```
https://github.com/MuqaddasRani/JAVA.git //Muqaddas Rani
//Roll no#18321519-078
//Section#B
//OOP final term
//Code of “Student give feedback to teacher”
public class student {
    String name[];
    String id;
    String courses;
    String DOB;
    String gender;
int rollno;
int marks;
public String[] getName() {
return name;
}
public void setName(String[] name) {
this.name = name;
}
public String getId() {
return id;
}
public void setId(String id) {
this.id = id;
}
public String getCourses() {
return courses;
```

```
}
public void setCourses(String courses) {
this.courses = courses;
}
public String getDOB() {
return DOB;
}
public void setDOB(String dOB) {
DOB = dOB;
}
public String getGender() {
return gender;
}
public void setGender(String gender) {
this.gender = gender;
}
public int getRollno() {
return rollno;
}
public void setRollno(int rollno) {
this.rollno = rollno;
}
public int getMarks() {
return marks;
}
public void setMarks(int marks) {
this.marks = marks;
```

```
public class teacher {
String name[];
String gender;
String courses;
int registeration_id;
String DOB;
int salary;
public String[] getName() {
return name;
}
public void setName(String[] name) {
this.name = name;
}
public String getGender() {
return gender;
}
public void setGender(String gender) {
this.gender = gender;
}
public String getCourses() {
return courses;
}
public void setCourses(String courses) {
this.courses = courses;
}
public int getRegisteration_id() {
```

```
return registeration_id;
}
public void setRegisteration_id(int registeration_id) {
this.registeration_id = registeration_id;
}
public String getDOB() {
return DOB;
}
public void setDOB(String dOB) {
DOB = dOB;
}
public int getSalary() {
return salary;
}
public void setSalary(int salary) {
this.salary = salary;
  public String toString() {

    String str = "null";
    str = this.TEACHER_ID
        + "," + this.TEACHER_NAME
        return str;

}

import java.util.Scanner;
 public class GetStudentDetails{
  public static void main(String args[])     {
String name;
   int roll, math, phy, eng;
 Scanner SC=new Scanner(System.in);                    System.out.print("Enter Name: ");
   name=SC.nextLine();
  System.out.print("Enter Roll Number: ");
roll=SC.nextInt();
System.out.print("Enter marks in Maths, Physics and English: ");         math=SC.nextInt();
phy=SC.nextInt();
  eng=SC.nextInt();
int total=math+eng+phy;
   float perc=(float)total/300*100;
 "Roll Number:".println("Roll Number:" + roll +"\tName: "+name);
 System.out.println("Marks (Maths, Physics,, English): " +math+","+phy+","+eng);
   System.out.println("Total: "+total +"\tPercentage: "+perc);
          }
}

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

public class studentData {
@SuppressWarnings("resource")
```

```
public List<Student> findAll() {
List<Student> students = new ArrayList<Student>();
String line;
try {
BufferedReader bufferreader = new BufferedReader(new FileReader(Student.csvFile));
// reading data from file and storing it in a List to be displayed
while ((line = bufferreader.readLine()) != null) {
Student student = new Student();
String[] studentRow = line.split(",");
student.setSTUDENT_ID(Integer.parseInt(studentRow[0]));
student.setSTUDENT_ROLLNO(studentRow[1]);
student.setSTUDY_gender(studentRow[2]);
Student.setSTUDY DOB(studentRow[3]);
student.setSTUDY_Marks(studentRow[4]);
students.add(student);
}
} catch (IOException e) {
e.printStackTrace();
}
return students;
}

@SuppressWarnings("resource")
public static Student findOne(int Student_ID) {
String line;
try {
BufferedReader bufferreader = new BufferedReader(new FileReader(Student.csvFile));
```

```
while ((line = bufferreader.readLine()) != null) {
Student student = new Student();
String[] studentRow = line.split(",");
if (Integer.parseInt(studentRow[0]) == Student_ID) {
student.setSTUDENT_ID(Integer.parseInt(studentRow[0]));
student.setSTUDENT_ROLLNO(studentRow[1]);
student.setSTUDY_Gender(studentRow[2]);
student.setSTUDY_DOB(studentRow[3]);
student.setSTUDY_Marks(studentRow[4]);
return student;
}
}
} catch (IOException e) {
System.out.println("the record does not exist!!!");
} finally {
System.out.println("enter again");
studentOptions.StudentsDataOptions();
}
return null;
}

@SuppressWarnings("resource")
public List<Student> search(String search)  {
List<Student> students = new ArrayList<Student>();
String line;
try {
BufferedReader bufferreader = new BufferedReader(new FileReader(Student.csvFile));
```

```
while ((line = bufferreader.readLine()) != null) {
Student student = new Student();
String[] studentRow = line.split(",");
if (studentRow[1].contains(search) == true) {
student.setSTUDENT_ID(Integer.parseInt(studentRow[0]));
student.setSTUDENT_ROLLNO(studentRow[1]);
student.setSTUDY_Gender(studentRow[2]);
student.setSTUDY_DOB(studentRow[3]);
student.setSTUDY_Marks(studentRow[4]);
students.add(student);
return students;
}
}
} catch (IOException e) {
e.printStackTrace();
}
return students;
}

@SuppressWarnings("static-access")
public Student Save(Student Student) {
FileWriter filewriter;

List<Student> students = findAll();

try {
filewriter = new FileWriter(Student.csvFile);
```

```
for (int i=0; i<students.size(); i++) {
filewriter.append(students.get(i).toString());
filewriter.append("\n");
}
if (students.size()>0)
Student.setSTUDENT_ID(students.get(students.size()-1).getSTUDENT_ID()+1);
else
Student.setSTUDENT_ID(1);
filewriter.append(Student.toString());
filewriter.append("\n");
filewriter.flush();
filewriter.close();
} catch (IOException e) {
e.printStackTrace();
}
return Student;
}
}
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

import edu.uog.teacher.Teacher;;
```

```
public class teacherData {
@SuppressWarnings("resource")
public static List<Teacher> findAll()  {
List<Teacher> teachers = new ArrayList<Teacher>();
String line;
try {
BufferedReader bufferreader = new BufferedReader(new FileReader(Teacher.csvFile));
// reading data from file and storing it in a List to be displayed
while ((line = bufferreader.readLine()) != null) {
Teacher teacher = new Teacher();
String[] teacherRow = line.split(",");
teacher.setTEACHER_ID(Integer.parseInt(teacherRow[0]));
teacher.setTEACHER_NAME(teacherRow[1]);
teacher.setTEACHER_Courses(teacherRow[2]);
teacher.setTEACHER_Gender(teacherRow[3]);
teacher.setTEACHER_DOB(teacherRow[4]);
teacher.setTEACHER_Salary(teacherRow[5]);
teachers.add(teacher);
}
} catch (IOException e) {
e.printStackTrace();
}
return teachers;
}

@SuppressWarnings("resource")
```

```
public static Teacher findOne(int Teacher_ID)  {
String line;
try {
BufferedReader bufferreader = new BufferedReader(new FileReader(Teacher.csvFile));
while ((line = bufferreader.readLine()) != null) {
Teacher teacher = new Teacher();
String[] teacherRow = line.split(",");
if (Integer.parseInt(teacherRow[0]) == Teacher_ID) {
teacher.setTEACHER_ID(Integer.parseInt(teacherRow[0]));
teacher.setTEACHER_NAME(teacherRow[1]);
teacher.setTEACHER_Courses(teacherRow[2]);
teacher.setTEACHER_Gender(teacherRow[3]);
teacher.setTEACHER_DOB(teacherRow[4]);
teacher.setTEACHER_Salary(teacherRow[5]);
return teacher;
}
}
} catch (IOException e) {
e.printStackTrace();
}
return null;
}
mport java.io.BufferedReader;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
public class teacherData {
    @SuppressWarnings("resource")
    public static List<Teacher> findAll()  {
        List<Teacher> teachers = new ArrayList<Teacher>();
        String line;

        try {
            BufferedReader bufferreader = new BufferedReader(new
FileReader(Teacher.csvFile));
            // reading data from file and storing it in a List to be displayed
            while ((line = bufferreader.readLine()) != null) {
                Teacher teacher = new Teacher();

                String[] teacherRow = line.split(",");

                                teacher.setTEACHER_ID(Integer.parseInt(teacherRow[0]));
teacher.setTEACHER_NAME(teacherRow[1]);
teacher.setTEACHER_Courses(teacherRow[2]);
teacher.setTEACHER_Gender(teacherRow[3]);
teacher.setTEACHER_DOB(teacherRow[4]);
teacher.setTEACHER_Salary(teacherRow[5]);


                teachers.add(teacher);
            }
        } catch (IOException e) {
            e.printStackTrace();
        }

        return teachers;
    }

    @SuppressWarnings("resource")
    public static Teacher findOne(int Teacher_ID)  {
        String line;

        try {
            BufferedReader bufferreader = new BufferedReader(new
FileReader(Teacher.csvFile));

            while ((line = bufferreader.readLine()) != null) {
                Teacher teacher = new Teacher();
                String[] teacherRow = line.split(",");

                if (Integer.parseInt(teacherRow[0]) == Teacher_ID) {

                    teacher.setTEACHER_ID(Integer.parseInt(teacherRow[0]));
teacher.setTEACHER_NAME(teacherRow[1]);
teacher.setTEACHER_Courses(teacherRow[2]);
teacher.setTEACHER_Gender(teacherRow[3]);
teacher.setTEACHER_DOB(teacherRow[4]);
teacher.setTEACHER_Salary(teacherRow[5]);
```

```
                    return teacher;
                }


            }
        } catch (IOException e) {
            e.printStackTrace();
        }

        return null;
    }

    @SuppressWarnings("resource")
    public static List<Teacher> search(String search)  {
        List<Teacher> teachers = new ArrayList<Teacher>();
        String line;

        try {
            BufferedReader bufferreader = new BufferedReader(new
FileReader(Teacher.csvFile));

            while ((line = bufferreader.readLine()) != null) {
                Teacher teacher = new Teacher();

                String[] teacherRow = line.split(",");
                if (teacherRow[1].contains(search) == true) {

                    teacher.setTEACHER_ID(Integer.parseInt(teacherRow[0]));
teacher.setTEACHER_Courses(teacherRow[2]);
teacher.setTEACHER_Gender(teacherRow[3]);
teacher.setTEACHER_DOB(teacherRow[4]);
teacher.setTEACHER_Salary(teacherRow[5]);

                    teachers.add(teacher);
                }

            }

        } catch (IOException e) {
            e.printStackTrace();
        }

        return teachers;
    }

    @SuppressWarnings("static-access")
    public static Teacher Save(Teacher Teacher) {
        FileWriter filewriter;

        List<Teacher> teachers = findAll();

        try {
            filewriter = new FileWriter(Teacher.csvFile);

            for (int i=0; i<teachers.size(); i++) {
                filewriter.append(teachers.get(i).toString());
                filewriter.append("\n");
