

**School of Engineering and Information Technology**

**ASSIGNMENT COVER SHEET**

* Please complete and insert this form as the first page of EACH electronic assignment.
* Submit the assignment with attached coversheet electronically as per the instructions in the Assignment Question sheet.
* Please make sure you keep a copy of the assignment.

**Student Details**

|  |  |  |  |
| --- | --- | --- | --- |
| Surname | Jacobsen Teixeira | Given name | Ricardo Laner |
| Student Number | 34031229 | Email | Ricardo.jacobsen@hotmail.com |

**Assignment details**

|  |  |  |  |
| --- | --- | --- | --- |
| Unit name | Principles of computer science | Unit Code | ICT164 |
| Unit Coordinator | Kevin | Tutor/Tutorial time | Upeka |
| Due date/time | 30/10/2020 | Submission date | 30/10/2020 |
| Assignment title | Assignment2 | | |
| Other information |  | | |

Assignment2

=Coursework and Research students=

Author: Ricardo Laner Jacobsen Teixeira (34031229)

Date: 30/10/2020

Filename: Assignment2

**PURPOSE:** This program was created to help the user manipulate data stored in an external text file about the coursework and research students.

**2. REQUIREMENTS/SPECIFICATION:** The program should first display a menu and ask which option the user wishes to use. After the menu is displayed, we must know which kind of student the user wants to deal with. There are ten options in the menu where the user can search, delete, load, store in an external csv file, and sort all the students' information.

While the user does not enter the number 1 to exit the program, it will be prompted a menu offering the options again in a loop.

There are five options in which number five terminates the program.

**3.USER GUIDE:** In Netbeans, goto file/ Import project / From zip / Select the directory where you downloaded the file. Now it is imported. Press the shortcut f6 or click on  to run the program.

**4.STRUCTURE/DESIGN/ALGORITHM:**

**< Student class>**

A. Declares all the variables as private String titleOfStudent, firstName,lastName, A long studentNumber AND A int day, month, year;

B. STUDENT()-Default constructor

C. STUDENT(String ts, String fn, String ln, long sn, int day, int month, int year)-Constuctor parametrized

D. Getters and Setters( titleOfStudent, firstName, lastName, studentNumber, day, month, year) – To get and set any required variable

**< Research class>**

A. Research class extends Student Class;

B. Instanciate all the variables as private proposalComponent, finalOralPresentation, finalThesis ;

C. ResearchStudent()-Default constructor

D. ResearchStudent (int proposalComponent, int finalOralPresentation, int finalThesis)-Constuctor parametrized

E. Getters and Setters() – titleOfStudent, firstName, lastName, studentNumber, day, month, year;

F.IsEquals()- To compare two different objects.

G. ResearchStudent(int proposalComponent, int finalOralPresentation, int finalThesis, String ts, String fn, String ln, long sn,int day, int month, int year)

super(ts, fn, ln, sn, day, month, year)- Constructor parametrized with the inheritance.

H. Getters and Setters(proposalComponent, finalOralPresentation, finalThesis ) – To get and set any required variable.

I. overallMark()- Calculates the overall mark.

J. finalGrade()- Define the final grade.

K. loadResearchStundent()- Insert into the ArrayList the information.

L. deleteResearchStudent()- Delete from the array information.

M. displayAllResearchStudents()- Display all the research students and info about them.

N. AvarageResearch()- Calculates the avarege of all the students in the array.

O. findID()- Search in the array by the id given.

P. findByName()- Search in the array by the name and lastname given.

Q. sortByID() – Organize in ascendent order the id into the original ArrayList

R. toCsv()- Create a csv file and stored the information about the students.

**<CourseWorkStudent class>**

This class contain the same Research students’ structure, the only change is the class and the objects.

**<Client class>**

A. As required in the assignment specification was declared the Arraylist object type courseWorkArray and researchArray which was necessary to store the various student stances of the object.

B. As additional variables these following int was created:

Int courseworkOrResearch was necessary to control the kind of student that will be handled .

Int menu was necessary to control the Menu option.

Int ret to Receives the return form the findID and to check if was found or not.

C. Display the student information.

D. Display the menu

E. Gets the menu option.

F. Enter in the swich case in the menu

G. Operates according to the user choice

H. Ask the user option for kind of student

I. Operates according to the user choice

I. Quit when the user enters option number 1

**PSEUDUCODE**

**<CLIENT CLASS>**

*BEGIN*

public static Scanner input = new Scanner(System.in)

Declare ArrayList<CourseWorkStudent>

Declare ArrayList<ResearchStudent>

Declare integer courseworkOrResearch

Display Student Info

Declare int menu;

Lopp

Display menu

While menu>1

Ask menu option

Enter into the menu

Case 1: You QUIT, thank you for using this program. Bye

Case 2: Ask the kind of student

If enter 1:

Call method Load CourseWorkArray

End if

If enter 2:

Call method Load ResearchStudentArray

Display overall determined

End if

break

Case 3: Ask the kind of student

If enter 1:

Call method Load CourseWorkArray

Ask for ENTER THE STUDENT ID

Declare an Int ID

Receives input in ID

Call method Load CourseWorkStudent.delete(ID)

Declare an Int Position

If position =-1

Display EXCLUSION WAS ABORTED OR NOT FOUND

Else

Display THE STUDENT NOW IS DELETED

End of if

If enter 2:

Call method Load ResearchStudentArray

Declare an Int Position

If position =-1

Display EXCLUSION WAS ABORTED OR NOT FOUND

Else

Display THE STUDENT NOW IS DELETED

End of if

break

Case 4:

Call the methods loadCourseWorkStundent

loadResearchStundent

displayAllCourseWorkStudents

displayAllResearchStudents

break

Case 5: Ask the kind of student

If enter 1:

Call method Load CourseWorkArray

Loop

Display Student #

Display Student info

Display the overall and grade

End loop

End if

If enter 2:

Call method Load ResearchStudentArray

Loop

Display Student #

Display Student info

Display the overall and grade

End if

End loop

break

Case 6: Call the methods loadCourseWorkStundent

AvarageResearchCourseWork

loadResearchStundent

AvarageResearch

Break

Case 7: Declare int ret

Declare long id

Ask id;

Call the method loadCourseWorkStundent

ret Receives findID(id)

If ret equal to -1

Display ID NOT FOUND IN COURSE WORK

Ret Receives 0

End if

Call the method loadResearchStundent

Ret Receives findID(id)

If ret equal to -1

Display ID NOT FOUND IN RESEARCH

Ret Receives 0

End if

Break

Case 8: Clean the input memory

Display ENTER THE FIRST NAME

Declare String Name

Name Receives Input

Display ENTER THE LAST NAME

Declare String LastName

Name Receives Input

Call method loadCourseWorkStundent

Ret Receives findByName(name, lastName)

If ret Equas to-1

Display THE FULNAME GIVEN WAS NOT FOUND IN COURSE WORK!

Ret Receives 0

End if

Call method loadResearchStundent

Call method loadCourseWorkStundent

Ret Receives findByName(name, lastName)

If ret Equas to-1

Display THE FULNAME GIVEN WAS NOT FOUND IN COURSE WORK!

Ret Receives 0

End if

break

Case 9: Ask the kind of student

If enter 1:

Call method LoadCourseWorkArray

Call method sortByID

End if

If enter 2:

Call method Load ResearchStudentArray

Call method sortByID

End if

Break

Case 10: Ask the kind of student

If enter 1:

Call method LoadCourseWorkArray

Call method sortByID

Call method toCsv

Display CSV Created

End if

If enter 2:

Call method Load ResearchStudentArray

Call method sortByID

Call method toCsv

Display CSV Created

End if

Break

End of while

Method DisplayMenu()

Display =========MENU===========

Display 1. QUIT PROGRAM.

2. DETERMINE THE STUDENT'S OVERALL MARK AND GRADE.

3. REMOVE A STUDENT INFORMATION .

4. DISPLAY ALL THE INFORMATION ABOUT OF ALL STUDENTS.

5. COMPUTE AND DISPLARY THE OVERALL MARK AND GRADE FOR ALL STUDENTS.

6. DISPLAY HOW MANY STUDENTS HAD (EUQUAL TO OR ABOVE ) OR BELOW

THE AVARAGE.

7. DISPLAY INFORMATION BY SUTDENT ID.

8. DISPLAY INFORMATION BY NAME AND LAST NAME.

9. SORT AND DISPLAY THE STUDENT BY ID.

10. CREATE A CSV FILE FROM THE SORTED STUDENTS.

Display WHAT IS YOUR OPTION

Method StudentInfo()

Display Name: Ricardo Laner Jacobsen Teixeira

DisplayStudent Number: 34031229

Display Mode of enrolment: Internal

Display Tutor: Upeka Somaratne

Display Tutorial: Thursday 3:30 pm

*FINISH*

**<CourseWorkStudent CLASS>**

*BEGIN*

Declare integer assignment1

Declare integer assignment2

Declare integer weeklyPracticalWork

Declare integer finalExam

Default constructor CourseWorkStudent

Parametrized constructor CourseWorkStudent

Parametrized constructor with super class

getAssignment1 return assignment

SetAssignment1 receives assignment 1

getAssignment2 return assignment

SetAssignment2 receives assignment 1

getWeeklyPractical returns weeklyPracticalWork

setWeeklyPracticalWork receives weeklyPracticalWork

getFinalExam returns finalExam

steFinalExam receives finalExam

Method overallMark

Declare int overall

overall receives (assinment1\*25 + weeklyPractivalWork\*20 +finalExam\*30)/2

return overAll

Method finalGrade

Declare int fg

Fg receives this.overallMark

Declare String situation

Situation receives “”

If fg>80

Display OVERALL + GRADED AS High Distintion(HD)

Else if fg>=70 and <80

Display OVERALL + GRADED AS Distintion (D)

Else if fg>=60 and <70

Display OVERALL + the GRADED AS Credit (C)

Else if fg>=50 and <60

Display OVERALL + the GRADED AS Pass (P)

Else if fg<50

Display OVERALL + the GRADED AS (N)

return situation

Method loadCourseWorkStundent

If Client courseWorkArray isEmpty

File f receives new File (coursework.txt)

Scanner txt receives new Scanner (f)

While txt hasNext

Declare String record

Record receives txt nextLine

Define Scanner ScanRecord

ScanRecord Receives new Scanner(record)

ScanRecord uses Delimiter #

Declare String titleOfStudent, firstName, lastName

Declare long studentNumber

Declare int day,month,year,assignment1, assignment2, weeklyPracticalWork,

finalExam

titleOfStudent receives Scan Record next

firstName receives Scan Record next

lastName receives Scan Recordnext

studentNumber receives Scan Record nextLong

day receives Scan Record nextInt

month Receives ScanRecord.nextInt

year Receives ScanRecord.nextInt

assignment1 Receives ScanRecord.nextInt

assignment2 Receives ScanRecord.nextInt

weeklyPracticalWork Receives ScanRecord.nextInt

finalExam Receives ScanRecord.nextInt

CourseWorkStudent cws Receives new CourseWorkStudent(paramenter);

