```java
/**
 ******************************************************************************
 *                              *****************                             *
 ******************************* FINAL VERSION ********************************
 *                              *****************                             *
 *                                                                            *
 * PROGRAM: PROJECT MARKING ASSISTANT                                         *
 *                                                                            *
 * AUTHOR: ALEKSANDAR TSANKOV MLADENOV                                        *
 *                                                                            *
 * STUDENT NUMBER: 2976196                                                    *
 *                                                                            *
 * ****************************************************************************
 * */


import java.io.*;
import java.text.*;
import java.util.*;
import java.util.regex.Pattern;

public class MainClass {

    public static ArrayList<String> projectSkeleton = new ArrayList<String>(); // Scans for subjects and add them to table
    public static ArrayList<Project> projectList = new ArrayList<Project>(); // Stores every project info
    private static File test = new File(""); // used to point to files
    private static ArrayList<Project> temp = new ArrayList<Project>(); // temp array for storing changes
    private static Project project = new Project(); // single project variable for everyone
    private static Scanner scanRead = new Scanner(System.in); // used scanner as usually more is more simple
    private static BufferedReader buffRead = new BufferedReader(new InputStreamReader(System.in)); //some instances had to be used as
the Scanner class was showing bugs for no apparent reason
    private static String studentID;
    private static String studentNumber;
    private static double mark;
    private static DecimalFormat df2 = new DecimalFormat("#.##");

    static void Print() { // method is used for printing subjects which can be any number and projects

        Boolean firstRun = true;

        for (int i = 0; i < temp.size(); i++) {
            if (firstRun == true) {
                for (int j = 0; j < projectSkeleton.size(); j++) {
                    System.out.print((projectSkeleton.get(j) + "               ").substring(0, 16));
                }
                System.out.println("");
                firstRun = false;
            }
            Format(i);
        }
    }

    static void Print2(String a) { // method prints only amended project for grater clarity
    for (int i = 0; i < temp.size(); i++) {
        project = temp.get(i);
        if (project.studentId.equalsIgnoreCase(a)) {
            for (int j = 0; j < projectSkeleton.size(); j++) {
                System.out.print((projectSkeleton.get(j) + "               ").substring(0, 16));
            }
            System.out.println("");
            Format(i);
            break;
        }
    }

}

static void Format(int i) { // gives a format to above print methods
    String format;
    project = temp.get(i);
    if (project.studentId.equals(null)) {
        System.out.print("         -          ");
    } else {
        format = project.studentId.concat("                    ");
        System.out.print(format.substring(0, 16));
    }
    if (project.studentNumber == null) {
        System.out.print("       -        ");
    } else {
        format = " " + project.studentNumber.concat("                ");
        System.out.print(format.substring(0, 16));
    }
    if (Double.isNaN(project.mark1)) {
        System.out.print(" -              ");
    } else {
        format = " " + String.valueOf(project.mark1).concat("                ");
        System.out.print(format.substring(0, 16));
    }
    if (Double.isNaN(project.mark2)) {
        System.out.print(" -              ");
    } else {
        format = " " + String.valueOf(project.mark2).concat("                ");
        System.out.print(format.substring(0, 16));
    }
    if (Double.isNaN(project.mark3)) {
        System.out.print(" -              ");
    } else {
        format = " " + String.valueOf(project.mark3).concat("                ");
        System.out.print(format.substring(0, 16));
    }
    if (Double.isNaN(project.mark4)) {
        System.out.print(" -              ");
    } else {
        format = " " + String.valueOf(project.mark4).concat("                ");
        System.out.print(format.substring(0, 16));
    }
```

```
        if (Double.isNaN(project.mark5)) {
            System.out.print(" -              ");
        } else {
            format = " " + String.valueOf(project.mark5).concat("                ");
            System.out.print(format.substring(0, 16));
        }
        if (Double.isNaN(project.mark6)) {
            System.out.print(" -              ");
        } else {
            format = " " + String.valueOf(project.mark6).concat("                ");
            System.out.print(format.substring(0, 16));
        }
        if (project.total < 0) {
            System.out.print(" No data  ");
        } else {
            format = " " + String.valueOf(project.total).concat("                ");
            System.out.print(format.substring(0, 16));
        }
        System.out.println("");
    }

    static void AddNewProject() throws Exception { // used to add projects to list
        String choice = "y";
        while (choice.equalsIgnoreCase("y")) {