            }
            if (teachers.size()>0)
                Teacher.setTEACHER_ID(teachers.get(teachers.size()-1).getTEACHER_ID()+1);
            else
                Teacher.setTEACHER_ID(1);

            filewriter.append(Teacher.toString());
            filewriter.append("\n");
            filewriter.flush();
            filewriter.close();
        } catch (IOException e) {
            e.printStackTrace();
        }


        return Teacher;
    }
}


@SuppressWarnings("resource")
public static List<Teacher> search(String search)  {
```

```
List<Teacher> teachers = new ArrayList<Teacher>();
String line;
try {
BufferedReader bufferreader = new BufferedReader(new FileReader(Teacher.csvFile));
while ((line = bufferreader.readLine()) != null) {
Teacher teacher = new Teacher();
String[] teacherRow = line.split(",");
if (teacherRow[1].contains(search) == true) {
teacher.setTEACHER_ID(Integer.parseInt(teacherRow[0]));
teacher.setTEACHER_NAME(teacherRow[1]);
teacher.setTEACHER_Courses(teacherRow[2]);
teacher.setTEACHER_Gender(teacherRow[3]);
teacher.setTEACHER_DOB(teacherRow[4]);
teacher.setTEACHER_Salary(teacherRow[5]);
teachers.add(teacher);
}
}
} catch (IOException e) {
e.printStackTrace();
}
return teachers;
}

@SuppressWarnings("static-access")
public static Teacher Save(Teacher Teacher) {
FileWriter filewriter;
```

```
List<Teacher> teachers = findAll();

try {
filewriter = new FileWriter(Teacher.csvFile);

for (int i=0; i<teachers.size(); i++) {
filewriter.append(teachers.get(i).toString());
filewriter.append("\n");
}
if (teachers.size()>0)
Teacher.setTEACHER_ID(teachers.get(teachers.size()-1).getTEACHER_ID()+1);
else
Teacher.setTEACHER_ID(1);
filewriter.append(Teacher.toString());
filewriter.append("\n");
filewriter.flush();
filewriter.close();
} catch (IOException e) {
e.printStackTrace();
}
return Teacher;
}
}
import java.util.List;
import java.util.Scanner;
public class teacherOptions {
    static Scanner scan = new Scanner(System.in);
```

```
public static void TeachersDataOptions() {
    char op;
    System.out.println("\n\n\n\n\n");
    System.out.println("-------------------------------------------- ");
    System.out.println("       ------Accessed Teachers Data------   ");
    System.out.println("================================================");
    System.out.println("       ------Enter Your Choice------        ");
    System.out.println("================================================");
    System.out.println("1. For Viewing All The Records .");
    System.out.println("2. For Finding a Record.");
    System.out.println("3. For Searching A Record.");
    System.out.println("4. For Adding a New Record.");
    System.out.println("5. For Deleting a Record.");
    System.out.println("6. For Returning to Main Menu.");
    System.out.println("================================================");
    do {
        System.out.println("Choice: ");
        scan.hasNext();
        op = scan.next().charAt(0);
        switch(op) {
        case '1':
            System.out.println("----------------------------------");
            findallOption();
            break;
        case '2':
            System.out.println("----------------------------------");
    findoneOption();
    break;
case '3':
    System.out.println("---------------------------------");
    searchOption();
    break;
case '4':
    System.out.println("---------------------------------");
    saveOption();
    break;
case '5':
    //deleting();
    break;
case '6':
    Option option = new Option();
    System.out.println("....");
    try {
        option.optionSelect();
    } catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    break;
default:
    System.out.println("Invalid Input!!!!\nEnter Again!!!");
    break;
}
```

```
    }while(!(op=='1'||op=='2'||op=='3'||op=='4'||op=='5'||op=='6'));
}
public static void findallOption() {
    System.out.println("--------------------------------");
    System.out.println("FindAll");
    List<Teacher> teachers = teacherData.findAll();
    for (int i=0; i<teachers.size(); i++) {
        System.out.println(teachers.get(i).toString());
    }
    System.out.println("\n\n\n\n\n");
    try {
        Thread.sleep(1000);
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
    TeachersDataOptions();
}
public static void findoneOption() {
    System.out.println("--------------------------------");
    System.out.println("Enter Id number of record to be Found");
    Scanner scan = new Scanner(System.in);
    int op = scan.nextInt();
    Teacher teacher = teacherData.findOne(op);
    System.out.println(teacher.toString());
    System.out.println("\n\n\n\n\n");
    try {
        Thread.sleep(1000);
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
    TeachersDataOptions();
    scan.close();
}
public static void searchOption() {
    Scanner scan = new Scanner(System.in);
    System.out.println("--------------------------------");
    System.out.println("Enter Teacher Name to search its data");
    String name = scan.nextLine();
    List<Teacher> teachers = teacherData.search(name.toUpperCase());
    for (int i=0; i<teachers.size(); i++) {
        System.out.println(teachers.get(i).toString());
    }
    System.out.println("\n\n\n\n\n");
    try {
        Thread.sleep(1000);
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
    TeachersDataOptions();
    scan.close();
}
public static void saveOption() {
    System.out.println("--------------------------------");
```

```
        Scanner scan = new Scanner(System.in);
        Teacher teacher = new Teacher();

        teacher.setTEACHER_ID(1);
        teacher.setTEACHER_NAME("Nouman Riaz".toUpperCase());

        teacher = teacherData.Save(teacher);
        System.out.println(teacher.toString());

        System.out.println("\n\n\n\n\n");
        try {
            Thread.sleep(1000);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
        TeachersDataOptions();
        scan.close();
    }

}
import java.util.List;
import java.util.Scanner;

public class studentOptions {

    static studentData studentData = new studentData();
```

```
public static void  StudentsDataOptions() {
    Scanner scan = new Scanner(System.in);
    char op;
    System.out.println("\n\n\n\n\n");
    System.out.println("------------------------------------------------------ ");
    System.out.println("      ------Accessed Students Data------   ");
    System.out.println("=========================================================");
    System.out.println("      ------Enter Your Choice------        ");
    System.out.println("=========================================================");
    System.out.println("1. For Viewing All The Records .");
    System.out.println("2. For Finding a Record.");
    System.out.println("3. For Searching A Record.");
    System.out.println("4. For Adding a New Record.");
    System.out.println("5. For Deleting a Record.");
    System.out.println("6. For Returning to Main Menu.");
    System.out.println("=========================================================");
    do {
        System.out.println("Choice: ");
        scan.hasNext();
        op = scan.next().charAt(0);
        switch(op) {
        case '1':
            System.out.println("--------------------------------------------");
            findallOption();
            break;
        case '2':
            System.out.println("--------------------------------------------");
    findoneOption();
    break;
case '3':
    System.out.println("---------------------------");
    searchOption();
    break;
case '4':
    System.out.println("---------------------------");
    saveOption();
    break;
case '5':
    //deleting();
    break;
case '6':
    Option option = new Option();
    System.out.println("....");
    try {
        option.optionSelect();
    } catch (Exception e) {
        e.printStackTrace();
    }
    break;
default:
    System.out.println("Invalid Input!!!!\nEnter Again!!!");
    break;
}
```

```
    }while(!(op=='1'||op=='2'||op=='3'||op=='4'||op=='5'||op=='6'));
    scan.close();
}
public static void  findallOption() {
    System.out.println("--------------------------------");
    System.out.println("FindAll");
    List<Student> Students = studentData.findAll();
    for (int i=0; i<Students.size(); i++) {
        System.out.println(Students.get(i).toString());
    }
    System.out.println("\n\n\n\n\n");
    try {
        Thread.sleep(1000);
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
    StudentsDataOptions();
}
public static void  findoneOption() {
    System.out.println("--------------------------------");
    System.out.println("Enter Id number of record to be Found");
    Scanner scan = new Scanner(System.in);
    int op = scan.nextInt();
    Student Student = studentData.findOne(op);
    System.out.println(Student.toString());
    System.out.println("\n\n\n\n\n");
    try {
        Thread.sleep(1000);
    } catch (InterruptedException e) {
        e.printStackTrace();
    }

    StudentsDataOptions();
    scan.close();
}
public static void  searchOption() {
    Scanner scan = new Scanner(System.in);
    System.out.println("--------------------------------");
    System.out.println("Enter Student Roll number to search its data");
    String stdCode = scan.nextLine();
    List<Student> Students = studentData.search(stdCode);
    for (int i=0; i<Students.size(); i++) {
        System.out.println(Students.get(i).toString());
    }
    System.out.println("\n\n\n\n\n");
    try {
        Thread.sleep(1000);
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
    StudentsDataOptions();
    scan.close();
}
public static void  saveOption() {
        System.out.println("--------------------------------");
        Scanner scan = new Scanner(System.in);
        Student student = new Student();

        student.setSTUDENT_ID(1);
        System.out.println("Enter Student Roll Number: ");
            String studentRollNo = scan.nextLine();
            student.setSTUDENT_ROLLNO(studentRollNo);
        System.out.println("Enter Gender ");
            String study = scan.nextLine();
            student.setSTUDY_DO(Data_of_Birth);
        System.out.println("Enter Marks");
            String study = scan.nextLine();

        student = studentData.Save(student);
        System.out.println(student.toString());

        System.out.println("\n\n\n\n\n");
        try {
            Thread.sleep(1000);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
        StudentsDataOptions();
        scan.close();
    }
}
```

