VISION

To impart quality education through state-of-the-art technologies to achieve academic excellence for transforming students into innovators.

MISSION

To impart quality education through state-of-the-art technologies to achieve academic excellence for transforming students into innovators.

Creating a teaching-learning environment to produce industry ready and self-confident graduates. Motivate students to engage in creative projects throughout graduation.

To produce competitive graduates having creative skills and ethical values to succeed in their fields as well as the foundation for life-long learning.

PROGRAM EDUCATIONAL OBJECTIVES (PEO)

1. To provide students with strong basic and advanced programming concepts so that they can build solutions or systems for complex problems.
2. The program provides the fundamental and perspective to attain life-long learning in the thrust areas of Computer Programming.
3. To produce graduates who have ability to pursue research or have a successful career in academia or industries or as entrepreneurs.
4. The aim is inculcating technical knowledge of the programme and imbibe ethics with moral behavior in the graduates.

PROGRAM SPECIFIC OBJECTIVES (PSO)

1. To acquire basic knowledge in hardware/software, algorithms, System Software, Computer graphics, Web design, Networking, and advanced computing for solving real-life and Research problems with the perspective of lifelong learning.
2. An ability to demonstrate Knowledge of data management systems like data acquisition and big data, Intelligent systems like AI, Data Science and Machine Learning, The techniques of data analytics like pattern recognition and knowledge discovery.
3. To develop skills which help to expand professional careers.

COURSE OUTCOMES (CO)

1. To study the asymptotic performance of algorithms.
2. Apply various complexity measures and find out performance of the algorithm through divide and conquer like searching and sorting.
3. To generate optimal solutions by applying various Greedy and Dynamic algorithms.
4. To apply fundamental algorithms to model engineering problem solving using various graph methods or using suitable data structures.

~TABLE OF CONTENT~

Name:	Enrolment No:	Semester:

09	Rewrite all .jsp pages of previous labs using EL and JSTL.				
10	Write Hibernate application to store student records and retrieve the student record including name, enrolment no, sem, div, dept, sgpa, cgpa, etc.				
11	Write an application to keep, update and retrieve record of student. The record includes student name, enrollment no, semester, dept, percentage, etc. Use SPRING MVC architecture.				
12	Mini Project to create web application by using Advance Java Programming course.				

MADHUBEN & BHANUBHAI PATEL INSTITUTE OF TECHNOLOGY (A CONSTITUENT COLLEGE OF CVM UNIVERSITY)

DEPARTMENT OF COMPUTER ENGINEERING

A. Y. 2022 - 23, ODD TERM

SUBJECT CODE: 102010505

SUBJECT NAME: Advanced Java Programming

~TABLE OF CONTENT~

Name:	Enrolment No:	Semester:

Practical -1

Aim: Write a JDBC desktop program to perform following CRUD and Search operation. Create appropriate table in database to store objects of Student class.

1. Add Student. (Create)
2. View Students. (Display all students)
3. Edit Student. (Update)
4. Delete Student. (Delete)
5. Search Student (Find student based on enrolment No)

