to Arduino and interfacing with LED.
7. Interfacing IR sensor to Arduino and interfacing with LED and buzzer.
8. Interfacing to PIR sensor to Arduino.
**V-Bluetooth & RFID**
Introduction, Principle, Bluetooth pin configuration, Two modes of Bluetooth, Interfacing Bluetooth to Arduino and LCD.

**RFID:** Introduction RFID, Principle, Interfacing to Arduino and LED.

**Expecting outcomes through assessments:**

1: Designing Automatic Street Light
2: Smart stick for blind person.
3: Design Smart water tank

**TEXTBOOKS:**

**REFERENCE BOOKS:**

**CO-PO MAPPING:**

<table>
<thead>
<tr>
<th>Course Outcomes (COs)</th>
<th>Program Outcomes (POs)</th>
<th>Program Specific Outcomes*</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PO 1</td>
<td>PO 2</td>
</tr>
<tr>
<td>CO1</td>
<td>H</td>
<td>M</td>
</tr>
<tr>
<td>CO2</td>
<td>H</td>
<td></td>
</tr>
<tr>
<td>CO3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>