Data Science-LAB

Computational Facility :

65 Computers (HP Compaq 8100 Elite CMT PC)

System Configuration:

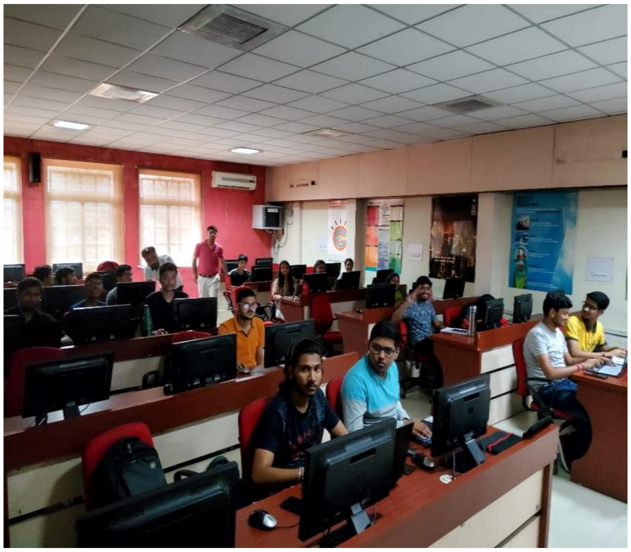
(i) HP LED 18” Monitor

(ii) Intel Core i3 CPU 3.20 GHz.

(iii) RAM 3.00 GB.

(iv) HDD 320 GB.

(v) Wi-Fi LAN Card. (Inbuilt)



Softwares Available:

S.No.	Name of Software	Type of Software
1	Microsoft windows 2007 Professional.	Operating System (Licensed)
2	Mat lab R2013b	Simulator
3	Python-3.7.2	Editor
4	Weka 3 -6-4	Simulator (Data Mining Tool)
5	Mysql 5.5	DBMS
6	Open office 4.1	Utility Software
7	Anaconda3-2018	Editor
8	Code Block	Editor
9	Wire shark	monitoring Network Simulator
10	JDK 6u17	Compiler
11	R-Studio	Data Mining Tool

In Charge

Dr. Sanjiv Sharma
(9407243740)

Physical In Charge:

Shri Imteyaz Husain
(9993765840)

Department Of Information Technology

Data Science-LAB

GENERAL PRACTICES TO BE FOLLOWED IN LABORATORY:

1. Enter in Lab with enclosed Shoes/Footwear (no sandals).
2. In case of fire, please use fire extinguisher.
3. Please shut down your system after use.

DO's

- Be on time, at the start of the lab period, there will be a short introduction to the experiment you will perform that day.
- Maintain silence and clean environment in the lab.
  After completion of lab session, Please arrange chair properly

DON't

- Do not try to run and operate any computer without permission and knowledge of the Lab Personnel
- In case of any mishap - Do not Panic, be calm but quick to report at once to the Lab Personnel.
- Do not eat or drink in the Lab at any time.

Department Of Information Technology

LIST OF EXPERIMENTS:

DESIGN & ANALYSIS OF ALGORITHMS

1. WAP to implement the following using array as data structure and analyze its time Complexity.

   a. Insertion sort b. Selection sort c. Bubble sort d. Quick sort e. Merge sort

   f. Bucket sort g. Heap sort

2. WAP to implement Linear and Binary Search and analyze its time complexity.

3. WAP to implement Strassen's Matrix Multiplication.

4. WAP to implement Matrix Chain Multiplication and analyze its time complexity.

5. WAP to implement Longest Common Subsequence Problem and analyze its time complexity.

6. WAP to implement Optimal Binary Search Tree Problem and analyze its time complexity.

7. WAP to implement 0/1 knapsack using dynamic programming.

8. WAP to implement Dijkstra's Algorithm and analyze its time complexity.

9. WAP to implement Bellman Ford Algorithm and analyze its time complexity.

10. WAP to implement DFS and BFS and analyze their time complexities.

11. WAP to implement Travelling Salesman Problem using backtracking.

12. WAP to implement Topological sort algorithm and analyze their time complexities.

COURSE OUTCOMES:

After completion of this course, the students would be able to:

CO1. relate the principles of algorithm design in solving problems.

CO2. demonstrate basic algorithms and different problem solving strategies.

CO3. build creativeness and confidence to solve non-conventional problems.

CO4. analyze running times of algorithms using asymptotic analysis.

CO5. compare various algorithm design approaches for solving real world problems.

CO6. design and implement optimization algorithms in specific applications.

LIST OF EXPERIMENTS:

THEORY OF COMPUTATION

1. Design a Program for creating machine that accepts three consecutive one.
2. Design a Program for creating machine that accepts the string always ending with 101.
3. Design a program for accepting decimal number divisible by 5.
4. Design a Program for creating machine, which accepts 2 Mod 3.
5. Design a program for creating a machine, which accepts even of 1's and 0's.
6. Design a Program to find 2's complement of a given binary number.
7. Design a Program, which will increment the given binary number by 1.
8. Design a Program to convert NDFA to DFA.
9. Design a program to create PDA to accept anbn where n>0.
10. Design a Program to create PDA machine that accept the well-formed parenthesis.
11. Design a program to create PDA to accept WCWR where w is any string, WR is reverse of that string, and C is a Special symbol.
12. Design a Turing machine that accepts the following language an bn cn where n>0.

