Giorgi Samushia

Cmpt 360 Project

Student Class

```
//Student class with all necessary members
public class Student {
  private String student_id;
  private String[] course_id = new String[200];
  private char[] grade = new char[200];
  private int numOfCourses = 0;
  private int numOfGrades = 0;
  public Student () {
  }
  public Student(String student_id, String course_id, char grade) {
    this.student_id = student_id;
    this.course_id[numOfCourses] = course_id;
    this.grade[numOfGrades] = grade;
    numOfCourses++;
    numOfGrades++;
  }
  public String getStudentID() {
    return student_id;
  }
  public String getCourseID(int index) {
    return course_id[index];
  }
  public char getGrade(int index) {
    return grade[index];
  }
```

```
    public int getNumOfCourses() {
        return numOfCourses;
    }
    public void setStudentID(String student_id) {
        this.student_id = student_id;
    }
    public void setCourseID(String course_id) {
        this.course_id[numOfCourses] = course_id;
        numOfCourses++;
    }
    public void setGrade(char grade) {
        this.grade[numOfGrades] = grade;
        numOfGrades++;
    }

}
```

Course Class

```
//Course class with all necessary members
public class Course {
    private String course_id;
    private String course_name;
    public Course () {
        course_id = "";
        course_name = "";
    }
    public Course (String course_id, String course_name) {
        this.course_id = course_id;
        this.course_name = course_name;
    }
    public String getCourseID() {
        return course_id;
    }
    public String getCourseName() {
        return course_name;
    }
}
```

Main Class

```
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.ArrayList;
import java.util.Scanner;

public class Main {

    //Main
    public static void main(String[] args) throws FileNotFoundException, IOException {
        //Declaring the files and arraylists to hold values
        String student_id;
        String course_id;
        char grade;
        Scanner input = new Scanner(System.in);
```

```
File s = new File("C:\\Users\\giorg\\Downloads\\student.txt");

File c = new File("C:\\Users\\giorg\\Downloads\\course.txt");

ArrayList<Student> students = new ArrayList<>();

ArrayList<Course> courses = new ArrayList<>();

putStudentsInList(students, s);

putCoursesInList(courses, c);

//Displaying the menu until the user enters 4

while(true) {

  System.out.println("\n1. Get student information");

  System.out.println("2. Get course information");

  System.out.println("3. Insert a new record in the student file");

  System.out.println("4. Exit");

  System.out.print("Enter your selection: ");

  int choice = input.nextInt();

  switch(choice) {

    case 1:

      System.out.print("Enter the student ID: ");