Client.courseWorkArray.add(cws);

End of if

End of while loop

End of loading method

Method deleteCourseWorkStundent(long id)

Declares int position Receives -1;

For loop i< Client courseWorkArray size

If Client course WorkArray StudentNumber equals to id)

position Receives i;

DisplayThe STUDENT ID + NAME + LASTNAME WAS FOUND");

Declare char confirmation receives ' '

While confirmation Different Y AND confirmation Differebt N

DisplaY DO YOU CONFIRM THE EXCLUSION? [Y/N]

confirmation Receives input to Upper Case char

ifconfirmation equals to Y

position Receives i

else

position Receives -1

break

return position

Method displayAllCourseWorkStudents

Display All the INFORMATION about COURSE WORK STUDENTS

Display +++++++++++++++

Loop for i< Client courseWorkArray size

Display STUDENT + TITLE +FIRSTNAME+ LASTNAME+ STUDENTNUMBER + DAY+MONTH

+YEAR +ASSIGNMEN1 +ASSIGNMENT2+ WEEKLYPRACTICALWORK + FINALEXAM

String situation Receives Client courseWorkArray finalGrade

Display situation

end of for loop

end of method

Method AvarageCourseWork

Declare int counter Receives 0

Declare int overall;

For i< Client courseWorkArray size

overall Receives Client courseWorkArray overallMark

if overall>=50

Counter +=1;

Display The system found FROM COURSE WORK STUDENTS +COUNTER+ STUDEND(S) which had the overall EQUAL TO OR ABOVE THE AVARAGE

counter Receives

for int i < Client courseWorkArray size

overall Receives Client courseWorkArray overallMark

if overall<50

counter +=1;

Display The system found FROM COURSE WORK STUDENTS +COUNTER+ STUDEND(S) which had the overall BELLOW THE AVARAGE.

end of method

Mehtod findID(long id )

int ret Receives 0;

for i < Client courseWorkArray size

if (Client courseWorkArray StudentNumber equas to id

Display The ID WAS FOUND in Research student.

Display STUDENT INFORMATION

Display +++++++++++++++++++++++++++++++++++++++++++++++++

break

else

ret Receives -1;

end of for

for i < Client courseWorkArray size

if Client courseWorkArray StudentNumber equals to id

Display STUDENT + TITLE +FIRSTNAME+ LASTNAME+ STUDENTNUMBER + DAY+MONTH

+YEAR +ASSIGNMEN1 +ASSIGNMENT2+ WEEKLYPRACTICALWORK + FINALEXAM

end of if

end of for

return ret

end of find ID

Method findByName

int ret Receives 0

for i < Client courseWorkArray size

String nameinarr Receives Client courseWorkArray FirstName

if nameinarr equals Ignore Case name

nameinarr Receives ClientcourseWorkArray getLastName

if nameinarr equals Ignore Case LastName

Display THE NAME WAS FOUND IN COURSE WORK

Display STUDENT INFORMATION

Display +++++++++++++++++++++++++++++++++++++++++++++

break;

else

ret Receives -1;

for i < Client coursework Array size

String nameinarr Receives Client courseWorkArray tFirstName

if nameinarr equals Ignore Case name

nameinarr Receives Client courseWorkArrayLastName

if nameinarr equals IgnoreCase lastName

Display STUDENT + TITLE +FIRSTNAME+ LASTNAME+ STUDENTNUMBER + DAY+MONTH

+YEAR +ASSIGNMEN1 +ASSIGNMENT2+ WEEKLYPRACTICALWORK + FINALEXAM

return ret;

end of findByName

Method sortByID

ArrayList<CourseWorkStudent> temp Receives new ArrayList

for < Client courseWorkArray size

for Client courseWorkArray size j > i

If ClientcourseWorkArray StudentNumber>Client courseWorkArray StudentNumber

Temp add Client.courseWorkArray

Client courseWorkArray set Client courseWorkArray

Client courseWorkArray set temp

Display The sorted Array by ID is:

for i < Client courseWorkArray size

Display STUDENT + TITLE +FIRSTNAME+ LASTNAME+ STUDENTNUMBER + DAY+MONTH

+YEAR +ASSIGNMEN1 +ASSIGNMENT2+ WEEKLYPRACTICALWORK + FINALEXAM

end of if

end of method

Method toCsv

String fileName Receives "coursework.csv"

PrintWriter outputStream Receives null;

outputStream Receives new PrintWriter fileName

for i< Client courseWorkArray size

OutputStream.println(STUDENT + TITLE +FIRSTNAME+ LASTNAME+ STUDENTNUMBER + DAY+MONTH

+YEAR +ASSIGNMEN1 +ASSIGNMENT2+ WEEKLYPRACTICALWORK + FINALEXAM

outputStream.close()

end of class

*FINISH*

**<ResearchStudent CLASS>**

*BEGIN*

Declare integer finalThesis

Declare integer proposalComponent

Declare integer finalOralPresentation

Default constructor ResearchStudent

Parametrized constructor ResearchStudent

Parametrized constructor with super class

getFinalThesis return assignment

SetFinalThesis receives assignment 1

getProposalComponent returns proposalComponent

setProposalComponent receives proposalComponent

getFinalOralPresentation returns finalOralPresentation

steFinalOralPresentation receives finalOralPresentation

Method overallMark

Declare int overall

overall receives (assinment1\*25 + weeklyPractivalWork\*20 +finalOralPresentation\*30)/2

return overAll

Method finalGrade

Declare int fg

Fg receives this.overallMark

Declare String situation

Situation receives “”

If fg>80

Display OVERALL + GRADED AS High Distintion(HD)

Else if fg>=70 and <80

Display OVERALL + GRADED AS Distintion (D)

Else if fg>=60 and <70

Display OVERALL + the GRADED AS Credit (C)

Else if fg>=50 and <60

Display OVERALL + the GRADED AS Pass (P)

Else if fg<50

Display OVERALL + the GRADED AS (N)

return situation

Method load ResearchStudent

If Client ResearchStudentArray isEmpty

File f receives new File (coursework.txt)

Scanner txt receives new Scanner (f)

While txt hasNext

Declare String record

Record receives txt nextLine

Define Scanner ScanRecord

ScanRecord Receives new Scanner(record)

ScanRecord uses Delimiter #

Declare String titleOfStudent, firstName, lastName

Declare long studentNumber

Declare int day,month,year , finalThesis, proposalComponent ,

finalOralPresentation

titleOfStudent receives Scan Record next

firstName receives Scan Record next

lastName receives Scan Recordnext

studentNumber receives Scan Record nextLong

day receives Scan Record nextInt

month Receives ScanRecord.nextInt

year Receives ScanRecord.nextInt

finalThesis Receives ScanRecord.nextInt

proposalComponent Receives ScanRecord.nextInt

finalOralPresentation Receives ScanRecord.nextInt

ResearchStudent cws Receives new ResearchStudent (paramenter);

Client.ResearchStudentArray.add(cws);

End of if

End of while loop

End of loading method

Method deleteResearchStudent (long id)

Declares int position Receives -1;

For loop i< Client ResearchStudentArray size

If Client course WorkArray StudentNumber equals to id)

position Receives i;

DisplayThe STUDENT ID + NAME + LASTNAME WAS FOUND");

Declare char confirmation receives ' '

While confirmation Different Y AND confirmation Differebt N

DisplaY DO YOU CONFIRM THE EXCLUSION? [Y/N]

confirmation Receives input to Upper Case char

ifconfirmation equals to Y

position Receives i

else

position Receives -1

break

return position

Method displayAllResearchStudent

Display All the INFORMATION about RESEARCH STUDENTS

Display +++++++++++++++

Loop for i< Client ResearchStudentArray size

Display STUDENT + TITLE +FIRSTNAME+ LASTNAME+ STUDENTNUMBER + DAY+MONTH

+YEAR +ASSIGNMEN1 +FINALTHESIS+ PROPOSALCOMPONENT + FINALORALPRESENTATION

String situation Receives Client ResearchStudentArray finalGrade

Display situation

end of for loop

end of method

Method AvarageResearch

Declare int counter Receives 0

Declare int overall;

For i< Client ResearchStudentArray size

overall Receives Client ResearchStudentArray overallMark

if overall>=50

Counter +=1;

Display The system found FROM RESEARCH STUDENTS +COUNTER+ STUDEND(S) which had the overall EQUAL TO OR ABOVE THE AVARAGE

counter Receives

for int i < Client ResearchStudentArray size

overall Receives Client ResearchStudentArray overallMark

if overall<50

counter +=1;

Display The system found FROM RESEARCH STUDENTS +COUNTER+ STUDEND(S) which had the overall BELLOW THE AVARAGE.

end of method

Mehtod findID(long id )

int ret Receives 0;

for i < Client ResearchStudentArray size

if (Client ResearchStudentArray StudentNumber equas to id

Display The ID WAS FOUND in Research student.

Display STUDENT INFORMATION

Display +++++++++++++++++++++++++++++++++++++++++++++++++

break

else

ret Receives -1;

end of for

for i < Client ResearchStudentArray size

if Client ResearchStudentArray StudentNumber equals to id

Display STUDENT + TITLE +FIRSTNAME+ LASTNAME+ STUDENTNUMBER + DAY+MONTH

+YEAR +ASSIGNMEN1 +FINALTHESIS+ PROPOSALCOMPONENT + FINALORALPRESENTATION

end of if

end of for

return ret

end of find ID

Method findByName

int ret Receives 0

for i < Client ResearchStudentArray size

String nameinarr Receives Client ResearchStudentArray FirstName

if nameinarr equals Ignore Case name

nameinarr Receives ClientResearchStudentArray getLastName

if nameinarr equals Ignore Case LastName

Display THE NAME WAS FOUND IN COURSE WORK

Display STUDENT INFORMATION

Display +++++++++++++++++++++++++++++++++++++++++++++

break;

else

ret Receives -1;

for i < Client coursework Array size

String nameinarr Receives Client ResearchStudentArray tFirstName

if nameinarr equals Ignore Case name

nameinarr Receives Client ResearchStudentArrayLastName

if nameinarr equals IgnoreCase lastName

Display STUDENT + TITLE +FIRSTNAME+ LASTNAME+ STUDENTNUMBER + DAY+MONTH

+YEAR +ASSIGNMEN1 +FINALTHESIS+ PROPOSALCOMPONENT + FINALORALPRESENTATION

return ret;

end of findByName

Method sortByID

ArrayList< ResearchStudent > temp Receives new ArrayList

for < Client ResearchStudentArray size

for Client ResearchStudentArray size j > i

If ClientResearchStudentArray StudentNumber>Client ResearchStudentArray StudentNumber