Code:

```
import  java.sql.*;
import java.util.Scanner;

public class prac1
{
  // JDBC driver name and database URL
  static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
  static final String DB_URL = "jdbc:mysql://localhost:3306/college";


  //  Database credentials
  static final String USER = "root";
  static final String PASS = "";


  public static void main(String[] args)
  {
```

```
Scanner sc = new Scanner(System.in);
Connection conn = null;
Statement stmt = null;

try{
   // Register JDBC driver
   Class.forName(JDBC_DRIVER);

   // Open a connection
   System.out.println("Connecting to database...");
   conn = DriverManager.getConnection(DB_URL,USER,PASS);

   int flag=0;


//  Creating Table
// String sql1 = "CREATE TABLE Student " +"(ID INTEGER(3) not NULL, " +" Name
VARCHAR(30), " + " PRIMARY KEY ( ID ))";
// stmt.executeUpdate(sql1);
// System.out.println("********************");
// System.out.println("Created table in given database...");
// System.out.println("********************");


   //choice
   do
   {
      // Execute a query
      stmt = conn.createStatement();
      String sql;
      sql = "SELECT * FROM student";
      ResultSet rs = stmt.executeQuery(sql);
```

```
//Query
String sql1;
//Object for query
PreparedStatement stmt1;
//Enrolment and name
String n,e;
int count=0;


System.out.println("Enter which operation you want to perform:-\n1. Add Student\n2.
View Students\n3. Edit Student\n4. Delete Student\n5. Search Student");
int c=sc.nextInt();
if(c<6&&c>0)
{
   switch(c)
   {
      case 1:
      int flag2=0;
         do
         {
            try {
               System.out.println("Enter Enrolment number of student:");
               e=sc.next();
               System.out.println("Enter Name of student:");
               n=sc.next();
               sql1 = "INSERT INTO student (Er_no, Name ) VALUES (?, ?)";

               stmt1 = conn.prepareStatement(sql1);
               stmt1.setString(1, e);
               stmt1.setString(2, n);

               int rowsInserted = stmt1.executeUpdate();
```

```
            if (rowsInserted > 0) {
                System.out.println("A new user "+n+" was inserted successfully!");
            }
            break;
        }
        catch (Exception e1)
        {
            System.out.println("Enrolment number already exists!!!");
            flag2=1;
        }
    }while(flag2==1);
    break;
case 2:
    while(rs.next())
    {
        //Retrieve by column name
        int id  = rs.getInt("Er_no");
        String name = rs.getString("Name");

        //Display values
        System.out.print("Enrolment No: " + id);
        System.out.println(", Name: " + name);
    }
    break;
case 3:
    System.out.println("Enter Enrolment number of student:");
    e=sc.next();
    System.out.println("Enter Name of student:");
    n=sc.next();
    sql1 = "UPDATE student SET Er_no=?,Name=? WHERE Er_no=?";

    stmt1 = conn.prepareStatement(sql1);
    stmt1.setString(1, e);
```

```
    stmt1.setString(2, n);
    stmt1.setString(3, e);

    int rowsUpdated = stmt1.executeUpdate();
    if (rowsUpdated > 0) {
       System.out.println("An existing user "+n+" was updated successfully!");
    }
       break;
case 4:
    System.out.println("Enter Enrolment number of student:");
    e=sc.next();

    sql1 = "DELETE FROM student WHERE Er_no=?";

    stmt1 = conn.prepareStatement(sql1);
    stmt1.setString(1, e);

    int rowsDeleted = stmt1.executeUpdate();
    if (rowsDeleted > 0) {
       System.out.println("A user with Enrolment no "+e+" is deleted successfully!");
    }
    break;
case 5:
    int flag3=0;
    do
    {
       System.out.println("How you want to search:-\n1. By Enrolment Number\n2.
          Name");
       int s=sc.nextInt();
       if(s==1)
       {
          System.out.println("Enter Enrolment number of student:");
          e=sc.next();
```

```
    sql1 = "SELECT * FROM student WHERE Er_no="+e;
    stmt1 = conn.prepareStatement(sql1);

    rs = stmt1.executeQuery(sql1);
    while(rs.next()){
       count++;
       //Retrieve by column name
       int id  = rs.getInt("Er_no");
       String name = rs.getString("Name");

       //Display values
       System.out.println("ID: " + id);
       System.out.println("Name: " + name);
    }
    if (count ==1) {
       System.out.println("User found!!");
    }
    else if (count >1) {
       System.out.println(count +" Users found!!");
    }
    else
    {
       System.out.println("User not found!!");
    }
    break;
}
else if(s==2)
{
    System.out.println("Enter Name of student:");
    n=sc.next();
    sql1 = "SELECT Name,Er_no FROM student WHERE Name like
    "+"'%"+n+"%'";
    stmt1 = conn.prepareStatement(sql1);
```

```
            rs = stmt1.executeQuery(sql1);
            while(rs.next()){
               count++;
               //Retrieve by column name
               int id  = rs.getInt("Er_no");
               String name = rs.getString("Name");

               //Display values
               System.out.println("ID: " + id);
               System.out.println("Name: " + name);
            }
            if (count ==1) {
               System.out.println("User found!!");
            }
            else if (count >1) {
               System.out.println(count+" Users found!!");
            }
            else
            {
               System.out.println("User not found!!");
            }
            break;
         }
         else
         {
            System.out.println("Invalid choice!!");
            flag3=1;
         }
      }while(flag3==1);
      break;
   default:
   System.out.println("Inavalid choice");
```

```
                flag=1;
            }
        }
        else
        {
            System.out.println("Invalid choice!!");
            flag=1;
        }
        //Repeat Process
        System.out.println("Enter 1 to do operation again:-");
        flag=sc.nextInt();
        rs.close();
    }while(flag==1);

    // Clean-up environment
    stmt.close();
    conn.close();
}
catch(Exception e)
{
    System.out.println(e);
}
finally
{
    //finally block used to close resources
    try
    {
        if(stmt!=null)
        stmt.close();
    }
    catch(SQLException se2)
    {
        // nothing we can do
```

```
        }
        try
        {
           if(conn!=null)
           conn.close();
        }
        catch(Exception e)
        {
           System.out.println(e);
           //end finally try
        }

      }//end try

      System.out.println("Goodbye!");
      sc.close();
   }//end main

}//end FirstExample
```