            Boolean correctFormat = false;
            System.out.println("Create student ID (Student name)");
            studentID = buffRead.readLine();
            while (true) {
                System.out.println("Create student Number");
                studentNumber = buffRead.readLine();
                correctFormat = Pattern.matches("[PpFf]{1}[0-9]{6,7}", studentNumber);
                if (correctFormat == true)
                    break;
                if (correctFormat == false)
                    System.out.println(
                            "\nIncorect Student Number format.\nUse P or F (Upper or Lower Case) and add 6-
7 digits after that:\nExample p456723\nExample P7900897\nExample F4589963\nExample f930467\n ");
            }
            project = new Project(studentID, studentNumber);
            temp.add(project);
            Print2(studentID);
            System.out.println("\nEnter marks for Attendace 0-5:");
            mark = scanRead.nextDouble();
            project.Subject1(mark);
            Print2(studentID);
            System.out.println("\nEnter marks for Final Build 0-10:");
            mark = scanRead.nextDouble();
            project.Subject2(mark);
            project.Total();
            Print2(studentID);
            System.out.println("\nEnter marks for Data Quality 0-15:");
            mark = scanRead.nextDouble();
            project.Subject3(mark);
            Print2(studentID);
```

```java
            System.out.println("\nEnter marks for Colaboration 0-20:");
            mark = scanRead.nextDouble();
            project.Subject4(mark);
            project.Total();
            Print2(studentID);
            System.out.println("\nEnter marks for Research 0-20:");
            mark = scanRead.nextDouble();
            project.Subject5(mark);
            Print2(studentID);
            System.out.println("\nEnter marks for Dissertation 0-30:");
            mark = scanRead.nextDouble();
            project.Subject6(mark);
            project.Total();
            Print2(studentID);
            System.out.println("\nAdd another project Y/N ?");
            choice = buffRead.readLine();
        }
    }

    static void EnterProjectMarks() throws Exception { // mathod to change and amend existing projects
        String choice = "y";
        while (choice.equalsIgnoreCase("y")) {
            String studentID;
            System.out.println("Enter student ID (Student name)");
            studentID = buffRead.readLine();
            Print2(studentID);
            for (int i = 0; i < temp.size(); i++) {
                project = temp.get(i);
                if (project.studentId.equalsIgnoreCase(studentID.trim())) {
                    System.out.println("\nEnter marks for Attendace 0-5:");
                    mark = scanRead.nextDouble();
                    project.Subject1(mark);
                    Print2(studentID);
                    System.out.println("\nEnter marks for Final Build 0-10:");
                    mark = scanRead.nextDouble();
                    project.Subject2(mark);
                    Print2(studentID);
                    System.out.println("\nEnter marks for Data Quality 0-15:");
                    mark = scanRead.nextDouble();
                    project.Subject3(mark);
                    Print2(studentID);
                    System.out.println("\nEnter marks for Colaboration 0-20:");
                    mark = scanRead.nextDouble();
                    project.Subject4(mark);
                    Print2(studentID);
                    System.out.println("\nEnter marks for Research 0-20:");
                    mark = scanRead.nextDouble();
                    project.Subject5(mark);
                    Print2(studentID);
                    System.out.println("\nEnter marks for Dissertation 0-30:");
                    mark = scanRead.nextDouble();
                    project.Subject6(mark);
                    Print2(studentID);
                    project.Total();
                    Print2(studentID);
                    System.out.println("\nEnter more project marks Y/N ?");
                    choice = buffRead.readLine();
                    break;
                }
                if (i == temp.size() - 1) {
                    System.out.println("\n***************************" + "\n*     ENTRY NOT FOUND!    *"
                            + "\n***************************");
                    System.out.println("\nEnter more project marks Y/N ?");
                    choice = buffRead.readLine();
                    break;
                }

            }
        }

    }

    static void DeleteProject() throws Exception { // method for project delition
        String choice = "y";
        while (choice.equalsIgnoreCase("y")) {
            String studentID;
            System.out.println("Enter student ID (Student name)");
            studentID = buffRead.readLine();
            Print2(studentID);
            for (int i = 0; i < temp.size(); i++) {
                project = temp.get(i);
                if (project.studentId.equalsIgnoreCase(studentID)) {
                    System.out.println("\nDelete project entry " + studentID + " Y/N ?");
                    choice = buffRead.readLine();
                    if (choice.equalsIgnoreCase("Y")) {
                        project = temp.remove(i);
                        System.out.println("\n***************************" + "\n*         DELETED !       *"
                                + "\n***************************");
                        System.out.println("\nDelete more projects Y/N ??");