Temp add Client ResearchStudentArray

Client ResearchStudentArray set Client ResearchStudentArray

Client ResearchStudentArray set temp

Display The sorted Array by ID is:

for i < Client ResearchStudentArray size

Display STUDENT + TITLE +FIRSTNAME+ LASTNAME+ STUDENTNUMBER + DAY+MONTH

+YEAR +ASSIGNMEN1 +FINALTHESIS+ PROPOSALCOMPONENT + FINALORALPRESENTATION

end of if

end of method

Method toCsv

String fileName Receives "coursework.csv"

PrintWriter outputStream Receives null;

outputStream Receives new PrintWriter fileName

for i< Client ResearchStudentArray size

OutputStream.printlnSTUDENT + TITLE +FIRSTNAME+ LASTNAME+ STUDENTNUMBER + DAY+MONTH

+YEAR +ASSIGNMEN1 +FINALTHESIS+ PROPOSALCOMPONENT + FINALORALPRESENTATION

outputStream close

end of class

*FINISH*

**<Student CLASS>**

*BEGIN*

Declare private String titleOfStudent ,firstName, lastName, studentNumber

Declares private day, month, year

Method Student constructo

Method Student parametrized

Method getTitleOfStudent returns titleOfStudent;

Method setTitleOfStudent set titleOfStudent;

Method getFirstName return firstName;

Method setFirstName Set FirstName

Method getLastName return lastName

Method setLastName set lastName

Method getStudentNumber return studentNumber

Method setStudentNumber set studentNumber

Mehtod getDay return day

Method setDay set day

Method getMonth return month

Method setMonth set month

Method getYear return year

Method setYear set year

MethodisEqual CourseWorkStudent cws, ResearchStudent rs

Declare boolean flag equals to false

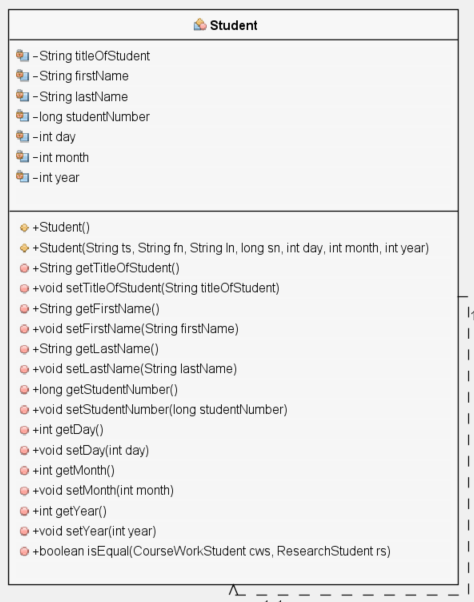
if FirstName equals Ignore Casers FirstName AND LastName equalsIgnoreCase LastName AND Day EQUALS TO Day AND Month EQUALS TO Month AND Year EQUALS TO Year

flag Receives true

return flag

*FINISH*

**<UML DIGRAM CLASS>**





****

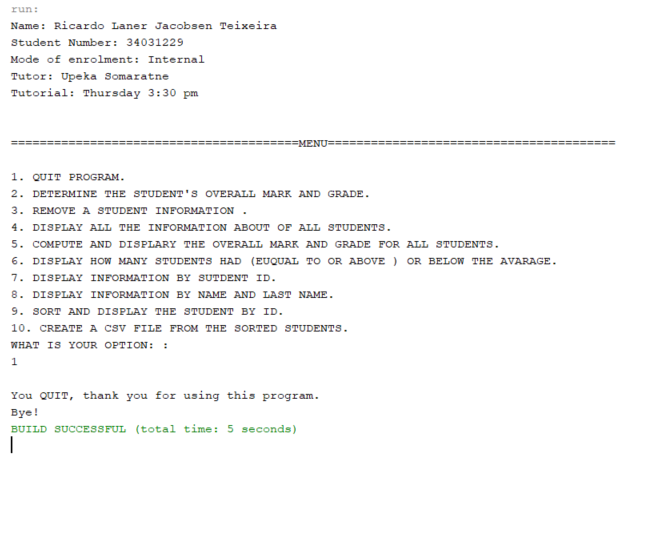
****

**5.LIMITATIONS: Based on the tests produced there are no limitations or shortfalls in the program.**

The program had some limitations about validating the input from the keyboard to the menu and for questions inside it. Then I handled it by using the if and the try and catch statement, now if the user does not type the appropriate kind of character number or string, the program will be able to ask for reentering it and not crashing.

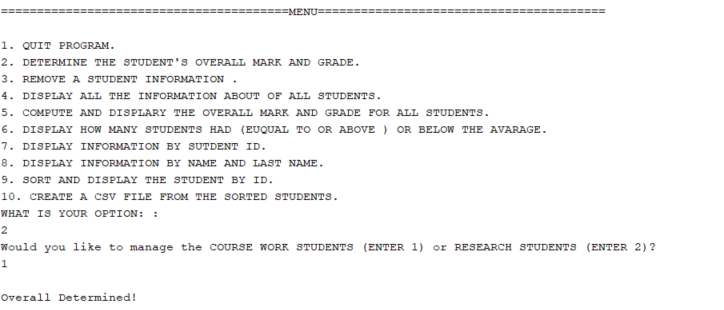
**6. TESTING / TABLE**

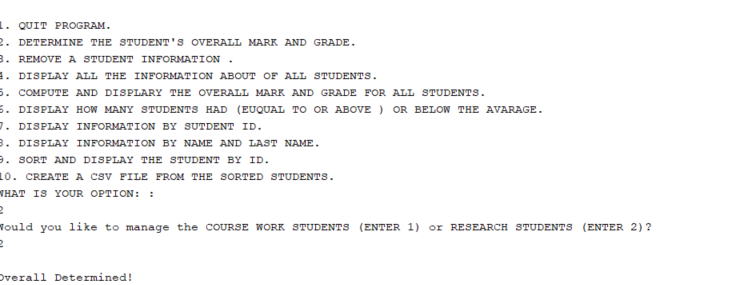
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TEST | DESCRIPTION | INPUT | EXPECTED OUTPUT | PASS/FAIL |
| 1 | MENU OPTION ENTERED <1 AND OPTION>10 | 0 OR >10 | THE MESSAGE TO ENTER THE MENU OPTION REAPPEARS | PASS |
| 2 | MENU OPTION ENTERED INAPROPRIETED | A OR ! | REDISPLAY MENU ASKING AGAIN | PASS |
| 3 | MENU OPTION 1(QUIT THE PROGRAM) | 1 | You QUIT, thank you for using this program. | PASS |
| 4 | MENU OPTION 2 | 2 | Would you like to manage the COURSE WORK STUDENTS (ENTER 1) or RESEARCH STUDENTS (ENTER 2)? | PASS |
| 5 | (SAME ERROR AND EXCEPTION HANDLING TO ALL NEXT OTHER OPTIONS) MENU OPTION 2 WHICH STUDENT? | A | This is not a number. TRY AGAIN!  Would you like to manage the COURSE WORK STUDENTS (ENTER 1) or RESEARCH STUDENTS (ENTER 2)? | PASS |
| 6 | (SAME ERROR AND EXCEPTION HANDLING TO ALL NEXT OTHER OPTIONS) MENU OPTION 2 WHICH STUDENT? | <1 OR >2 | REDISPLAY Would you like to manage the COURSE WORK STUDENTS (ENTER 1) or RESEARCH STUDENTS (ENTER 2)? | PASS |
| 7 | MENU OPTION 2 WHICH STUDENT | 1 OR 2 | Overall Determined! | PASS |
| 8 | MENU OPTION 3 | 3 | Would you like to manage the COURSE WORK STUDENTS (ENTER 1) or RESEARCH STUDENTS (ENTER 2)? | PASS |
| 9 | MENU OPTION 3 WHICH STUDENT | 1 OR 2 | ENTER THE STUDENT ID: | PASS |
| 10 | MENU OPTION 3 ENTER ID | 1 | The STUDENT ID 1 ricardo Teixeira WAS FOUND  DO YOU CONFIRM THE EXCLUSION? [Y/N] | PASS |
| 11 | DO YOU WISH TO CONFIRM THE EXCLUSION? | Y/N | Y=The STUDENT NOW IS DELETED.  N=( The EXCLUSION WAS ABORTED OR NOT  FOUND.) | PASS |
| 12 | MENU OPTION 4 | 4 | All the INFORMATION about COURSE WORK STUDENTS:…….  All the INFORMATION about RESEARCH STUDENTS:………… | PASS |
| 13 | MENU OPTION 5 | 5 | Would you like to manage the COURSE WORK STUDENTS (ENTER 1) or RESEARCH STUDENTS (ENTER 2)? | PASS |
| 14 | MENU OPTION 5 WHICH STUDENT | 1 | 1=DISPLAYED ALL INFO ABOUT COURSE WORK STUDENTS  2=DISPLAYED ALL INFO ABOUT RESEARCH  STUDENTS | PASS |
| 15 | MENU OPTION 5 WHICH STUDENT | 2 | 1=DISPLAYED ALL INFO ABOUT COURSE WORK STUDENTS  2=DISPLAYED ALL INFO ABOUT RESEARCH  STUDENTS | PASS |
| 16 | MENU OPTION 6 | 6 | The system found FROM COURSE WORK STUDENTS 9  STUDEND(S) which had the overall EQUAL TO OR ABOVE  THE AVARAGE.  The system found FROM COURSE WORK STUDENTS 0  STUDEND(S) which had the overall BELLOW  THE AVARAGE.  The system found FROM RESEARCH STUDENTS 4  STUDEND(S) which had the overall EQUAL TO OR ABOVE  THE AVARAGE.  The system found FROM RESEARCH STUDENTS 6  STUDEND(S) which had the overall BELLOW  THE AVARAGE. | PASS |
| 17 | MENU OPTION 7 | 7 | ENTER THE ID TO BE FOUND: | PASS |
| 18 | MENU OPTION 7 ENTER THE ID TO BE FOUND: | 3 | STUDENT INFORMATION  ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++  Mr Peter Pat 3 25/2/1993 ASSIGNMENT 1: 80% ASSIGNMENT 2: 90% Weekly pract. Work: 70% Final Exam: 90%  The ID WAS FOUND in Research student.  STUDENT INFORMATION  ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++  Mr John Smith 3 17/5/2025 Propoal Component: 80% Final Oral Presentation: 70% Final Thesis: 70%  ID NOT FOUND IN RESEARCH! | PASS |
| 19 | MENU OPTION 8 | 8 | ENTER THE FIRST NAME: | PASS |
| 20 | MENU OPTION 8 ENTER THE FIRST NAME: | RICARDO | ENTER THE LAST NAME: | PASS |
| 21 | MENU OPTION 8 ENTER THE LAST NAME: | TEIXEIRA | THE FULNAME GIVEN WAS NOT FOUND IN COURSE WORK!  THE FULNAME GIVEN WAS NOT FOUND IN RESEARCH! | PASS |
| 22 | MENU OPTION 8 ENTER THE FIRST NAME: | Cabela | ENTER THE LAST NAME: | PASS |
| 23 | MENU OPTION 8 ENTER THE LAST NAME: | Loko | THE NAME WAS FOUND IN COURSE WORK  STUDENT INFORMATION  ++++++++++++++++++++++++++++++++++++++++++++++  Miss Cabela Loko 26 2/2/1983 ASSIGNMENT 1: 80% ASSIGNMENT  2: 90% Weekly pract. Work: 70% Final Exam: 90%  THE FULNAME GIVEN WAS NOT FOUND IN COURSE WORK!  THE FULNAME GIVEN WAS NOT FOUND IN RESEARCH! | PASS |
| 24 | MENU OPTION 9 | 9 | Would you like to manage the COURSE WORK STUDENTS (ENTER 1) or RESEARCH STUDENTS (ENTER 2)? | PASS |
| 25 | MENU OPTION 9 WHICH STUDENT | 1 | The sorted Array by ID is:  ID: 2 Name: Sr Thiago Trodel 31/12/1993 ASSIGNMENT  1: 80% ASSIGNMENT 2: 90% Weekly pract. Work: 70%  Final Exam: 90%  ID: 3 Name: Mr Peter Pat 25/2/1993 ASSIGNMENT 1: 80%  ASSIGNMENT 2: 90% Weekly pract. Work: 70% Final Exam: 90%  ID: 6 Name: Mr Billy Crab 13/2/1993 ASSIGNMENT 1: 80%  ASSIGNMENT 2: 90% Weekly pract. Work: 70% Final Exam: 90%  And other results | PASS |
| 26 | MENU OPTION 10 | 10 | Would you like to manage the COURSE WORK STUDENTS (ENTER 1) or RESEARCH STUDENTS (ENTER 2)? | PASS |
| 27 | MENU OPTION 10 WHICH STUDENT | 1 | The sorted Array by ID is:  ID: 2 Name: Sr Thiago Trodel 31/12/1993 ASSIGNMENT  1: 80% ASSIGNMENT 2: 90% Weekly pract. Work: 70%  Final Exam: 90%  ID: 3 Name: Mr Peter Pat 25/2/1993 ASSIGNMENT 1: 80%  ASSIGNMENT 2: 90% Weekly pract. Work: 70% Final Exam: 90%  ID: 6 Name: Mr Billy Crab 13/2/1993 ASSIGNMENT 1: 80%  ASSIGNMENT 2: 90% Weekly pract. Work: 70% Final Exam: 90%  And other results  Csv created! | PASS |
| 28 | MENU OPTION 10 WHICH STUDENT | 2 | The sorted Array by ID is:  ID: 1 Name: Mr Satan DragonBall 17/10/2025 Propoal Component:  40% Final Oral Presentation: 70% Final Thesis: 70%  ID: 3 Name: Mr John Smith 17/5/2025 Propoal Component:  80% Final Oral Presentation: 70% Final Thesis: 70%  ID: 5 Name: Mr Man borrows 5/6/2020 Propoal Component:  100% Final Oral Presentation: 90% Final Thesis: 70%  Csv created! | PASS |