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

public class studentData {
    @SuppressWarnings("resource")
    public List<Student> findAll()  {
        List<Student> students = new ArrayList<Student>();
        String line;

        try {
            BufferedReader bufferreader = new BufferedReader(new
FileReader(Student.csvFile));
            // reading data from file and storing it in a List to be displayed
            while ((line = bufferreader.readLine()) != null) {
                Student student = new Student();

                String[] studentRow = line.split(",");

                student.setSTUDENT_ID(Integer.parseInt(studentRow[0]));
                student.setSTUDENT_ROLLNO(studentRow[1]);
                Student.set_Gender(studentRow[2]);
                student.set_DOB(studentRow[3]);
               student.set_Marks(studentRow[4]);
```

```
                students.add(student);
            }

        } catch (IOException e) {
            e.printStackTrace();
        }

        return students;
    }

    @SuppressWarnings("resource")
    public static Student findOne(int Student_ID)  {
        String line;

        try {
            BufferedReader bufferreader = new BufferedReader(new
FileReader(Student.csvFile));

            while ((line = bufferreader.readLine()) != null) {
                Student student = new Student();
                String[] studentRow = line.split(",");

                if (Integer.parseInt(studentRow[0]) == Student_ID) {
                    student.setSTUDENT_ID(Integer.parseInt(studentRow[0]));
                student.setSTUDENT_ROLLNO(studentRow[1]);
                Student.set_Gender(studentRow[2]);
                student.set_DOB(studentRow[3]);
               student.set_Marks(studentRow[4]);



                    return student;
                }



            }
        } catch (IOException e) {
            System.out.println("the record does not exist!!!");

        } finally {
            System.out.println("enter again");
            studentOptions.StudentsDataOptions();
        }

        return null;
    }

    @SuppressWarnings("resource")
    public List<Student> search(String search)  {
        List<Student> students = new ArrayList<Student>();
        String line;

        try {
          BufferedReader bufferreader = new BufferedReader(new
    FileReader(Student.csvFile));


          while ((line = bufferreader.readLine()) != null) {
            Student student = new Student();


            String[] studentRow = line.split(",");


            if (studentRow[1].contains(search) == true) {
              student.setSTUDENT_ID(Integer.parseInt(studentRow[0]));
             student.setSTUDENT_ROLLNO(studentRow[1]);
            Student.set_Gender(studentRow[2]);
            student.set_DOB(studentRow[3]);
           student.set_Marks(studentRow[4]);


              students.add(student);
              return students;
            }


          }


        } catch (IOException e) {
          e.printStackTrace();
        }


        return students;
      }
```

```
@SuppressWarnings("static-access")
public Student Save(Student Student) {
    FileWriter filewriter;

    List<Student> students = findAll();

    try {
        filewriter = new FileWriter(Student.csvFile);

        for (int i=0; i<students.size(); i++) {
            filewriter.append(students.get(i).toString());
            filewriter.append("\n");
        }
        if (students.size()>0)
            Student.setSTUDENT_ID(students.get(students.size()-1).getSTUDENT_ID()+1);
        else
            Student.setSTUDENT_ID(1);
        filewriter.append(Student.toString());
        filewriter.append("\n");
        filewriter.flush();
        filewriter.close();
    } catch (IOException e) {
        e.printStackTrace();
    }
```

```
        return Student;
    }
}
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

public class studentData {
    @SuppressWarnings("resource")
    public List<Student> findAll()  {
        List<Student> students = new ArrayList<Student>();
        String line;

        try {
            BufferedReader bufferreader = new BufferedReader(new FileReader(Student.csvFile));
            // reading data from file and storing it in a List to be displayed
            while ((line = bufferreader.readLine()) != null) {
                Student student = new Student();

                String[] studentRow = line.split(",");

                student.setSTUDENT_ID(Integer.parseInt(studentRow[0]));
                student.setSTUDENT_ROLLNO(student
Student.set_Gender(studentRow[2]);
                student.set_DOB(studentRow[3]);
               student.set_Marks(studentRow[4]);



                students.add(student);
            }

        } catch (IOException e) {
            e.printStackTrace();
        }

        return students;
    }

    @SuppressWarnings("resource")
    public static Student findOne(int Student_ID)  {
        String line;

        try {
            BufferedReader bufferreader = new BufferedReader(new
FileReader(Student.csvFile));

            while ((line = bufferreader.readLine()) != null) {
                Student student = new Student();
                String[] studentRow = line.split(",");
        if (Integer.parseInt(studentRow[0]) == Student_ID) {
            student.setSTUDENT_ID(Integer.parseInt(studentRow[0]));
                          student.setSTUDENT_ROLLNO(studentRow[1]);
        Student.set_Gender(studentRow[2]);
        student.set_DOB(studentRow[3]);
        student.set_Marks(studentRow[4]);



            return student;
        }



    }

public class studentOptions {
static studentData studentData = new studentData();

public static void  StudentsDataOptions() {
Scanner scan = new Scanner(System.in);
char op;
System.out.println("\n\n\n\n\n");
System.out.println("-------------------------------------------- ");
System.out.println("      ------Accessed Students Data------   ");
System.out.println("============================================");
System.out.println("      ------Enter Your Choice------       ");
System.out.println("============================================");
System.out.println("1. For Viewing All The Records .");
System.out.println("2. For Finding a Record.");
System.out.println("3. For Searching A Record.");
System.out.println("4. For Adding a New Record.");
System.out.println("5. For Deleting a Record.");
System.out.println("6. For Returning to Main Menu.");
System.out.println("=============================================");
do {
System.out.println("Choice: ");
scan.hasNext();
op = scan.next().charAt(0);
switch(op) {
case '1':
System.out.println("--------------------------------");
findallOption();
break;
case '2':
System.out.println("--------------------------------");
findoneOption();
break;
case '3':
System.out.println("--------------------------------");
searchOption();
break;
case '4':
System.out.println("--------------------------------");
saveOption();
break;
case '5':
//deleting();
break;
case '6':
Option option = new Option();
System.out.println("....");
try {
option.optionSelect();
} catch (Exception e) {
e.printStackTrace();
}
break;
default:
System.out.println("Invalid Input!!!!\nEnter Again!!!");
break;
}
}while(!(op=='1'||op=='2'||op=='3'||op=='4'||op=='5'||op=='6'));
scan.close();
}
public static void  findallOption() {
System.out.println("--------------------------------");
System.out.println("FindAll");
List<Student> Students = studentData.findAll();
for (int i=0; i<Students.size(); i++) {
System.out.println(Students.get(i).toString());
}
System.out.println("\n\n\n\n\n");
```