                        choice = buffRead.readLine();
                        break;
                    }
                    if (choice.equalsIgnoreCase("N")) {
                        System.out.println("\n***************************" + "\n*         CANCELED !      *"
                                + "\n***************************");
                        System.out.println("\nDelete more projects Y/N ?");
                        choice = buffRead.readLine();
                        break;
                    }
                }
                if (i == temp.size() - 1) {
                    System.out.println("\n***************************" + "\n*     ENTRY NOT FOUND!    *"
                            + "\n***************************");
                    System.out.println("\nDelete more projects Y/N ?");
                    choice = buffRead.readLine();
                    break;
                }
            }
        }

    }

    public static void main(String[] args) throws Exception {
        // TODO Auto-generated method stub

        String checkFile = "";
        System.out.println("Enter project file name and extencion ( Default - Project.csv ) :");
        checkFile = scanRead.next();
        test = new File(checkFile);
        Boolean error = test.exists(); // initial check if project file exists
        if (error == false) { // if not if initialises
            System.out.println("\n***************************"
                    + "\n*FILE NOT FOUND, EXITING !*"
                    + "\n***************************");
        }
        if (error == true) { // if file exists the rest of the program is nested in this tatement and trigers
            FileWriter fileW = new FileWriter(test, true);
            PrintWriter printW = new PrintWriter(fileW);
            BufferedReader reader = new BufferedReader(new FileReader(test));
            String line;
            Boolean splitArray1 = true;

            while ((line = reader.readLine()) != null) { // loop reads file

                String[] splitArray = line.split(",");// reads subjects

                if (splitArray1 == true) {
                    for (int i = 0; i < splitArray.length; i++) {
                        projectSkeleton.add(splitArray[i]); // writes subjects in to separate ArrayList
                    }

                    splitArray1 = false;
                } else { // adds project to another ArrayList
                    for (int i = 0; i < 1; i++) {

                        project = new Project(splitArray[0], splitArray[1]);
                        project.Subject1(Double.parseDouble(splitArray[2]));
                        project.Subject2(Double.parseDouble(splitArray[3]));
                        project.Subject3(Double.parseDouble(splitArray[4]));
                        project.Subject4(Double.parseDouble(splitArray[5]));
                        project.Subject5(Double.parseDouble(splitArray[6]));
                        project.Subject6(Double.parseDouble(splitArray[7]));
                        project.Total();
                        projectList.add(project);
                    }

                }

            }

            temp = new ArrayList<Project>(projectList);// copies from project
```

```java
while (true) { // heart of the program is below while loop which references all methods
    int choice;
    Print();
    System.out.println(
            "\n----------------------------" + "\nChoose option Below" + "\n----------------------------"
                    + "\n1. Enter project marks" + "\n2. Add new project" + "\n3. Delete Project"
                    + "\n4. Save and Exit" + "\n5. Exit without Saving" + "\n----------------------------");
    choice = scanRead.nextInt();
    if (choice == 1) { // calls on appropriate method to change existing marks
        EnterProjectMarks();
    }
    if (choice == 2) { // calls on appropriate method to add new project
        AddNewProject();
    }
    if (choice == 3) { // calls on appropriate method to delete project
        DeleteProject();
    }
    if (choice == 4) { // Save project
        FileWriter fileW1 = new FileWriter(test); // below 4 lines of code clear any data in the existing file
        PrintWriter printW1 = new PrintWriter(fileW1);
        printW1.print("");
        printW1.close();
        for (int j = 0; j < projectSkeleton.size(); j++) { // marking criteria is added to clear file first
            printW.print(projectSkeleton.get(j) + ",");
        }
        printW.print("\n");
        for (int i = 0; i < temp.size(); i++) { // populates project
            project = temp.get(i);
            printW.print(project.studentId + ",");
            printW.print(project.studentNumber + ",");
            printW.print(project.mark1 + ",");
            printW.print(project.mark2 + ",");
            printW.print(project.mark3 + ",");
            printW.print(project.mark4 + ",");
            printW.print(project.mark5 + ",");
            printW.print(project.mark6 + ",");
            printW.print(project.total + ",");
            printW.print("\n");
        }
        printW.close();
        System.out.println("\n***************************" + "\n*  EXITED CHANGES SAVED   *"
                + "\n***************************");
        break;
    }
    if (choice == 5) { // exits program without changes
        temp.clear();
        System.out.println("\n***************************" + "\n*    EXITED NO CHANGES    *"
                + "\n***************************");

        break;

    }

}
```

```
        }
    }
}
```