MENU OPTION 1 

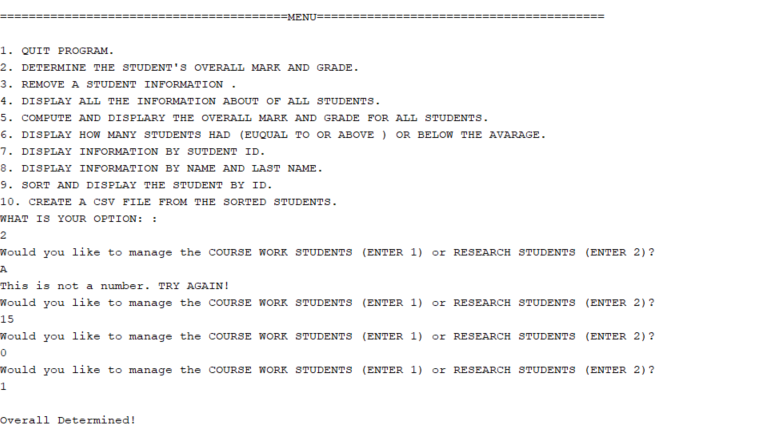
**TRYING TO CAUSE ERROR**

TEST MENU OPTION 2

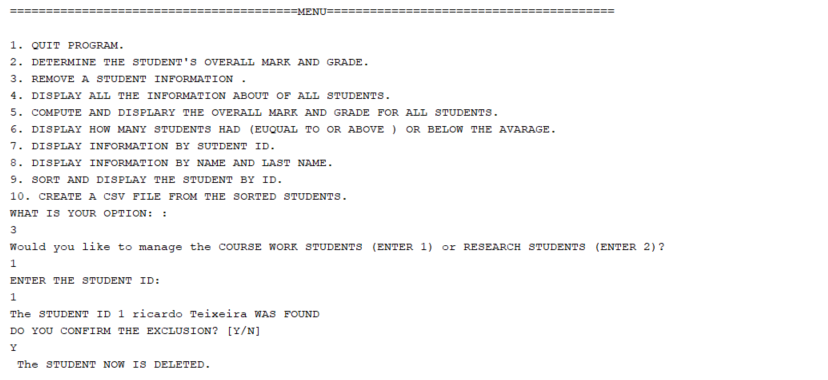


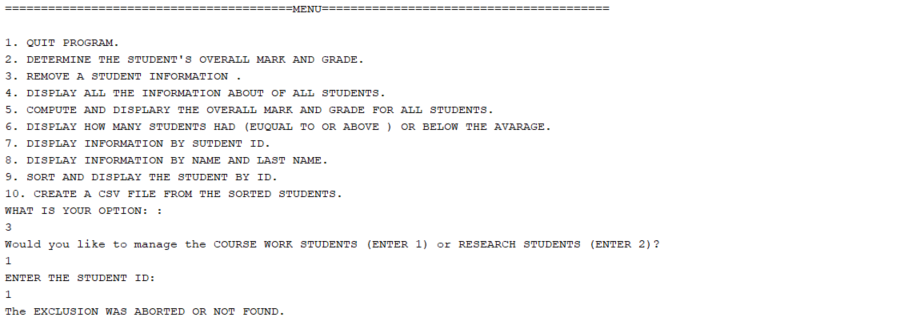


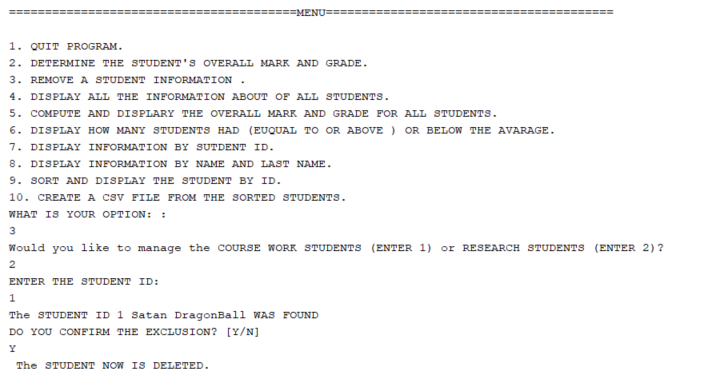
**TRYING TO CAUSE ERROR**

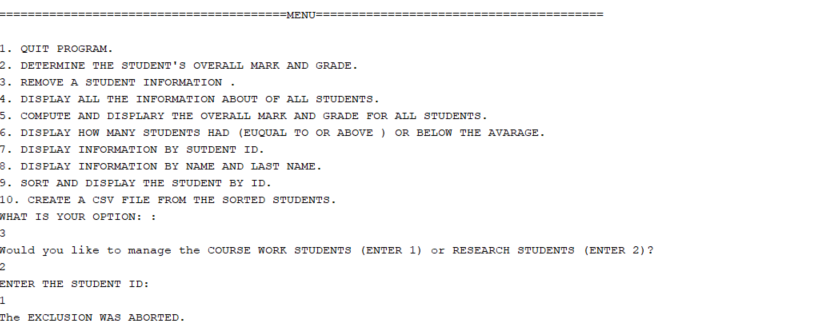


TEST MENU 3

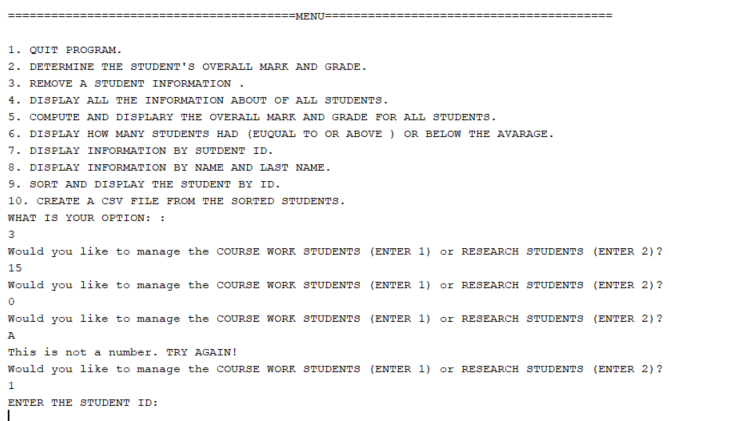








**TRYING TO CAUSE ERROR**

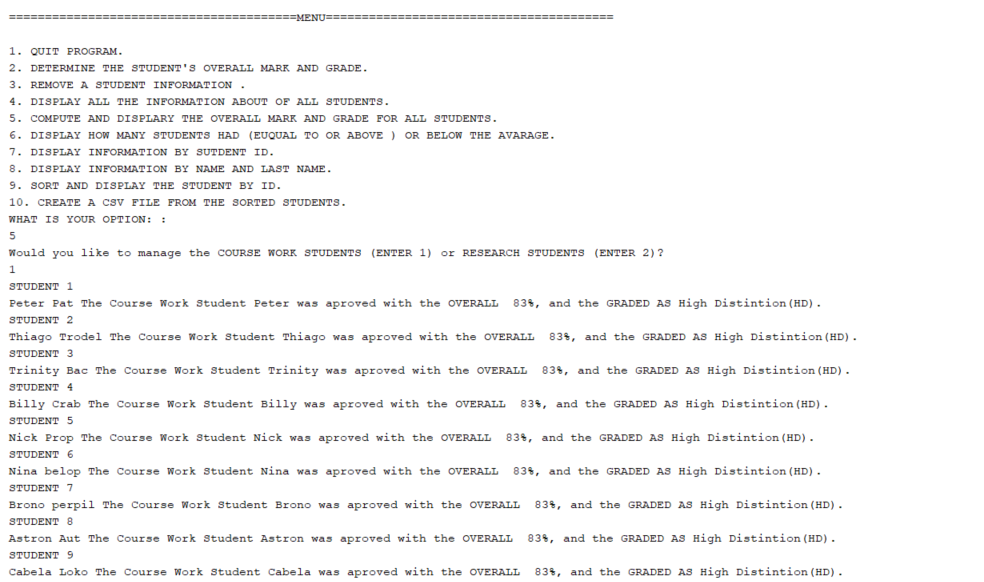


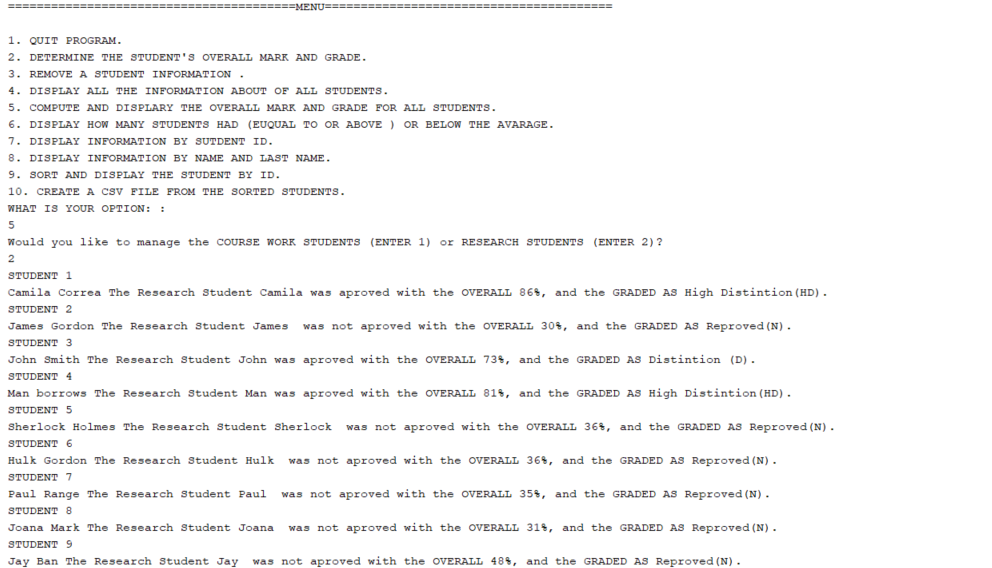
TEST MENU 4



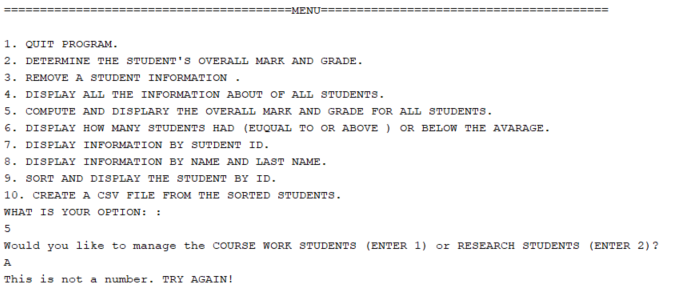


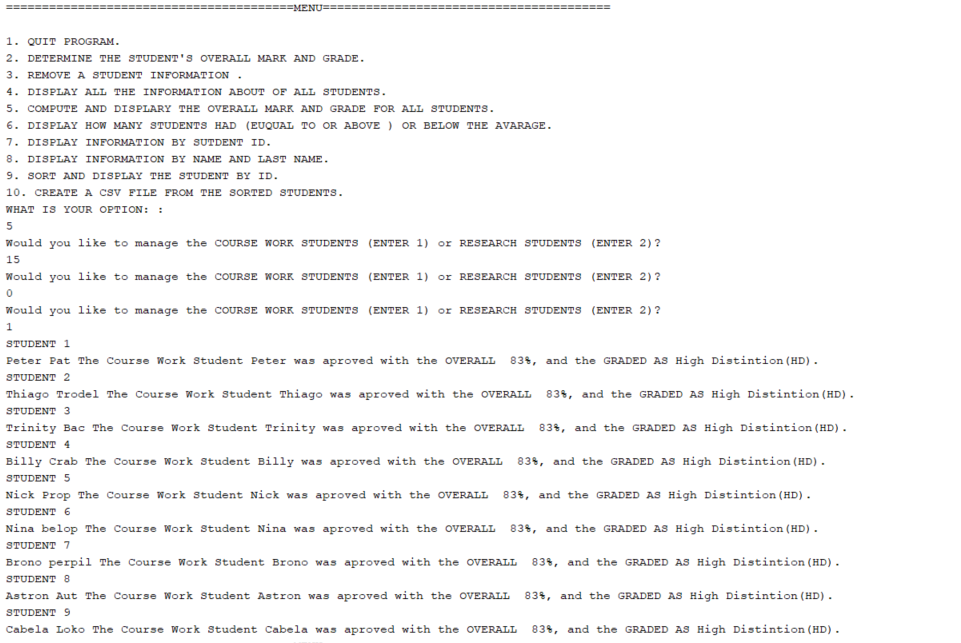
TEST MENU 5



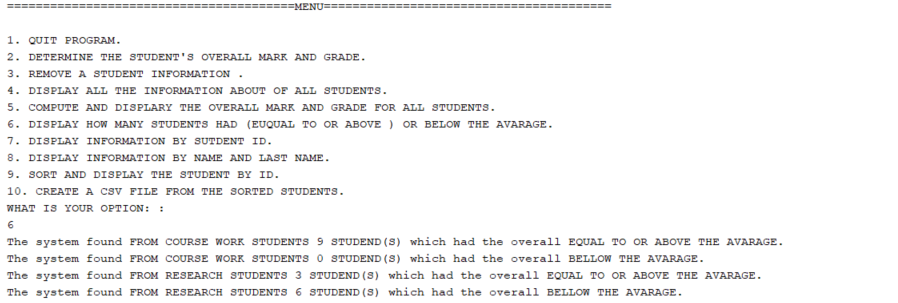


**TRYING TO CAUSE ERROR**

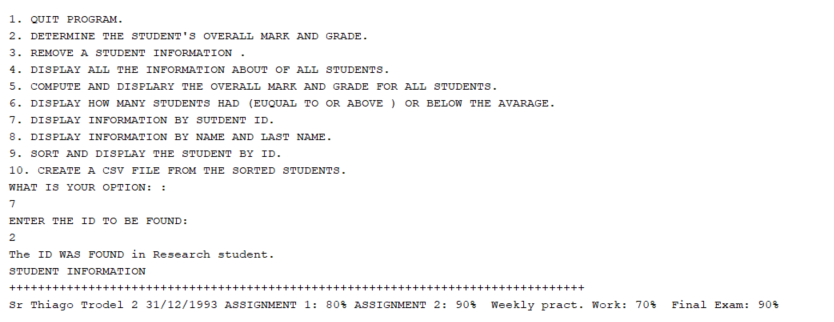




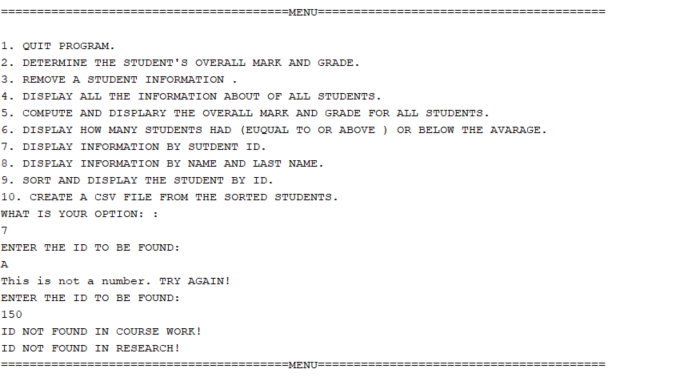
TEST 6



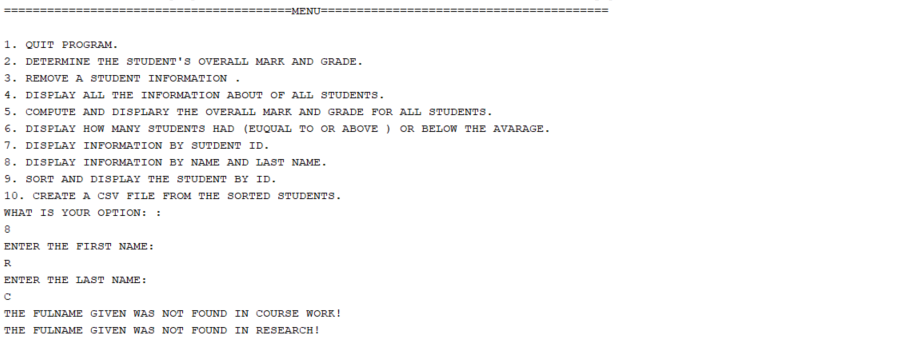
TEST 7

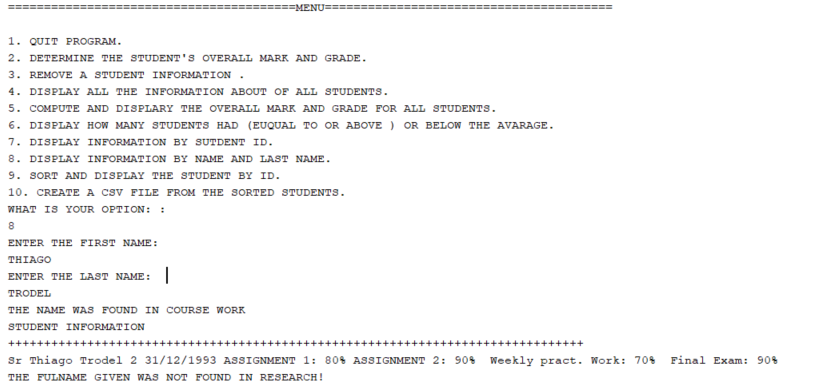


**TRYING TO CAUSE ERROR**

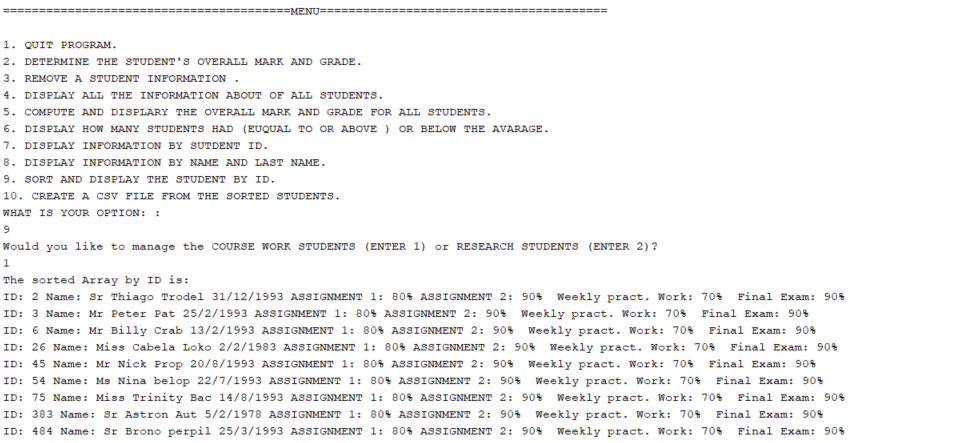


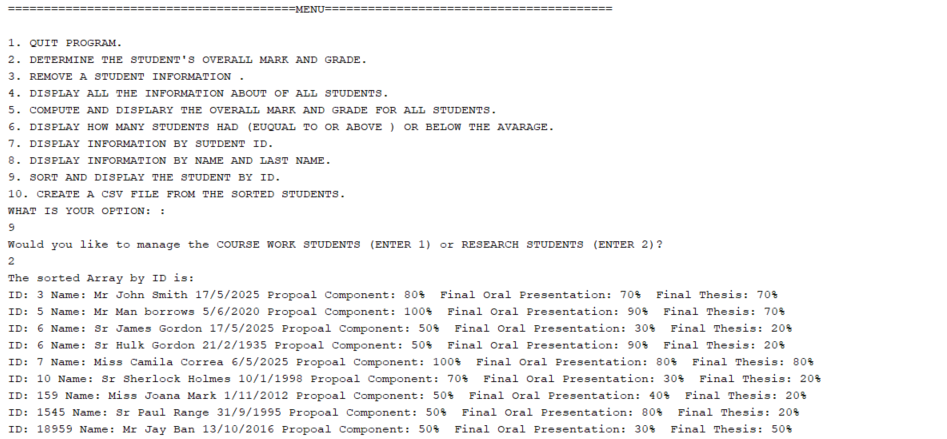
TEST MENU 8



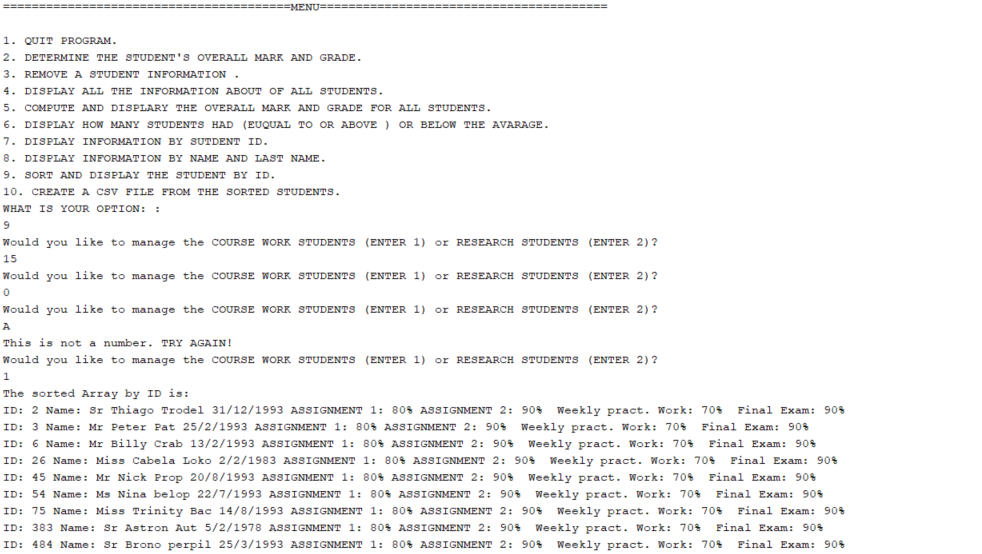


TEST MENU 9

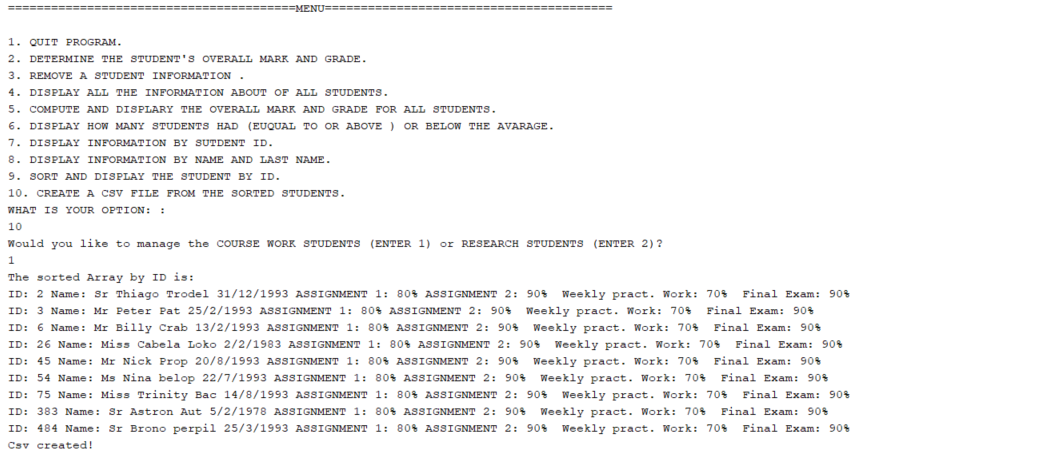


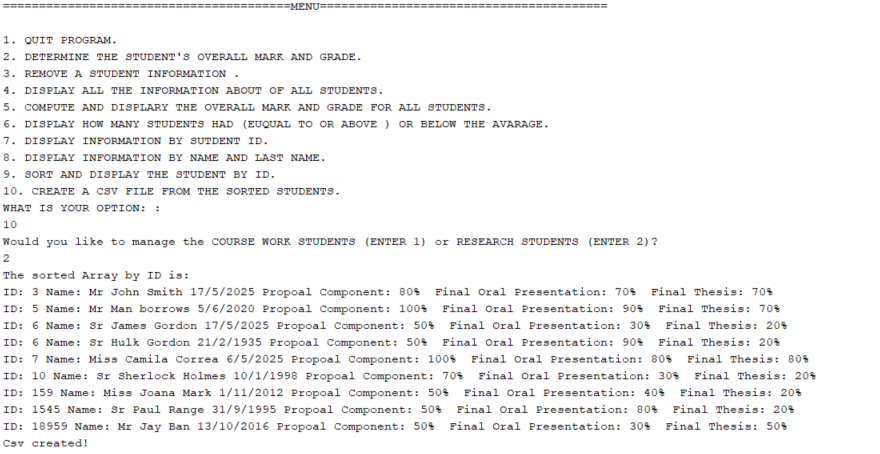


**TRYING TO CAUSE ERROR**

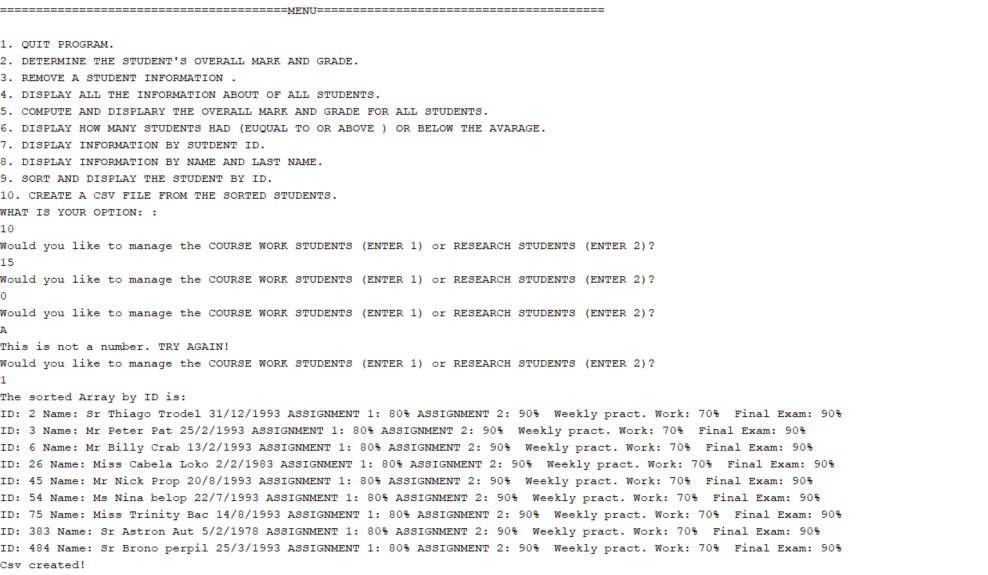