COURSE OUTCOMES

After completion of this course, the students would be able to:

CO1. judge various computational models.

CO2. construct abstract models of computing.

CO3. justify the power of abstract models in computing to recognize the languages.

CO4. demonstrate analytical thinking and intuition for problem solving in the related areas.

CO5. discuss the limitations of computation in problem solving.

CO6. follow set of rules for syntax verification.

LIST OF EXPERIMENTS:

PYTHON PROGRAMMING LAB

1. Write a program to demonstrate different number data types in python.
2. Write a program to perform different arithmetic operations on numbers in python.
3. Write a program to create, concatenate and print a string and accessing substring from a given string.
4. Write a python program to create, append and remove lists in python.
5. Write a program to demonstrate working with tuples in python.
6. Write a program to demonstrate working with dictionaries in python.
7. Write a python program to find the factorial of a number using recursion.
8. WAP to swap two integers without using a third variable. The swapping must be done in a different method in a different class.
9. WAP to read content of a file and write into another file.
10. Write a python program to define a module and import a specific function in that module to another program.

COURSE OUTCOMES

After completion of this course, the students would be able to:
CO1. write, test, and debug simple Python programs.
CO2. solve computational problem using python language.
CO3. familiar with basics syntax and features of python programming language.
CO4. use Python lists, tuples, dictionaries for representing compound data.
CO5. design a program utilizing the features of object oriented concept.
CO6. utilize some of the libraries available for solving problems.

LIST OF EXPERIMENTS:

DATABASE MANAGEMENT SYSTEM

1. Implementation of DDL commands of SQL with suitable examples.

   a. Create table b. Alter table c. Drop Table

2. Implementation of DML commands of SQL with examples.

   a. Insert b. Update c. Delete

3. Implementation of different type of function with suitable example

   a. Number function b. Aggregate function c. Character function d. Conversion function e. Data function

4. Implementation of different type of operators in SQL.

   a. Arithmetic operators b. Logical operators c. Set operator f. Comparison Operator g. Special operator

5. Implementation of type of joins.

   a. Inner Join b. Outer Join c. Natural Join etc.

6. Study and implementation of

   a. Group by & having clause b. order by clause c. Indexing

7. Study of Implementation of

   a. Sub queries b. Views

8. Study & implementation of different type of constraints.

9. Study & implementation of database backup & recovery command.

10. Study & implementation of Rollback, commit, savepoint.

11. Creating Database /Table Space

    a. Managing Users: Create User, Delete User

    b. Managing roles: Grant, Revoke.

COURSE OUTCOMES

After completion of this course, the students would be able to:

CO1. construct database schema for a given problem domain.

CO2. apply integrity constraints on a database schema using a state-of-the-art RDBMS.

CO3. apply SQL queries using DDL and DML to design and access database systems.

CO4. make use of operators and functions used in query.

CO5. distinguish Tables and Views for database systems.

CO6. develop a small project for a real world scenario.

LIST OF EXPERIMENTS:

JAVA PROGRAMMING LAB

1. Write a program to accept two numbers (int) as command line arguments and print their Sum.

2. Write a program to find the average and sum of the N numbers Using Command line argument.

3. Write a program to Demonstrate Type Casting.

4. Write a program to find the number of arguments provide at runtime.

5. Write a program to print Fibonacci series without using recursion and using recursion.

6. Write a program to check prime numbers and palindrome numbers.

7. Write a program to sort an array of elements using bubble sort algorithm.

8. Write a program to sort an array of elements using insertion sort algorithm.

9. Write a non-static function in java that prints the sum of two numbers.

10. Create an abstract class Shape which has a field PI=3.14 as final and it has an abstract method Volume. Make two subclasses Cone and Sphere from this class and they print their volume.

11. WAP to handle the Exception using try and multiple catch block.

12. Write a Java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +, -,*, % operations. Add a text field to display the result. Handle any possible exceptions like divide by zero.

13. Develop an Applet that receives an integer in one text field & compute its factorial value & returns it in another text field when the button "Compute" is clicked

14. Write a Java program that implements a multi-thread application that has three threads. First thread generates a random integer every first second and if the value is even, the second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of the cube of the number.

COURSE OUTCOMES

After completion of this course, the students would be able to:

CO1. tell the available features in Java programming language.

CO2. illustrate Java programming concepts for solving problems.

CO3. make use of the Java programming methods for connecting the various databases.

CO4. test for bugs in a software application written in the Java programming language.

CO5. determine different ways for handling exceptions, memory management, file handling, i/o management and internet based application development.

CO6. build a project for application development using Java programming language.