```
try {
Thread.sleep(1000);
} catch (InterruptedException e) {
e.printStackTrace();
}
StudentsDataOptions();
}
public static void  findoneOption() {
System.out.println("--------------------------------");
System.out.println("Enter Id number of record to be Found");
Scanner scan = new Scanner(System.in);
int op = scan.nextInt();
Student Student = studentData.findOne(op);
System.out.println(Student.toString());
System.out.println("\n\n\n\n\n");
try {
Thread.sleep(1000);
} catch (InterruptedException e) {
e.printStackTrace();
}
StudentsDataOptions();
scan.close();
}
public static void  searchOption() {
Scanner scan = new Scanner(System.in);
System.out.println("--------------------------------");
System.out.println("Enter Student Roll number to search its data");
```

```
String stdCode = scan.nextLine();
List<Student> Students = studentData.search(stdCode);
for (int i=0; i<Students.size(); i++) {
System.out.println(Students.get(i).toString());
}
System.out.println("\n\n\n\n\n");
try {
Thread.sleep(1000);
} catch (InterruptedException e) {
e.printStackTrace();
}
StudentsDataOptions();
scan.close();
}
public static void  saveOption() {
System.out.println("--------------------------------");
Scanner scan = new Scanner(System.in);
Student student = new Student();
student.setSTUDENT_ID(1);
System.out.println("Enter Student Roll Number: ");
String studentRollNo = scan.nextLine();
student.setSTUDENT_ROLLNO(studentRollNo);
System.out.println("Enter Gender  ");
String studentGender = scan.nextLine();
Student.setSTUDENT DOB(Dte_of_Birth);
System.out.println("Enter DOB");
String student Marks = scan.nextLine();
```

```
Student.setSTUDENT Marks(Marks);
student = studentData.Save(student);
System.out.println(student.toString());
System.out.println("\n\n\n\n\n");
try {
Thread.sleep(1000);
} catch (InterruptedException e) {
e.printStackTrace();
}
StudentsDataOptions();
scan.close();
}
}

import java.io.*;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.Statement;

public class teacherreport1 extends javax.servlet.http.HttpServlet implements
javax.servlet.Servlet {
  static final long serialVersionUID = 1L;

   /* (non-Java-doc)
* @see javax.servlet.http.HttpServlet#HttpServlet()
*/
public teacherreport1() {
```

```
super();

}

/* (non-Java-doc)

* @see javax.servlet.http.HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

}

/* (non-Java-doc)

* @see javax.servlet.http.HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

String dname=request.getParameter("dname");

String tname=request.getParameter("tname");

String sub=request.getParameter("sub");

double i=0;

double tab[][]= new double[8][4];

double avg[]= new double[9];

double x1,y1,z1,v1;

PrintWriter out=response.getWriter();

try{

String str,str1,str2,sql;

int m,n;
```

```
double x,y,z,v;
for(m=0;m<8;m++)
{
for(n=0;n<4;n++)
{
tab[m][n]=0;
}
}
for(n=0;n<=8;n++)
{
avg[n]=0;
}
ResultSet rs;
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection c1=DriverManager.getConnection("jdbc:odbc:abc","root","root");
Statement s =c1.createStatement();
sql="select * from teacherreportA";
rs=s.executeQuery(sql);
while(rs.next())
{
str=rs.getString("dname");
str1=rs.getString("tname");
str2=rs.getString("sub");
if(dname.compareTo(str)==0 && tname.compareTo(str1)==0 && sub.compareTo(str2)==0)
{
v=rs.getInt(4);
x=rs.getInt(5);
```

```
y=rs.getInt(6);
z=rs.getInt(7);
i=v+x+y+z;
tab[0][0]=(v/i)*100;
tab[0][1]=(x/i)*100;
tab[0][2]=(y/i)*100;
tab[0][3]=(z/i)*100;
}
}
avg[0]=((tab[0][0]*10 + (tab[0][1]*7.5) + tab[0][2]*5 + (tab[0][3]*2.5))/100);
sql="select * from teacherreportB";
rs=s.executeQuery(sql);
while(rs.next())
{
str=rs.getString("dname");
str1=rs.getString("tname");
str2=rs.getString("sub");
if(dname.compareTo(str)==0 && tname.compareTo(str1)==0 && sub.compareTo(str2)==0)
{
v=rs.getInt(4);
x=rs.getInt(5);
y=rs.getInt(6);
z=rs.getInt(7);
i=v+x+y+z;
tab[1][0]=((v/i)*100);
tab[1][1]=((x/i)*100);
tab[1][2]=((y/i)*100);
```

```
tab[1][3]=((z/i)*100);
}
}
avg[1]=((tab[1][0]*10 + (tab[1][1]*7.5) + tab[1][2]*5 + (tab[1][3]*2.5))/100);
sql="select * from teacherreportC";
rs=s.executeQuery(sql);
while(rs.next())
{
str=rs.getString("dname");
str1=rs.getString("tname");
str2=rs.getString("sub");
if(dname.compareTo(str)==0 && tname.compareTo(str1)==0 && sub.compareTo(str2)==0)
{
v=rs.getInt(4);
x=rs.getInt(5);
y=rs.getInt(6);
z=rs.getInt(7);
i=v+x+y+z;
v1=(v/i)*100;
x1=(x/i)*100;
y1=(y/i)*100;
z1=(z/i)*100;
tab[2][0]+=v1;
tab[2][1]+=x1;
tab[2][2]+=y1;
tab[2][3]+=z1;
}
```

```
}
avg[2]=((tab[2][0]*10 + (tab[2][1]*7.5) + tab[2][2]*5 + (tab[2][3]*2.5))/100);
sql="select * from teacherreportD";
rs=s.executeQuery(sql);
while(rs.next())
{
str=rs.getString("dname");
str1=rs.getString("tname");
str2=rs.getString("sub");
if(dname.compareTo(str)==0 && tname.compareTo(str1)==0 && sub.compareTo(str2)==0)
{
v=rs.getInt(4);
x=rs.getInt(5);
y=rs.getInt(6);
z=rs.getInt(7);
i=v+x+y+z;
v1=(v/i)*100;
x1=(x/i)*100;
y1=(y/i)*100;
z1=(z/i)*100;
tab[3][0]+=v1;
tab[3][1]+=x1;
tab[3][2]+=y1;
tab[3][3]+=z1;
}
}
avg[3]=((tab[3][0]*10 + (tab[3][1]*7.5) + tab[3][2]*5 + (tab[3][3]*2.5))/100);
```

```
sql="select * from teacherreportE";
rs=s.executeQuery(sql);
while(rs.next())
{
str=rs.getString("dname");
str1=rs.getString("tname");
str2=rs.getString("sub");
if(dname.compareTo(str)==0 && tname.compareTo(str1)==0 && sub.compareTo(str2)==0)
{
v=rs.getInt(4);
x=rs.getInt(5);
y=rs.getInt(6);
z=rs.getInt(7);
i=v+x+y+z;
v1=(v/i)*100;
x1=(x/i)*100;
y1=(y/i)*100;
z1=(z/i)*100;
tab[4][0]+=v1;
tab[4][1]+=x1;
tab[4][2]+=y1;
tab[4][3]+=z1;
}
}
avg[4]=((tab[4][0]*10 + (tab[4][1]*7.5) + tab[4][2]*5 + (tab[4][3]*2.5))/100);
sql="select * from teacherreportF";
rs=s.executeQuery(sql);
```