TEST MENU 10



****

**TRYING TO CAUSE ERROR**

****

**4.SOURCE PROGRAM LISTING:**

**<CLIENT CLASS>**

/\*\*

\* Title: Client Change

\* @author Ricardo Laner Jacobsen Teixeira

\* Date: 30/10/2020

\* File name: Assignment 2

\* Purpose: This program was created with the intuition of helping the user on manage data about work course and research students.

\*

\*/

package assignement2;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.IOException;

import java.io.PrintWriter;

import java.util.ArrayList;

import java.util.InputMismatchException;

import java.util.Scanner;

public class Client {

public static Scanner input = new Scanner(System.in);

public static ArrayList<CourseWorkStudent> courseWorkArray = new ArrayList();

public static ArrayList<ResearchStudent> researchArray = new ArrayList();

public static void main(String[] args) throws IOException {

int courseworkOrResearch = 0;

StudentInfo();

int menu;

do {

displayMenu();

try {

menu = input.nextInt();

} catch (InputMismatchException e) {

menu = 0;

input.next();

}

switch (menu) {

case 1:

System.out.println("\nYou QUIT, thank you for using this program.\nBye!");

System.exit(0);

break;

case 2:

do {

System.out.println("Would you like to manage the COURSE WORK STUDENTS (ENTER 1) or RESEARCH STUDENTS (ENTER 2)? ");

try {

courseworkOrResearch = input.nextInt();

if (courseworkOrResearch == 1) {

CourseWorkStudent.loadCourseWorkStundent();

System.out.println("\nOverall Determined!");

} else if (courseworkOrResearch == 2) {

ResearchStudent.loadResearchStundent();

System.out.println("\nOverall Determined!");

}

} catch (InputMismatchException e) {

System.out.println("This is not a number. TRY AGAIN!");

menu = 0;

input.next();

}

} while (courseworkOrResearch != 1 && courseworkOrResearch != 2);

break;

case 3:

do {

System.out.println("Would you like to manage the COURSE WORK STUDENTS (ENTER 1) or RESEARCH STUDENTS (ENTER 2)? ");

try {

courseworkOrResearch = input.nextInt();

if (courseworkOrResearch == 1) {

CourseWorkStudent.loadCourseWorkStundent();

System.out.println("ENTER THE STUDENT ID: ");

long id = input.nextLong();

int position = CourseWorkStudent.deleteCourseWorkStundent(id);

if (position == -1) {

System.out.println("The EXCLUSION WAS ABORTED OR NOT FOUND.\n");