      student_id = input.next();

      getStudentInfo(student_id, students, courses);

      break;

    case 2:

      System.out.print("Enter the course ID: ");

      course_id = input.next();

      getCourseInfo(course_id, courses, students);

      break;

    case 3:

      System.out.print("Enter the student ID:");

      student_id = input.next();

      System.out.print("Enter the course ID:");
```

```
                course_id = input.next();
                System.out.print("Enter the course grade:");
                grade = input.next().charAt(0);
                addStudent(s, students, courses, student_id, course_id, grade);
                break;
              case 4:
                System.out.println("Thank you and goodbye!");
                return;
              default:
                System.out.println("Invalid input, try again:");
                break;
            }
          }
        }

        //Method that puts students from the file into a Student type array list
        public static void putStudentsInList(ArrayList<Student> students, File s) throws FileNotFoundException
      {
          Scanner in = new Scanner(s);
          int count = 1;
          while(in.hasNext()) {
            String str = in.nextLine();
            String student_id = "";
            String course_id = "";
            char grade = ' ';
            //I googled this method to split strings
            String[] splitted = str.split(",");
            student_id = splitted[0];
            course_id = splitted[1];
        grade = splitted[2].charAt(0);
        /*This part is checking if the student already was added
        if they were, then the course and grade are added to the
        same student object at the index it is in the arraylist,
        instead of creating a new spot in the list for the same
        student. For this reason, the course_id and grade members
        of the Student class are arrays.*/
        if(searchForStudent(student_id, students) < 0) {
          students.add(new Student(student_id, course_id, grade));
        }
        else {
          int index = searchForStudent(student_id, students);
            students.get(index).setCourseID(course_id);
            students.get(index).setGrade(grade);
        }
    }
}

//Method that puts courses from the file into a Course type array list
public static void putCoursesInList(ArrayList<Course> courses, File c) throws FileNotFoundException {
    Scanner in = new Scanner(c);
    int count = 1;
    //In this case, just going through the file and adding the objects line by line
    while(in.hasNext()) {
        String str = in.nextLine();
        String course_id = "";
        String course_name = "";
        //Used the same method for splitting
        String[] splitted = str.split(",");
```