```
while(rs.next())
{
str=rs.getString("dname");
str1=rs.getString("tname");
str2=rs.getString("sub");
if(dname.compareTo(str)==0 && tname.compareTo(str1)==0 && sub.compareTo(str2)==0)
{
v=rs.getInt(4);
x=rs.getInt(5);
y=rs.getInt(6);
z=rs.getInt(7);
i=v+x+y+z;
v1=(v/i)*100;
x1=(x/i)*100;
y1=(y/i)*100;
z1=(z/i)*100;
tab[5][0]+=v1;
tab[5][1]+=x1;
tab[5][2]+=y1;
tab[5][3]+=z1;
}
}
avg[5]=((tab[5][0]*10 + (tab[5][1]*7.5) + tab[5][2]*5 + (tab[5][3]*2.5))/100);
sql="select * from teacherreportG";
rs=s.executeQuery(sql);
while(rs.next())
{
```

```
str=rs.getString("dname");
str1=rs.getString("tname");
str2=rs.getString("sub");
if(dname.compareTo(str)==0 && tname.compareTo(str1)==0 && sub.compareTo(str2)==0)
{
v=rs.getInt(4);
x=rs.getInt(5);
y=rs.getInt(6);
z=rs.getInt(7);
i=v+x+y+z;
v1=(v/i)*100;
x1=(x/i)*100;
y1=(y/i)*100;
z1=(z/i)*100;
tab[6][0]+=v1;
tab[6][1]+=x1;
tab[6][2]+=y1;
tab[6][3]+=z1;
}
}
avg[6]=((tab[6][0]*10 + (tab[6][1]*7.5) + tab[6][2]*5 + (tab[6][3]*2.5))/100);
sql="select * from teacherreportH";
rs=s.executeQuery(sql);
while(rs.next())
{
str=rs.getString("dname");
str1=rs.getString("tname");
```

```
str2=rs.getString("sub");
if(dname.compareTo(str)==0 && tname.compareTo(str1)==0 && sub.compareTo(str2)==0)
{
v=rs.getInt(4);
x=rs.getInt(5);
y=rs.getInt(6);
z=rs.getInt(7);
i=v+x+y+z;
v1=(v/i)*100;
x1=(x/i)*100;
y1=(y/i)*100;
z1=(z/i)*100;
tab[7][0]+=v1;
tab[7][1]+=x1;
tab[7][2]+=y1;
tab[7][3]+=z1;
}
}
avg[7]=((tab[7][0]*10 + (tab[7][1]*7.5) + tab[7][2]*5 + (tab[7][3]*2.5))/100);
avg[8]=((avg[0]+avg[1]+avg[2]+avg[3]+avg[4]+avg[5]+avg[6]+avg[7])/8);
}catch(Exception e){out.println(e);}
if(i !=0)
{
out.println("<html>");
out.println("<head>");
out.println("<meta http-equiv="+"Content-Type"+" content="+"text/html; charset=ISO-8859-1"+">");
out.println("<title>Report</title>");
out.println("</head>");

out.println("<body style="+"background-color:cyan"+">");
out.println("<center>" +
"<img src=logo.png width=780"+" height="+"151"+" alt="+"logo"+">" +
"<marquee bgcolor="+"Yellow"+" behavior=alternate width="+"100% "+">TEACHER REPORT</marquee>" +
"</center>" +
"<br><br><br>" +
"<center>" +
"<table border="+"1"+" cellspacing=0 cellpadding=5 >" +
"<tr>" +
"<th align=left>NAME :</th>" +
"<th align=left>"+tname+"</th>" +
"<th align=left>DEPARTMENT :</th>" +
"<th align=left>"+dname+"</th>" +
"</tr>" +
"<tr>" +
"<th align=left>NO OF FORMS :</th>" +
"<th align=left>"+(int)i+"</th>" +
"<th align=left>SUBJECT :</th>" +
"<th align=left>"+sub+"</th>" +
"</tr>" +
"</table>" +
"<table border="+"1"+" cellspacing=0 cellpadding=5 >" +
"<tr>"+
"<th align=left>Performance Variables</th>" +
"<th>Excellent</th>" +
"<th>V. Good</th>" +
"<th>Good</th>" +
"<th>Fair</th>" +
"<th align=left>Wt. Avg Index</th>" +
"</tr>" +
"<tr>" +
"<th></th>" +
"<th>(10)</th>" +
"<th>(7.5)</th>" +
"<th>(5)</th>" +
"<th>(2.5)</th>" +
"<th></th>" +
"</tr>" +
"<caption>The Numbers Represent Student Response in % </caption>" +
"<tr>" +
"<th align=left>Communication Skills :</th>" +
"<th>"+((double)(Math.round(tab[0][0]*100))/100)+"</th>" +
"<th>"+((double)(Math.round(tab[0][1]*100))/100)+"</th>" +
"<th>"+((double)(Math.round(tab[0][2]*100))/100)+"</th>" +
"<th>"+((double)(Math.round(tab[0][3]*100))/100)+"</th>" +
"<th>"+((double)(Math.round(avg[0]*100))/100)+"</th>" +
"</tr>" +
"<tr>" +
"<th align=left>Ability To Explain & Clear Doubts :</th>" +
"<th>"+((double)(Math.round(tab[1][0]*100))/100)+"</th>" +
"<th>"+((double)(Math.round(tab[1][1]*100))/100)+"</th>" +
"<th>"+((double)(Math.round(tab[1][2]*100))/100)+"</th>" +
"<th>"+((double)(Math.round(tab[1][3]*100))/100)+"</th>" +
"<th>"+((double)(Math.round(avg[1]*100))/100)+"</th>" +
"</tr>" +
"<tr>" +
"<th align=left>Presentation:</th>" +
"<th>"+((double)(Math.round(tab[2][0]*100))/100)+"</th>" +
"<th>"+((double)(Math.round(tab[2][1]*100))/100)+"</th>" +
"<th>"+((double)(Math.round(tab[2][2]*100))/100)+"</th>" +
"<th>"+((double)(Math.round(tab[2][3]*100))/100)+"</th>" +
"<th>"+((double)(Math.round(avg[2]*100))/100)+"</th>" +
"</tr>" +
"<tr>" +
"<th align=left>Teaching Methodology :</th>" +
"<th>"+((double)(Math.round(tab[3][0]*100))/100)+"</th>" +
"<th>"+((double)(Math.round(tab[3][1]*100))/100)+"</th>" +
"<th>"+((double)(Math.round(tab[3][2]*100))/100)+"</th>" +
"<th>"+((double)(Math.round(tab[3][3]*100))/100)+"</th>" +
"<th>"+((double)(Math.round(avg[3]*100))/100)+"</th>" +
"</tr>" +
"<tr>" +
"<th align=left>Overall Interest Created :</th>" +
"<th>"+((double)(Math.round(tab[4][0]*100))/100)+"</th>" +
"<th>"+((double)(Math.round(tab[4][1]*100))/100)+"</th>" +
"<th>"+((double)(Math.round(tab[4][2]*100))/100)+"</th>" +
"<th>"+((double)(Math.round(tab[4][3]*100))/100)+"</th>" +
"<th>"+((double)(Math.round(avg[4]*100))/100)+"</th>" +