} else {

courseWorkArray.remove(position);

System.out.println(" The STUDENT NOW IS DELETED.\n");

}

} else if (courseworkOrResearch == 2) {

ResearchStudent.loadResearchStundent();

System.out.println("ENTER THE STUDENT ID: ");

long id = input.nextLong();

int position = ResearchStudent.deleteResearchStudent(id);

if (position == -1) {

System.out.println("The EXCLUSION WAS ABORTED.\n");

} else {

researchArray.remove(position);

System.out.println(" The STUDENT NOW IS DELETED.\n");

}

}

} catch (InputMismatchException e) {

System.out.println("This is not a number. TRY AGAIN!");

menu = 0;

input.next();

}

} while (courseworkOrResearch != 1 && courseworkOrResearch != 2);

break;

case 4:

CourseWorkStudent.loadCourseWorkStundent();

ResearchStudent.loadResearchStundent();

CourseWorkStudent.displayAllCourseWorkStudents();

ResearchStudent.displayAllResearchStudents();

break;

case 5:

do {

System.out.println("Would you like to manage the COURSE WORK STUDENTS (ENTER 1) or RESEARCH STUDENTS (ENTER 2)? ");

try {

courseworkOrResearch = input.nextInt();

if (courseworkOrResearch == 1) {

CourseWorkStudent.loadCourseWorkStundent();

for (int i = 0; i < courseWorkArray.size(); i++) {

System.out.println("STUDENT " + (i + 1));

System.out.println(courseWorkArray.get(i).getFirstName() + " " + courseWorkArray.get(i).getLastName() + " " + courseWorkArray.get(i).finalGrade());

}

} else if (courseworkOrResearch == 2) {

ResearchStudent.loadResearchStundent();

for (int i = 0; i < researchArray.size(); i++) {

System.out.println("STUDENT " + (i + 1));

System.out.println(researchArray.get(i).getFirstName() + " " + researchArray.get(i).getLastName() + " " + researchArray.get(i).finalGrade());

}

}

} catch (InputMismatchException e) {

System.out.println("This is not a number. TRY AGAIN!");

menu = 0;

input.next();

}

} while (courseworkOrResearch != 1 && courseworkOrResearch != 2);

break;

case 6:

CourseWorkStudent.loadCourseWorkStundent();

CourseWorkStudent.AvarageCourseWork();

ResearchStudent.loadResearchStundent();

ResearchStudent.AvarageResearch();

break;

case 7:

int ret = 0;

do {

System.out.println("ENTER THE ID TO BE FOUND: ");

try {

long id = input.nextLong();

CourseWorkStudent.loadCourseWorkStundent();

ret = CourseWorkStudent.findID(id);

if (ret == -1) {

System.out.println("ID NOT FOUND IN COURSE WORK!");

}

ret = 0;// cleaning the return

ResearchStudent.loadResearchStundent();

ret = ResearchStudent.findID(id);

if (ret == -1) {

System.out.println("ID NOT FOUND IN RESEARCH!");

}

ret = 0;// cleaning the return

break;

} catch (InputMismatchException e) {

System.out.println("This is not a number. TRY AGAIN!");

menu = 0;

input.next();

}

} while (true);

break;

case 8:

input.nextLine();//cleaning the input

do {

try {

System.out.println("ENTER THE FIRST NAME: ");

String name = input.nextLine();

System.out.println("ENTER THE LAST NAME: ");

String lastName = input.nextLine();

CourseWorkStudent.loadCourseWorkStundent();

ret = CourseWorkStudent.findByName(name, lastName);

if (ret == -1) {

System.out.println("THE FULNAME GIVEN WAS NOT FOUND IN COURSE WORK!");

}

ret = 0;// cleaning the return

ResearchStudent.loadResearchStundent();

ret = ResearchStudent.findByName(name, lastName);

if (ret == -1) {

System.out.println("THE FULNAME GIVEN WAS NOT FOUND IN RESEARCH!");

}

ret = 0;// cleaning the return

break;

} catch (InputMismatchException e) {

System.out.println("This is not a number. TRY AGAIN!");

menu = 0;

input.next();

}

} while (true);

break;

case 9:

do {

System.out.println("Would you like to manage the COURSE WORK STUDENTS (ENTER 1) or RESEARCH STUDENTS (ENTER 2)? ");

try {

courseworkOrResearch = input.nextInt();

if (courseworkOrResearch == 1) {

CourseWorkStudent.loadCourseWorkStundent();

CourseWorkStudent.sortByID();

} else if (courseworkOrResearch == 2) {

ResearchStudent.loadResearchStundent();

ResearchStudent.sortByID();

}

} catch (InputMismatchException e) {

System.out.println("This is not a number. TRY AGAIN!");

menu = 0;

input.next();

}

} while (courseworkOrResearch != 1 && courseworkOrResearch != 2);

break;

case 10:

do {

System.out.println("Would you like to manage the COURSE WORK STUDENTS (ENTER 1) or RESEARCH STUDENTS (ENTER 2)? ");

try {

courseworkOrResearch = input.nextInt();

if (courseworkOrResearch == 1) {

CourseWorkStudent.loadCourseWorkStundent();

CourseWorkStudent.sortByID();

CourseWorkStudent.toCsv();

System.out.println("Csv created!");

} else if (courseworkOrResearch == 2) {

ResearchStudent.loadResearchStundent();

ResearchStudent.sortByID();

ResearchStudent.toCsv();

System.out.println("Csv created!");

}

} catch (InputMismatchException e) {

System.out.println("This is not a number. TRY AGAIN!");

menu = 0;

input.next();

}

} while (courseworkOrResearch != 1 && courseworkOrResearch != 2);

}//End of Switch

} while (true);

}

public static void displayMenu() {

System.out.println("========================================MENU========================================");

System.out.println(" ");

System.out.println("1. QUIT PROGRAM.\n"

+ "2. DETERMINE THE STUDENT'S OVERALL MARK AND GRADE.\n"

+ "3. REMOVE A STUDENT INFORMATION .\n"

+ "4. DISPLAY ALL THE INFORMATION ABOUT OF ALL STUDENTS.\n"

+ "5. COMPUTE AND DISPLARY THE OVERALL MARK AND GRADE FOR ALL STUDENTS.\n"

+ "6. DISPLAY HOW MANY STUDENTS HAD (EUQUAL TO OR ABOVE ) OR BELOW THE AVARAGE.\n"

+ "7. DISPLAY INFORMATION BY SUTDENT ID.\n"

+ "8. DISPLAY INFORMATION BY NAME AND LAST NAME.\n"

+ "9. SORT AND DISPLAY THE STUDENT BY ID.\n"

+ "10. CREATE A CSV FILE FROM THE SORTED STUDENTS. ");

System.out.println("WHAT IS YOUR OPTION: : ");

}

public static void StudentInfo() {

System.out.println("Name: Ricardo Laner Jacobsen Teixeira");

System.out.println("Student Number: 34031229");

System.out.println("Mode of enrolment: Internal ");

System.out.println("Tutor: Upeka Somaratne");

System.out.println("Tutorial: Thursday 3:30 pm\n\n");

}

}

**<STUDENT CLASS>**

/\*\*

\* Title: Client Change

\* @author Ricardo Laner Jacobsen Teixeira

\* Date: 30/10/2020

\* File name: Assignment 2

\* Purpose: This program was created with the intuition of helping the user on manage data about work course and research students.

\*

\*/

package assignement2;

import java.io.Serializable;

public class Student {

private String titleOfStudent;

private String firstName;

private String lastName;

private long studentNumber;

private int day;

private int month;

private int year;

public Student(){

titleOfStudent= this.titleOfStudent;

firstName = this.firstName;

lastName = this.lastName;

studentNumber=this.studentNumber;

day= this.day;

month= this.month;

year= this.year;

}

public Student(String ts, String fn, String ln, long sn, int day, int month, int year){

this. titleOfStudent= ts;

this.firstName = fn;

this.lastName = ln;

this.studentNumber=sn;

this.day= day;

this.month= month;

this.year= year;

}

public String getTitleOfStudent() {

return titleOfStudent;

}

public void setTitleOfStudent(String titleOfStudent) {

this.titleOfStudent = titleOfStudent;

}

public String getFirstName() {

return firstName;

}

public void setFirstName(String firstName) {

this.firstName = firstName;

}

public String getLastName() {

return lastName;

}

public void setLastName(String lastName) {

this.lastName = lastName;

}

public long getStudentNumber() {

return studentNumber;

}

public void setStudentNumber(long studentNumber) {

this.studentNumber = studentNumber;

}

public int getDay() {

return day;

}

public void setDay(int day) {

this.day = day;

}

public int getMonth() {

return month;

}

public void setMonth(int month) {

this.month = month;

}

public int getYear() {

return year;

}

public void setYear(int year) {

this.year = year;

}

public boolean isEqual (CourseWorkStudent cws, ResearchStudent rs){

boolean flag= false;

if((cws.getFirstName().equalsIgnoreCase(rs.getFirstName())) && (cws.getLastName().equalsIgnoreCase(rs.getLastName())) && (cws.getDay()==rs.getDay()) && (cws.getMonth()==rs.getMonth()) && (cws.getYear())==rs.getYear()){

flag = true;

}

return flag;

}

}

**<COURSEWORKSTUDENT CLASS>**

/\*\*

\* Title: Client Change

\* @author Ricardo Laner Jacobsen Teixeira

\* Date: 30/10/2020

\* File name: Assignment 2

\* Purpose: This program was created with the intuition of helping the user on manage data about work course and research students.