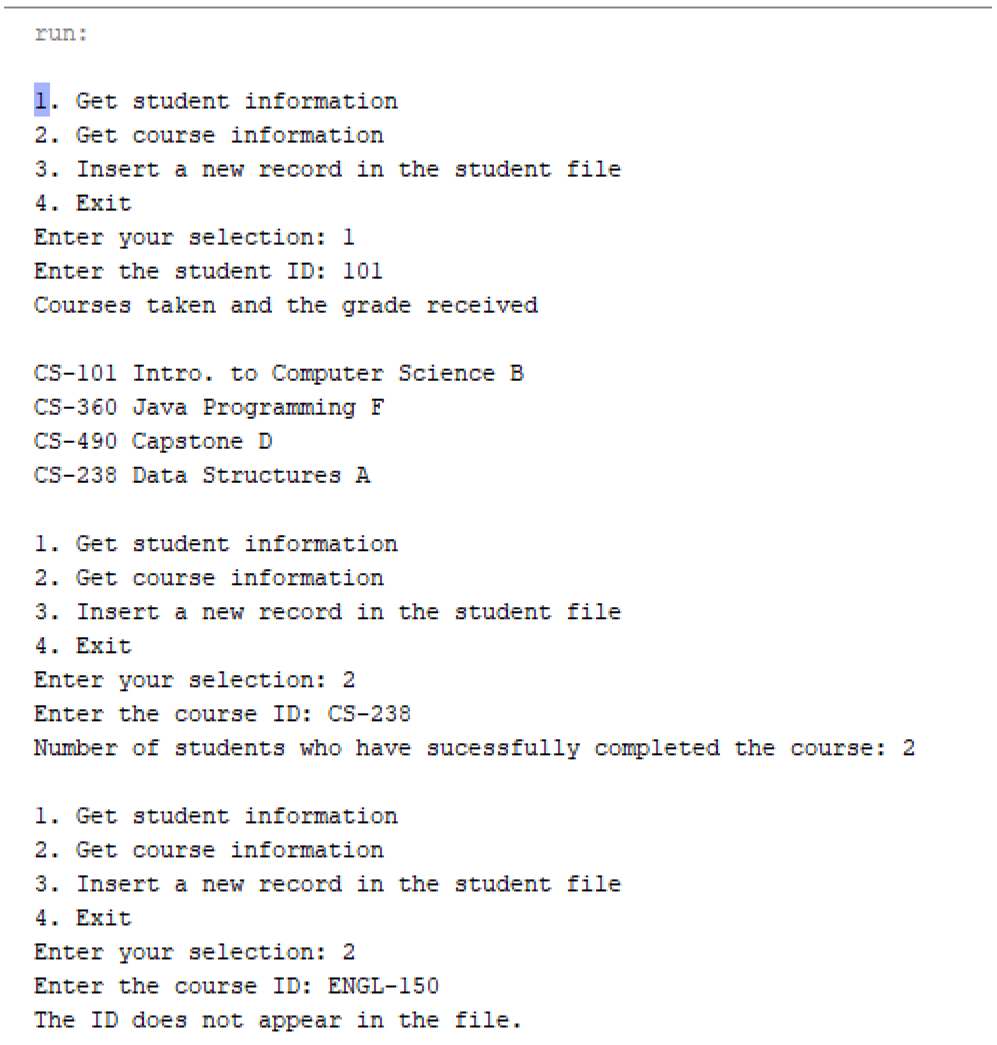
```
            course_id = splitted[0];

            course_name = splitted[1];

            courses.add(new Course(course_id, course_name));

        }

    }


    //Method that prints a certain students info, taking their id, this method calls the searchForStudent
method

    public static void getStudentInfo(String student_id, ArrayList<Student> students, ArrayList<Course>
courses) {

        int studentIndex = searchForStudent(student_id, students);

        //Checking the return of the searchForStudents method

        if(studentIndex >= 0) {

            System.out.println("Courses taken and the grade received");

            System.out.println();

            //For loop goes through all the courses the student is studying

            for(int i = 0; i < students.get(studentIndex).getNumOfCourses(); i++) {

                //Through the index of the student, searching for the index of the course, to display the name

                int courseIndex = searchForCourse(students.get(studentIndex).getCourseID(i), courses);

                System.out.println(students.get(studentIndex).getCourseID(i) + " " +
courses.get(courseIndex).getCourseName() + " " + students.get(studentIndex).getGrade(i));

            }

        }

        else

            System.out.println("The student ID does not appear in the file.");

    }


    //Method that searches for a student in a list and returns its index, if not found, returns -1

    public static int searchForStudent(String student_id, ArrayList<Student> students) {

        for (int i = 0; i < students.size(); i++) {
            if(students.get(i).getStudentID().equals(student_id))
                return i;
        }
        return -1;
    }

    //Method that displays the number of students passing the course, this method callss the courseExists 
and howManyPasssed methods
    public static void getCourseInfo(String course_id, ArrayList<Course> courses, ArrayList<Student> 
students) {
        //Checking if course exists, if it does, printing info, if not, displaying so
        if(courseExists(courses, course_id))
            System.out.println("Number of students who have sucessfully completed the course: " + 
howManyPassed(course_id, students));
        else
            System.out.println("The ID does not appear in the file.");
    }

    //Method that searches for a course in a list and returns its index, if not found, returns -1
    public static int searchForCourse(String course_id, ArrayList<Course> courses) {
        for (int i = 0; i < courses.size(); i++) {
            if(courses.get(i).getCourseID().equals(course_id))
                return i;
        }
        return -1;
    }

    //Method that returns how many people passed a course with given id number