"</tr>" +

"<tr>" +

"<th align=left>Regular And Punctual :</th>" +

"<th>"+((double)(Math.round(tab[5][0]*100))/100)+"</th>" +

"<th>"+((double)(Math.round(tab[5][1]*100))/100)+"</th>" +

"<th>"+((double)(Math.round(tab[5][2]*100))/100)+"</th>" +

"<th>"+((double)(Math.round(tab[5][3]*100))/100)+"</th>" +

"<th>"+((double)(Math.round(avg[5]*100))/100)+"</th>" +

"</tr>" +

"<tr>" +

"<th align=left>Maintains Desciplain of Class :</th>" +

"<th>"+((double)(Math.round(tab[6][0]*100))/100)+"</th>" +

"<th>"+((double)(Math.round(tab[6][1]*100))/100)+"</th>" +

"<th>"+((double)(Math.round(tab[6][2]*100))/100)+"</th>" +

"<th>"+((double)(Math.round(tab[6][3]*100))/100)+"</th>" +

"<th>"+((double)(Math.round(avg[6]*100))/100)+"</th>" +

"</tr>" +

"<tr>" +

"<th align=left>Attitude Towards Student :</th>" +

"<th>"+((double)(Math.round(tab[7][0]*100))/100)+"</th>" +

"<th>"+((double)(Math.round(tab[7][1]*100))/100)+"</th>" +

"<th>"+((double)(Math.round(tab[7][2]*100))/100)+"</th>" +

"<th>"+((double)(Math.round(tab[7][3]*100))/100)+"</th>" +

"<th>"+((double)(Math.round(avg[7]*100))/100)+"</th>" +

"</tr>" +

"<th align=left>OVERALL RATINGS :</th>" +
"<th></th>" +
"<th></th>" +
"<th></th>" +
"<th></th>" +
"<th>"+((double)(Math.round(avg[8]*100))/100)+"</th>" +
"</tr>" +
"</table>" +
"<h3>" +
"<a href="+"adminhome.html"+">BACK TO HOME PAGE</a><br><br>" +
"</h3>" +
"</center>");
out.println("</body>");
out.println("</html>");
}
else
{
{out.println("No Feedback for This Teacher");}
}
}
import java.io.*;
import java.sql.*;
import javax.servlet.*;
PrintWriter out=response.getWriter();
    try{
      String dname=request.getParameter("dname");
      String tname=request.getParameter("tname");
      String sub=request.getParameter("sub");
      String a=request.getParameter("A");
      int a1=Integer.parseInt(a);
      String b=request.getParameter("B");
      int b1=Integer.parseInt(b);
      String c=request.getParameter("C");
      int c2=Integer.parseInt(c);
      String d=request.getParameter("D");
      int d2=Integer.parseInt(d);
      String e=request.getParameter("E");
      int e2=Integer.parseInt(e);
      String f=request.getParameter("F");
      int f1=Integer.parseInt(f);
      String g=request.getParameter("G");
      int g1=Integer.parseInt(g);
      String h=request.getParameter("H");
      int h1=Integer.parseInt(h);
      Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
      Connection c1=DriverManager.getConnection("jdbc:odbc:abc","root","root");
      Statement s =c1.createStatement();
      String sql=null;
      String sql2=null;
      ResultSet rs=null;
      int num=0;
      switch(a1)
      {
      case 1:
         sql2="select * from teacherreportA where tname='"+tname+"'";
```

```
                try
                {
                    rs=s.executeQuery(sql2);
                    rs.next();
                    num=rs.getInt("o1")+1;
                    sql="update teacherreportA set o1="+num+" where tname='"+tname+"'";
                }catch(SQLException e1)
                {
                    num=1;
                    sql="insert into teacherreportA 
values('"+dname+"','"+tname+"','"+sub+"',"+num+",0,0,0)";
                }
                break;
            case 2:
                sql2="select * from teacherreportA where tname='"+tname+"'";
                try
                {
                    rs=s.executeQuery(sql2);
                    rs.next();
                    num=rs.getInt("o2")+1;
                    sql="update teacherreportA set o2="+num+" where tname='"+tname+"'";
                }catch(SQLException e1)
                {
                    num=1;
                    sql="insert into teacherreportA 
values('"+dname+"','"+tname+"','"+sub+"',0,"+num+",0,0)";
                }
                break;
```

```
        case 3:
            sql2="select * from teacherreportA where tname='"+tname+"'";
            try
            {
                rs=s.executeQuery(sql2);
                rs.next();
                num=rs.getInt("o3")+1;
                sql="update teacherreportA set o3="+num+" where tname='"+tname+"'";
            }catch(SQLException e1)
            {
                num=1;
                sql="insert into teacherreportA
values('"+dname+"','"+tname+"','"+sub+"',0,0,"+num+",0)";
            }
            break;
        case 4:
            sql2="select * from teacherreportA where tname='"+tname+"'";
            try
            {
                rs=s.executeQuery(sql2);
                rs.next();
                num=rs.getInt("o4")+1;
                sql="update teacherreportA set o4="+num+" where tname='"+tname+"'";
            }catch(SQLException e1)
            {
                num=1;
                sql="insert into teacherreportA
values('"+dname+"','"+tname+"','"+sub+"',0,0,0,"+num+")";
```

```
                }
                break;
            }
            s.execute(sql);
            switch(b1)
            {
            case 1:
                sql2="select * from teacherreportB where tname='"+tname+"'";
                try
                {
                    rs=s.executeQuery(sql2);
                    rs.next();
                    num=rs.getInt("o1")+1;
                    sql="update teacherreportB set o1="+num+" where tname='"+tname+"'";
                }catch(SQLException e1)
                {
                    num=1;
                    sql="insert into teacherreportB
values('"+dname+"','"+tname+"','"+sub+"',"+num+",0,0,0)";
                }
                break;
            case 2:
                sql2="select * from teacherreportB where tname='"+tname+"'";
                try
                {
                    rs=s.executeQuery(sql2);
                    rs.next();
                    num=rs.getInt("o2")+1;
                    sql="update teacherreportB set o2="+num+" where tname='"+tname+"'";
                }catch(SQLException e1)
                {
                    num=1;
                    sql="insert into teacherreportB
values('"+dname+"','"+tname+"','"+sub+"',0,"+num+",0,0)";
                }
                break;
            case 3:
                sql2="select * from teacherreportB where tname='"+tname+"'";
                try
                {
                    rs=s.executeQuery(sql2);
                    rs.next();
                    num=rs.getInt("o3")+1;
                    sql="update teacherreportB set o3="+num+" where tname='"+tname+"'";
                }catch(SQLException e1)
                {
                    num=1;
                    sql="insert into teacherreportB
values('"+dname+"','"+tname+"','"+sub+"',0,0,"+num+",0)";
                }
                break;
            case 4:
                sql2="select * from teacherreportB where tname='"+tname+"'";
                try
                {
                    rs=s.executeQuery(sql2);
                    rs.next();
                    num=rs.getInt("o4")+1;
                    sql="update teacherreportB set o4="+num+" where tname='"+tname+"'";
                }catch(SQLException e1)
                {
                    num=1;
                    sql="insert into teacherreportB
values('"+dname+"','"+tname+"','"+sub+"',0,0,0,"+num+")";
                }
                break;
            }
            s.execute(sql);
            switch(c2)
            {
            case 1:
                sql2="select * from teacherreportC where tname='"+tname+"'";
                try
                {
                    rs=s.executeQuery(sql2);
                    rs.next();
                    num=rs.getInt("o1")+1;
                    sql="update teacherreportC set o1="+num+" where tname='"+tname+"'";
                }catch(SQLException e1)
                {
                    num=1;
                    sql="insert into teacherreportC
values('"+dname+"','"+tname+"','"+sub+"',"+num+",0,0,0)";
```

```
                }
                break;
            case 2:
                sql2="select * from teacherreportC where tname='"+tname+"'";
                try
                {
                    rs=s.executeQuery(sql2);
                    rs.next();
                    num=rs.getInt("o2")+1;
                    sql="update teacherreportC set o2="+num+" where tname='"+tname+"'";
                }catch(SQLException e1)
                {
                    num=1;
                    sql="insert into teacherreportC
values('"+dname+"','"+tname+"','"+sub+"',0,"+num+",0,0)";
                }
                break;
            case 3:
                sql2="select * from teacherreportC where tname='"+tname+"'";
                try
                {
                    rs=s.executeQuery(sql2);
                    rs.next();
                    num=rs.getInt("o3")+1;
                    sql="update teacherreportC set o3="+num+" where tname='"+tname+"'";
                }catch(SQLException e1)
                {
```

```
            num=1;
            sql="insert into teacherreportC
values('"+dname+"','"+tname+"','"+sub+"',0,0,"+num+",0)";
        }
        break;
    case 4:
        sql2="select * from teacherreportC where tname='"+tname+"'";
        try
        {
            rs=s.executeQuery(sql2);
            rs.next();
            num=rs.getInt("o4")+1;
            sql="update teacherreportC set o4="+num+" where tname='"+tname+"'";
        }catch(SQLException e1)
        {
            num=1;
            sql="insert into teacherreportC
values('"+dname+"','"+tname+"','"+sub+"',0,0,0,"+num+")";
        }
        break;
    }
    s.execute(sql);
    switch(d2)
    {
    case 1:
        sql2="select * from teacherreportD where tname='"+tname+"'";
        try
        {
```