Output:

```
Enter which operation you want to perform:-
1. Add Student
2. View Students
3. Edit Student
4. Delete Student
5. Search Student
1
Enter Enrolment number of student:
10
Enter Name of student:
Aeshwary
A new user Aeshwary was inserted successfully!
Enter 1 to do operation again:-
1
Enter which operation you want to perform:-
1. Add Student
2. View Students
3. Edit Student
4. Delete Student
5. Search Student
1
Enter Enrolment number of student:
39
Enter Name of student:
Dhruval
A new user Dhruval was inserted successfully!
Enter 1 to do operation again:-
1
Enter which operation you want to perform:-
1. Add Student
2. View Students
3. Edit Student
4. Delete Student
5. Search Student
1
Enter Enrolment number of student:
67
Enter Name of student:
Huned
A new user Huned was inserted successfully!
```

```
Enter 1 to do operation again:-
1
Enter which operation you want to perform:-
1. Add Student
2. View Students
3. Edit Student
4. Delete Student
5. Search Student
1
Enter Enrolment number of student:
57
Enter Name of student:
Harshad
A new user Harshad was inserted successfully!
Enter 1 to do operation again:-
1
Enter which operation you want to perform:-
1. Add Student
2. View Students
3. Edit Student
4. Delete Student
5. Search Student
1
Enter Enrolment number of student:
66
Enter Name of student:
Hitendra
A new user Hitendra was inserted successfully!
Enter 1 to do operation again:-
1
Enter which operation you want to perform:-
1. Add Student
2. View Students
3. Edit Student
4. Delete Student
5. Search Student
2
Enrolment No: 10, Name: Aeshwary
Enrolment No: 39, Name: Dhruval
Enrolment No: 57, Name: Harshad
Enrolment No: 66, Name: Hitendra
Enrolment No: 67, Name: Huned
Enter 1 to do operation again:-
1
Enter which operation you want to perform:-
1. Add Student
2. View Students
3. Edit Student
4. Delete Student
5. Search Student
3
Enter Enrolment number of student:
```

```
67
Enter Name of student:
Hunaid
An existing user Hunaid was updated successfully!
Enter 1 to do operation again:-
1
Enter which operation you want to perform:-
1. Add Student
2. View Students
3. Edit Student
4. Delete Student
5. Search Student
2
Enrolment No: 10, Name: Aeshwary
Enrolment No: 39, Name: Dhruval
Enrolment No: 57, Name: Harshad
Enrolment No: 66, Name: Hitendra
Enrolment No: 67, Name: Hunaid
Enter 1 to do operation again:-
1
Enter which operation you want to perform:-
1. Add Student
2. View Students
3. Edit Student
4. Delete Student
5. Search Student
4
Enter Enrolment number of student:
66
A user with Enrolment no 66 is deleted successfully!
Enter 1 to do operation again:-
1
Enter which operation you want to perform:-
1. Add Student
2. View Students
3. Edit Student
4. Delete Student
5. Search Student
2
Enrolment No: 10, Name: Aeshwary
Enrolment No: 39, Name: Dhruval
Enrolment No: 57, Name: Harshad
Enrolment No: 67, Name: Hunaid
Enter 1 to do operation again:-
1
Enter which operation you want to perform:-
1. Add Student
2. View Students
3. Edit Student
4. Delete Student
5. Search Student
5
How you want to search:-
1. By Enrolment Number
2. Name
1
```

```
1
Enter Enrolment number of student:
10
ID: 10
Name: Aeshwary
User found!!
Enter 1 to do operation again:-
0
Goodbye!
```