    public static int howManyPassed(String course_id, ArrayList<Student> students) {
        int count = 0;
```

```
        for(int i = 0; i < students.size(); i++) {
            for (int j = 0; j < students.get(i).getNumOfCourses(); j++) {
                if(students.get(i).getCourseID(j).equals(course_id)) {
                    switch(students.get(i).getGrade(j)) {
                        case 'A':
                        case 'B':
                        case 'C':
                        case 'D':
                            count++;
                        default:
                            break;
                    }
                }
            }
        }
        return count;
    }

    //Method that writes student info into the file, and also adds the student to the array list, this method
calls the searchForStudent, studentExists and studentHasCourse methods
    public static void addStudent(File s, ArrayList<Student> students, ArrayList<Course> courses, String
student_id, String course_id, char grade) throws FileNotFoundException, IOException {
        //To append
        FileWriter fw = new FileWriter(s, true);
        PrintWriter write = new PrintWriter(fw);
        //Checking all the possibilities
        if(studentExists(students, student_id)) {
            if(studentHasCourse(students, course_id))
                System.out.println("The record cannot be inserted as it already exists in the student file.");
            else {
                write.print(student_id + "," + course_id + "," + grade);
                int index = searchForStudent(student_id, students);
                students.get(index).setCourseID(course_id);
                students.get(index).setGrade(grade);
                write.close();
            }
        }
        else {
            if(!courseExists(courses, course_id))
                System.out.println("The course you entered does not exist.");
            else if(!checkGrade(grade))
                System.out.println("The grade you entered is not valid.");
            else {
                write.print(student_id + "," + course_id + "," + grade);
                students.add(new Student(student_id, course_id, grade));
                write.close();
            }
        }
    }

    //Method that checks if a student exists
    public static boolean studentExists(ArrayList<Student> students, String student_id) {
        for (int i = 0; i < students.size(); i++) {
            if(students.get(i).getStudentID().equals(student_id))
                return true;
        }
        return false;
    }
```

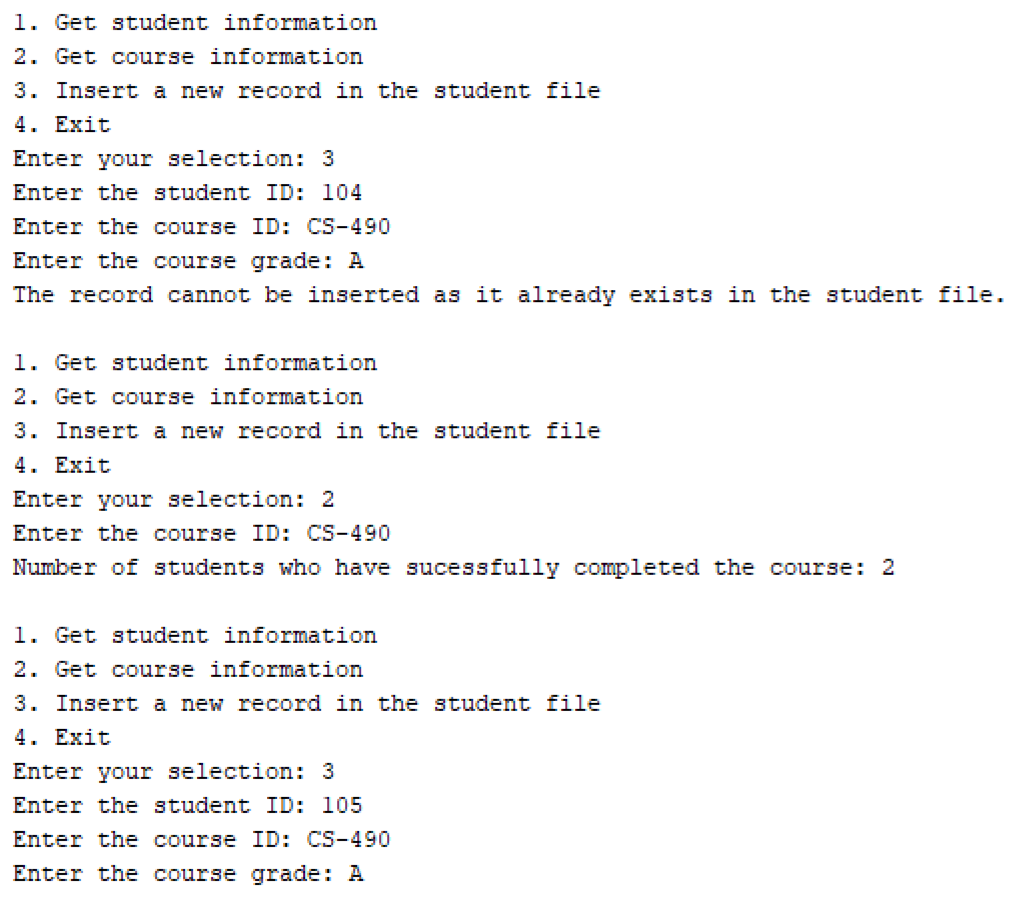
```
//Method that checks if a student has a course with given id number
public static boolean studentHasCourse(ArrayList<Student> students, String course_id) {
    for(int i = 0; i < students.size(); i++) {