```
                    rs=s.executeQuery(sql2);
                    rs.next();
                    num=rs.getInt("o1")+1;
                    sql="update teacherreportD set o1="+num+" where tname='"+tname+"'";
                }catch(SQLException e1)
                {
                    num=1;
                    sql="insert into teacherreportD
values('"+dname+"','"+tname+"','"+sub+"',"+num+",0,0,0)";
                }
                break;
            case 2:
                sql2="select * from teacherreportD where tname='"+tname+"'";
                try
                {
                    rs=s.executeQuery(sql2);
                    rs.next();
                    num=rs.getInt("o2")+1;
                    sql="update teacherreportD set o2="+num+" where tname='"+tname+"'";
                }catch(SQLException e1)
                {
                    num=1;
                    sql="insert into teacherreportD
values('"+dname+"','"+tname+"','"+sub+"',0,"+num+",0,0)";
                }
                break;
            case 3:
                sql2="select * from teacherreportD where tname='"+tname+"'";
                try
                {
                    rs=s.executeQuery(sql2);
                    rs.next();
                    num=rs.getInt("o3")+1;
                    sql="update teacherreportD set o3="+num+" where tname='"+tname+"'";
                }catch(SQLException e1)
                {
                    num=1;
                    sql="insert into teacherreportD 
values('"+dname+"','"+tname+"','"+sub+"',0,0,"+num+",0)";
                }
                break;
            case 4:
                sql2="select * from teacherreportD where tname='"+tname+"'";
                try
                {
                    rs=s.executeQuery(sql2);
                    rs.next();
                    num=rs.getInt("o4")+1;
                    sql="update teacherreportD set o4="+num+" where tname='"+tname+"'";
                }catch(SQLException e1)
                {
                    num=1;
                    sql="insert into teacherreportD 
values('"+dname+"','"+tname+"','"+sub+"',0,0,0,"+num+")";
                }
                break;
            }
            s.execute(sql);
            switch(e2)
            {
            case 1:
                sql2="select * from teacherreportE where tname='"+tname+"'";
                try
                {
                    rs=s.executeQuery(sql2);
                    rs.next();
                    num=rs.getInt("o1")+1;
                    sql="update teacherreportE set o1="+num+" where tname='"+tname+"'";
                }catch(SQLException e1)
                {
                    num=1;
                    sql="insert into teacherreportE
values('"+dname+"','"+tname+"','"+sub+"',"+num+",0,0,0)";
                }
                break;
            case 2:
                sql2="select * from teacherreportE where tname='"+tname+"'";
                try
                {
                    rs=s.executeQuery(sql2);
                    rs.next();
                    num=rs.getInt("o2")+1;
                    sql="update teacherreportE set o2="+num+" where tname='"+tname+"'";
```

```
        }catch(SQLException e1)
        {
          num=1;
          sql="insert into teacherreportE
values('"+dname+"','"+tname+"','"+sub+"',0,"+num+",0,0)";
        }
        break;
      case 3:
        sql2="select * from teacherreportE where tname='"+tname+"'";
        try
        {
          rs=s.executeQuery(sql2);
          rs.next();
          num=rs.getInt("o3")+1;
          sql="update teacherreportE set o3="+num+" where tname='"+tname+"'";
        }catch(SQLException e1)
        {
          num=1;
          sql="insert into teacherreportE
values('"+dname+"','"+tname+"','"+sub+"',0,0,"+num+",0)";
        }
        break;
      case 4:
        sql2="select * from teacherreportE where tname='"+tname+"'";
        try
        {
          rs=s.executeQuery(sql2);
          rs.next();
```

```
                    num=rs.getInt("o4")+1;
                    sql="update teacherreportE set o4="+num+" where tname='"+tname+"'";
                }catch(SQLException e1)
                {
                    num=1;
                    sql="insert into teacherreportE
values('"+dname+"','"+tname+"','"+sub+"',0,0,0,"+num+")";
                }
                break;
            }
            s.execute(sql);
            switch(f1)
            {
            case 1:
                sql2="select * from teacherreportF where tname='"+tname+"'";
                try
                {
                    rs=s.executeQuery(sql2);
                    rs.next();
                    num=rs.getInt("o1")+1;
                    sql="update teacherreportF set o1="+num+" where tname='"+tname+"'";
                }catch(SQLException e1)
                {
                    num=1;
                    sql="insert into teacherreportF
values('"+dname+"','"+tname+"','"+sub+"',"+num+",0,0,0)";
                }
                break;
            case 2:
                sql2="select * from teacherreportF where tname='"+tname+"'";
                try
                {
                    rs=s.executeQuery(sql2);
                    rs.next();
num=rs.getInt("o2")+1;
                    sql="update teacherreportF set o2="+num+" where tname='"+tname+"'";
                }catch(SQLException e1)
                {
                    num=1;
                    sql="insert into teacherreportF
values('"+dname+"','"+tname+"','"+sub+"',0,"+num+",0,0)";
                }
                break;
            case 3:
                sql2="select * from teacherreportF where tname='"+tname+"'";
                try
                {
                    rs=s.executeQuery(sql2);
                    rs.next();
                    num=rs.getInt("o3")+1;
                    sql="update teacherreportF set o3="+num+" where tname='"+tname+"'";
                }catch(SQLException e1)
                {
                    num=1;
                    sql="insert into teacherreportF
values('"+dname+"','"+tname+"','"+sub+"',0,0,"+num+",0)";
```

```
                }
                break;
            case 4:
                sql2="select * from teacherreportF where tname='"+tname+"'";
                try
                {
                    rs=s.executeQuery(sql2);
                    rs.next();
                    num=rs.getInt("o4")+1;
                    sql="update teacherreportF set o4="+num+" where tname='"+tname+"'";
                }catch(SQLException e1)
                {
                    num=1;
                    sql="insert into teacherreportF
values('"+dname+"','"+tname+"','"+sub+"',0,0,0,"+num+")";
                }
                break;
            }
            s.execute(sql);
            switch(g1)
            {
            case 1:
                sql2="select * from teacherreportG where tname='"+tname+"'";
                try
                {
                    rs=s.executeQuery(sql2);
                    rs.next();
```

```
          num=rs.getInt("o1")+1;
          sql="update teacherreportG set o1="+num+" where tname='"+tname+"'";
        }catch(SQLException e1)
        {
          num=1;
          sql="insert into teacherreportG
values('"+dname+"','"+tname+"','"+sub+"',"+num+",0,0,0)";
        }
        break;
      case 2:
        sql2="select * from teacherreportG where tname='"+tname+"'";
        try
        {
          rs=s.executeQuery(sql2);
          rs.next();
          num=rs.getInt("o2")+1;
          sql="update teacherreportG set o2="+num+" where tname='"+tname+"'";
        }catch(SQLException e1)
        {
          num=1;
          sql="insert into teacherreportG
values('"+dname+"','"+tname+"','"+sub+"',0,"+num+",0,0)";
        }
        break;
      case 3:
        sql2="select * from teacherreportG where tname='"+tname+"'";
        try
        {
          rs=s.executeQuery(sql2);
          rs.next();
          num=rs.getInt("o3")+1;
          sql="update teacherreportG set o3="+num+" where tname='"+tname+"'";
        }catch(SQLException e1)
        {
          num=1;
          sql="insert into teacherreportG
values('"+dname+"','"+tname+"','"+sub+"',0,0,"+num+",0)";
        }
        break;
      case 4:
        sql2="select * from teacherreportG where tname='"+tname+"'";
        try
        {
          rs=s.executeQuery(sql2);
          rs.next();
          num=rs.getInt("o4")+1;
          sql="update teacherreportG set o4="+num+" where tname='"+tname+"'";
        }catch(SQLException e1)
        {
          num=1;
          sql="insert into teacherreportG
values('"+dname+"','"+tname+"','"+sub+"',0,0,0,"+num+")";
        }
        break;
      }
      s.execute(sql);
```

```
    switch(h1)
    {
    case 1:
       sql2="select * from teacherreportH where tname='"+tname+"'";
       try
       {
          rs=s.executeQuery(sql2);
          rs.next();
          num=rs.getInt("o1")+1;
          sql="update teacherreportH set o1="+num+" where tname='"+tname+"'";
       }catch(SQLException e1)
       {
          num=1;
          sql="insert into teacherreportH
values('"+dname+"','"+tname+"','"+sub+"',"+num+",0,0,0)";
       }
       break;
    case 2:
       sql2="select * from teacherreportH where tname='"+tname+"'";
       try
       {
          rs=s.executeQuery(sql2);
          rs.next();
          num=rs.getInt("o2")+1;
          sql="update teacherreportH set o2="+num+" where tname='"+tname+"'";
       }catch(SQLException e1)
       {
          num=1;
          sql="insert into teacherreportH
values('"+dname+"','"+tname+"','"+sub+"',0,"+num+",0,0)";
        }
        break;
      case 3:
        sql2="select * from teacherreportH where tname='"+tname+"'";
        try
        {
          rs=s.executeQuery(sql2);
          rs.next();
          num=rs.getInt("o3")+1;
          sql="update teacherreportH set o3="+num+" where tname='"+tname+"'";
        }catch(SQLException e1)
        {
          num=1;
          sql="insert into teacherreportH
values('"+dname+"','"+tname+"','"+sub+"',0,0,"+num+",0)";
        }
        break;
      case 4:
        sql2="select * from teacherreportH where tname='"+tname+"'";
        try
        {
          rs=s.executeQuery(sql2);
          rs.next();
          num=rs.getInt("o4")+1;
          sql="update teacherreportH set o4="+num+" where tname='"+tname+"'";
        }catch(SQLException e1)
        {
          num=1;
          sql="insert into teacherreportH 
values('"+dname+"','"+tname+"','"+sub+"',0,0,0,"+num+")";
        }
        break;
      }
      s.execute(sql);



    }catch(Exception e){out.println(e);}
      }
```

OUTPUT:

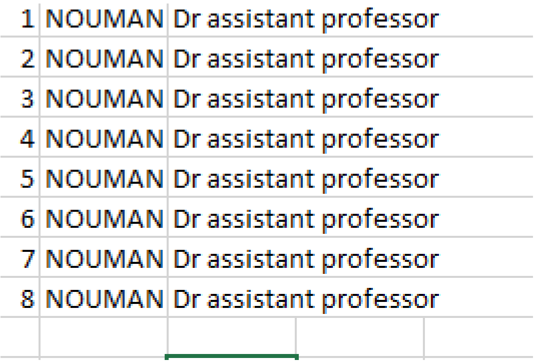
```
num=rs.getInt("o2")+1;
          sql="update teacherreportF set o2="+num+" where tname='"+tname+"'";
        }catch(SQLException e1)
        {
          num=1;
          sql="insert into teacherreportF 
values('"+dname+"','"+tname+"','"+sub+"',0,"+num+",0,0)";
        }
        break;
      case 3:
        sql2="select * from teacherreportF where tname='"+tname+"'";
        try
        {
          rs=s.executeQuery(sql2);
```

```
                    rs.next();
                    num=rs.getInt("o3")+1;
                    sql="update teacherreportF set o3="+num+" where tname='"+tname+"'";
                }catch(SQLException e1)
                {
                    num=1;
                    sql="insert into teacherreportF
values('"+dname+"','"+tname+"','"+sub+"',0,0,"+num+",0)";
                }
                break;
            case 4:
                sql2="select * from teacherreportF where tname='"+tname+"'";
                try
                {
                    rs=s.executeQuery(sql2);
                    rs.next();
                    num=rs.getInt("o4")+1;
                    sql="update teacherreportF set o4="+num+" where tname='"+tname+"'";
                }catch(SQLException e1)
                {
                    num=1;
                    sql="insert into teacherreportF
values('"+dname+"','"+tname+"','"+sub+"',0,0,0,"+num+")";
                }
                break;
            }
            s.execute(sql);
            switch(g1)
            {
            case 1:
                sql2="select * from teacherreportG where tname='"+tname+"'";
                try
                {
                    rs=s.executeQuery(sql2);
                    rs.next();
                    num=rs.getInt("o1")+1;
                    sql="update teacherreportG set o1="+num+" where tname='"+tname+"'";
                }catch(SQLException e1)
                {
                    num=1;
                    sql="insert into teacherreportG
values('"+dname+"','"+tname+"','"+sub+"',"+num+",0,0,0)";
                }
                break;
            case 2:
                sql2="select * from teacherreportG where tname='"+tname+"'";
                try
                {
                    rs=s.executeQuery(sql2);
                    rs.next();
                    num=rs.getInt("o2")+1;
                    sql="update teacherreportG set o2="+num+" where tname='"+tname+"'";
                }catch(SQLException e1)
                {
                    num=1;
                    sql="insert into teacherreportG
values('"+dname+"','"+tname+"','"+sub+"',0,"+num+",0,0)";
                }
                break;
            case 3:
                sql2="select * from teacherreportG where tname='"+tname+"'";
                try
                {
                    rs=s.executeQuery(sql2);
                    rs.next();
                    num=rs.getInt("o3")+1;
                    sql="update teacherreportG set o3="+num+" where tname='"+tname+"'";
                }catch(SQLException e1)
                {
                    num=1;
                    sql="insert into teacherreportG
values('"+dname+"','"+tname+"','"+sub+"',0,0,"+num+",0)";
                }
                break;
            case 4:
                sql2="select * from teacherreportG where tname='"+tname+"'";
                try
                {
                    rs=s.executeQuery(sql2);
                    rs.next();
                    num=rs.getInt("o4")+1;
                    sql="update teacherreportG set o4="+num+" where tname='"+tname+"'";
                }catch(SQLException e1)
```

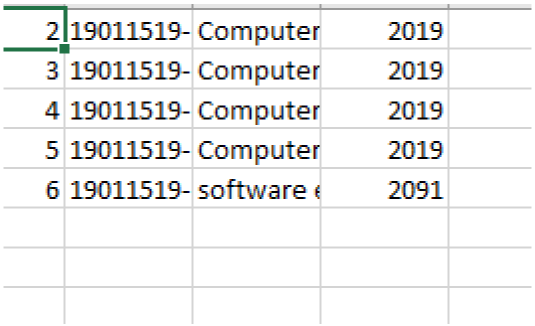
```
                {
                    num=1;
                    sql="insert into teacherreportG
values('"+dname+"','"+tname+"','"+sub+"',0,0,0,"+num+")";
                }
                break;
            }
            s.execute(sql);
            switch(h1)
            {
            case 1:
                sql2="select * from teacherreportH where tname='"+tname+"'";
                try
                {
                    rs=s.executeQuery(sql2);
                    rs.next();
                    num=rs.getInt("o1")+1;
                    sql="update teacherreportH set o1="+num+" where tname='"+tname+"'";
                }catch(SQLException e1)
                {
                    num=1;
                    sql="insert into teacherreportH
values('"+dname+"','"+tname+"','"+sub+"',"+num+",0,0,0)";
                }
                break;
            case 2:
                sql2="select * from teacherreportH where tname='"+tname+"'";
                try
```

```
          {
            rs=s.executeQuery(sql2);
            rs.next();
            num=rs.getInt("o2")+1;
            sql="update teacherreportH set o2="+num+" where tname='"+tname+"'";
          }catch(SQLException e1)
          {
            num=1;
            sql="insert into teacherreportH
values('"+dname+"','"+tname+"','"+sub+"',0,"+num+",0,0)";
          }
          break;
        case 3:
          sql2="select * from teacherreportH where tname='"+tname+"'";
          try
          {
            rs=s.executeQuery(sql2);
            rs.next();
            num=rs.getInt("o3")+1;
            sql="update teacherreportH set o3="+num+" where tname='"+tname+"'";
          }catch(SQLException e1)
          {
            num=1;
            sql="insert into teacherreportH
values('"+dname+"','"+tname+"','"+sub+"',0,0,"+num+",0)";
          }
          break;
        case 4:
                sql2="select * from teacherreportH where tname='"+tname+"'";
                try
                {
                    rs=s.executeQuery(sql2);
                    rs.next();
                    num=rs.getInt("o4")+1;
                    sql="update teacherreportH set o4="+num+" where tname='"+tname+"'";
                }catch(SQLException e1)
                {
                    num=1;
                    sql="insert into teacherreportH
values('"+dname+"','"+tname+"','"+sub+"',0,0,0,"+num+")";
                }
                break;
            }
            s.execute(sql);



        }catch(Exception e){out.println(e);}
        response.sendRedirect("thank.html");
    }
```

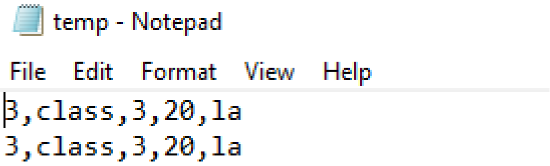
OUTPUT:



| | | |
|---|---|---|
| 1 | NOUMAN | Dr assistant professor |
| 2 | NOUMAN | Dr assistant professor |
| 3 | NOUMAN | Dr assistant professor |
| 4 | NOUMAN | Dr assistant professor |
| 5 | NOUMAN | Dr assistant professor |
| 6 | NOUMAN | Dr assistant professor |
| 7 | NOUMAN | Dr assistant professor |
| 8 | NOUMAN | Dr assistant professor |



| | | | |
|---|---|---|---|
| 2 | 19011519- | Computer | 2019 |
| 3 | 19011519- | Computer | 2019 |
| 4 | 19011519- | Computer | 2019 |
| 5 | 19011519- | Computer | 2019 |
| 6 | 19011519- | software e | 2091 |



```
3,class,3,20,la
3,class,3,20,la
```