        for (int j = 0; j < students.get(i).getNumOfCourses(); j++) {
            if(students.get(i).getCourseID(j).equals(course_id))
                return true;
        }
    }
    return false;
}

//Method that checks if a course exists
public static boolean courseExists(ArrayList<Course> courses, String course_id) {
    for (int i = 0; i < courses.size(); i++) {
        if(courses.get(i).getCourseID().equals(course_id))
            return true;
    }
    return false;
}

//Method checking if the given grade is valid
public static boolean checkGrade(char grade) {
    switch(grade) {
        case 'A':
        case 'B':
        case 'C':
        case 'D':
        case 'F':
            return true;
          default:
            return false;
        }
      }

    }
```

Screenshots

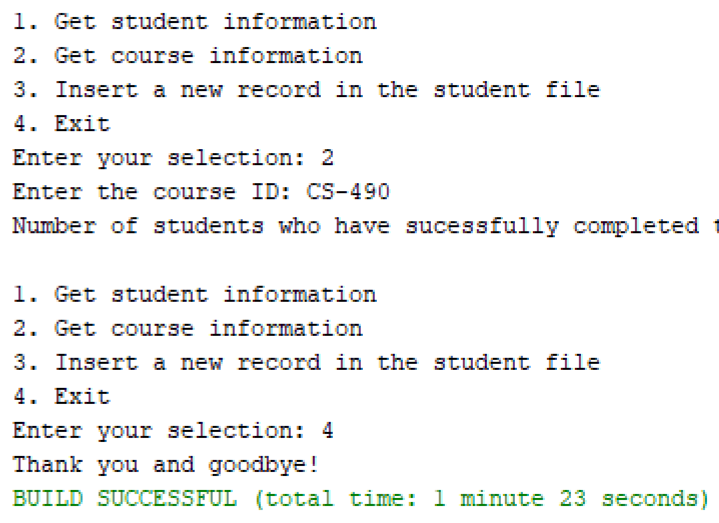
All of these screenshots are from a single run, I couldn't fit them in one picture



```
run:

1. Get student information
2. Get course information
3. Insert a new record in the student file
4. Exit
Enter your selection: 1
Enter the student ID: 101
Courses taken and the grade received

CS-101 Intro. to Computer Science B
CS-360 Java Programming F
CS-490 Capstone D
CS-238 Data Structures A

1. Get student information
2. Get course information
3. Insert a new record in the student file
4. Exit
Enter your selection: 2
Enter the course ID: CS-238
Number of students who have sucessfully completed the course: 2

1. Get student information
2. Get course information
3. Insert a new record in the student file
4. Exit
Enter your selection: 2
Enter the course ID: ENGL-150
The ID does not appear in the file.
```



```
1. Get student information
2. Get course information
3. Insert a new record in the student file
4. Exit
Enter your selection: 3
Enter the student ID: 104
Enter the course ID: CS-490
Enter the course grade: A
The record cannot be inserted as it already exists in the student file.

1. Get student information
2. Get course information
3. Insert a new record in the student file
4. Exit
Enter your selection: 2
Enter the course ID: CS-490
Number of students who have sucessfully completed the course: 2

1. Get student information
2. Get course information
3. Insert a new record in the student file
4. Exit
Enter your selection: 3
Enter the student ID: 105
Enter the course ID: CS-490
Enter the course grade: A
```



```
1. Get student information
2. Get course information
3. Insert a new record in the student file
4. Exit
Enter your selection: 2
Enter the course ID: CS-490
Number of students who have sucessfully completed the course: 3

1. Get student information
2. Get course information
3. Insert a new record in the student file
4. Exit
Enter your selection: 4
Thank you and goodbye!
BUILD SUCCESSFUL (total time: 1 minute 23 seconds)
```