\*

\*/

package assignement2;

import static assignement2.Client.courseWorkArray;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.PrintWriter;

import java.util.ArrayList;

import java.util.Scanner;

public class CourseWorkStudent extends Student {

public static Scanner input = new Scanner (System.in);

//declaring the variables

private int assignment1;

private int assignment2;

private int weeklyPracticalWork;

private int finalExam;

// Default constructor

public CourseWorkStudent() {

this.assignment1= assignment1;

this.assignment2= assignment2;

this.weeklyPracticalWork= weeklyPracticalWork;

this.finalExam= finalExam;

}

//parametrized constructor

public CourseWorkStudent(int assignment1, int assignment2, int weeklyPracticalWork, int finalExam) {

this.assignment1 = assignment1;

this.assignment2 = assignment2;

this.weeklyPracticalWork = weeklyPracticalWork;

this.finalExam = finalExam;

}

//parametrized constructor with the super class attributes

public CourseWorkStudent(int assignment1, int assignment2, int weeklyPracticalWork, int finalExam, String ts, String fn, String ln, long sn, int day, int month, int year) {

super(ts, fn, ln, sn, day, month, year);

this.assignment1 = assignment1;

this.assignment2 = assignment2;

this.weeklyPracticalWork = weeklyPracticalWork;

this.finalExam = finalExam;

}

public int getAssignment1() {

return assignment1;

}

public void setAssignment1(int assignment1) {

this.assignment1 = assignment1;

}

public int getAssignment2() {

return assignment2;

}

public void setAssignment2(int assignment2) {

this.assignment2 = assignment2;

}

public int getWeeklyPracticalWork() {

return weeklyPracticalWork;

}

public void setWeeklyPracticalWork(int weeklyPracticalWork) {

this.weeklyPracticalWork = weeklyPracticalWork;

}

public int getFinalExam() {

return finalExam;

}

public void setFinalExam(int finalExam) {

this.finalExam = finalExam;

}

public int overallMark(){

int overall= (((this.assignment1\*25)+(this.assignment2\*25)+(this.weeklyPracticalWork\*20)+(this.finalExam\*30))/100);

return overall;

}

public String finalGrade(){

int fg = this.overallMark();

String situation =" ";

if(fg>=80){situation ="The Course Work Student " +getFirstName()+" was aproved with the OVERALL "+overallMark()+ "%, and the GRADED AS High Distintion(HD)."; }

else if (fg>=70 && fg<80){situation="The Course Work Student " +getFirstName()+" was aproved with the OVERALL "+overallMark()+ "%, and the GRADED AS Distintion (D)."; }

else if (fg>=60 && fg<70){situation="The Course Work Student " +getFirstName()+" was aproved with the OVERALL "+overallMark()+ "%, and the GRADED AS Credit (C)."; }

else if (fg>=50 && fg<60){situation="The Course Work Student " +getFirstName()+" was aproved with the OVERALL "+overallMark()+ "%, and the GRADED AS Pass (P)."; }

else if (fg<50){situation="The Course Work Student " +getFirstName()+" was not aproved with " + overallMark() + "%, and the GRADED AS (N)."; }

return situation;

}

public static void loadCourseWorkStundent() throws FileNotFoundException{

if(Client.courseWorkArray.isEmpty()){

File f = new File ("coursework.txt");

Scanner txt = new Scanner (f);

while(txt.hasNext()){

String record = txt.nextLine();

Scanner ScanRecord = new Scanner(record);

ScanRecord.useDelimiter("#");

String titleOfStudent, firstName, lastName;

long studentNumber;

int day,month,year,assignment1, assignment2, weeklyPracticalWork, finalExam;

// needed to be recreated because of scan order;

titleOfStudent = ScanRecord.next();

firstName = ScanRecord.next();

lastName = ScanRecord.next();

studentNumber = ScanRecord.nextLong();

day = ScanRecord.nextInt();

month = ScanRecord.nextInt();

year = ScanRecord.nextInt();

assignment1=ScanRecord.nextInt();

assignment2=ScanRecord.nextInt();

weeklyPracticalWork= ScanRecord.nextInt();

finalExam= ScanRecord.nextInt();

CourseWorkStudent cws = new CourseWorkStudent(assignment1,assignment2,weeklyPracticalWork,finalExam,titleOfStudent,firstName,lastName,studentNumber,day,month,year);

Client.courseWorkArray.add(cws);

} //end of if

}//end of while loop

}//end of loading method

public static int deleteCourseWorkStundent(long id){

int position = -1;

for (int i = 0; i < Client.courseWorkArray.size(); i++) {

if(Client.courseWorkArray.get(i).getStudentNumber()==id){

position = i;

System.out.println("The STUDENT ID "+id+" "+ Client.courseWorkArray.get(i).getFirstName()+" "+Client.courseWorkArray.get(i).getLastName()+" WAS FOUND");

char confirmation= ' ' ;

while(confirmation !='Y' && confirmation !='N' ){

System.out.println("DO YOU CONFIRM THE EXCLUSION? [Y/N]");

confirmation = input.next().toUpperCase().charAt(0); }

if(confirmation=='Y'){

position = i;

}else {

position = -1;

}

break;

}

}

return position;

}

public static void displayAllCourseWorkStudents(){

System.out.println("All the INFORMATION about COURSE WORK STUDENTS:");

System.out.println("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++");

for (int i = 0; i < Client.courseWorkArray.size(); i++) {

System.out.println("STUDENT "+(i+1)+": ");

System.out.println(Client.courseWorkArray.get(i).getTitleOfStudent()+" "+

Client.courseWorkArray.get(i).getFirstName()+" "+

Client.courseWorkArray.get(i).getLastName()+" "+

Client.courseWorkArray.get(i).getStudentNumber()+" "+

Client.courseWorkArray.get(i).getDay()+"/"+

Client.courseWorkArray.get(i).getMonth()+"/"+

Client.courseWorkArray.get(i).getYear()+" "+

"ASSIGNMENT 1: "+Client.courseWorkArray.get(i).getAssignment1()+"% "+

"ASSIGNMENT 2: "+Client.courseWorkArray.get(i).getAssignment2()+"% "+

" Weekly pract. Work: "+Client.courseWorkArray.get(i).getWeeklyPracticalWork()+"% "+

" Final Exam: "+Client.courseWorkArray.get(i).getFinalExam()+"%");

String situation= Client.courseWorkArray.get(i).finalGrade();

System.out.println(situation+"\n");

}//end of for loop

}//end of method

public static void AvarageCourseWork(){

int counter =0;

int overall;

for (int i = 0; i < Client.courseWorkArray.size(); i++) {

overall=Client.courseWorkArray.get(i).overallMark();

if (overall>=50) {

counter +=1;

}

}

System.out.println("The system found FROM COURSE WORK STUDENTS "+counter+" STUDEND(S) which had the overall EQUAL TO OR ABOVE THE AVARAGE.");

counter =0;//cleaning for the next loop

for (int i = 0; i < Client.courseWorkArray.size(); i++) {

overall=Client.courseWorkArray.get(i).overallMark();

if (overall<50) {

counter +=1;

}

}

System.out.println("The system found FROM COURSE WORK STUDENTS "+counter+" STUDEND(S) which had the overall BELLOW THE AVARAGE.");

}//end of method

public static int findID(long id ){

int ret=0;

for (int i = 0; i < Client.courseWorkArray.size(); i++) {

if (Client.courseWorkArray.get(i).getStudentNumber()==id) {

System.out.println("The ID WAS FOUND in Research student.");

System.out.println("STUDENT INFORMATION");

System.out.println("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++");

break;

} else{

ret=-1;

}

}//end of for

for (int i = 0; i < Client.courseWorkArray.size(); i++) {

if (Client.courseWorkArray.get(i).getStudentNumber()==id) {

System.out.println(Client.courseWorkArray.get(i).getTitleOfStudent()+" "+

Client.courseWorkArray.get(i).getFirstName()+" "+

Client.courseWorkArray.get(i).getLastName()+" "+

Client.courseWorkArray.get(i).getStudentNumber()+" "+

Client.courseWorkArray.get(i).getDay()+"/"+

Client.courseWorkArray.get(i).getMonth()+"/"+

Client.courseWorkArray.get(i).getYear()+" "+

"ASSIGNMENT 1: "+Client.courseWorkArray.get(i).getAssignment1()+"% "+

"ASSIGNMENT 2: "+Client.courseWorkArray.get(i).getAssignment2()+"% "+

" Weekly pract. Work: "+Client.courseWorkArray.get(i).getWeeklyPracticalWork()+"% "+

" Final Exam: "+Client.courseWorkArray.get(i).getFinalExam()+"%\n");

}//end of if

}//end of for

return ret;

}//end of find ID

public static int findByName(String name , String lastName){

int ret=0;

for (int i = 0; i < Client.courseWorkArray.size(); i++) {

//to receive the name from the arrray

String nameinarr =Client.courseWorkArray.get(i).getFirstName();

if (nameinarr.equalsIgnoreCase(name)) {

nameinarr=Client.courseWorkArray.get(i).getLastName();

if (nameinarr.equalsIgnoreCase(lastName)) {

System.out.println("THE NAME WAS FOUND IN COURSE WORK ");

System.out.println("STUDENT INFORMATION");

System.out.println("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++");

break;

}

}else {

ret=-1;

}

}

for (int i = 0; i < Client.courseWorkArray.size(); i++) {

//to receive the name from the arrray

String nameinarr =Client.courseWorkArray.get(i).getFirstName();

if (nameinarr.equalsIgnoreCase(name)) {

nameinarr=Client.courseWorkArray.get(i).getLastName();

if (nameinarr.equalsIgnoreCase(lastName)) {

System.out.println(Client.courseWorkArray.get(i).getTitleOfStudent()+" "+

Client.courseWorkArray.get(i).getFirstName()+" "+

Client.courseWorkArray.get(i).getLastName()+" "+

Client.courseWorkArray.get(i).getStudentNumber()+" "+

Client.courseWorkArray.get(i).getDay()+"/"+

Client.courseWorkArray.get(i).getMonth()+"/"+

Client.courseWorkArray.get(i).getYear()+" "+

"ASSIGNMENT 1: "+Client.courseWorkArray.get(i).getAssignment1()+"% "+

"ASSIGNMENT 2: "+Client.courseWorkArray.get(i).getAssignment2()+"% "+

" Weekly pract. Work: "+Client.courseWorkArray.get(i).getWeeklyPracticalWork()+"% "+

" Final Exam: "+Client.courseWorkArray.get(i).getFinalExam()+"%");

}

}

}

return ret;

}//end of findByName

public static void sortByID(){

ArrayList<CourseWorkStudent> temp = new ArrayList();

for (int i = 0; i < Client.courseWorkArray.size(); i++) {

for (int j = Client.courseWorkArray.size()- 1; j > i; j--) {

if (Client.courseWorkArray.get(i).getStudentNumber()>Client.courseWorkArray.get(j).getStudentNumber()) {

temp.add(0, Client.courseWorkArray.get(i));

Client.courseWorkArray.set(i,Client.courseWorkArray.get(j));

Client.courseWorkArray.set(j,temp.get(0));

}

}

}

System.out.println("The sorted Array by ID is: ");

for (int i = 0; i < Client.courseWorkArray.size(); i++) {

System.out.println( "ID: " +Client.courseWorkArray.get(i).getStudentNumber()+" Name: "+

Client.courseWorkArray.get(i).getTitleOfStudent()+" "+

Client.courseWorkArray.get(i).getFirstName()+" "+

Client.courseWorkArray.get(i).getLastName()+" "+

Client.courseWorkArray.get(i).getDay()+"/"+

Client.courseWorkArray.get(i).getMonth()+"/"+

Client.courseWorkArray.get(i).getYear()+" "+

"ASSIGNMENT 1: "+Client.courseWorkArray.get(i).getAssignment1()+"% "+

"ASSIGNMENT 2: "+Client.courseWorkArray.get(i).getAssignment2()+"% "+

" Weekly pract. Work: "+Client.courseWorkArray.get(i).getWeeklyPracticalWork()+"% "+

" Final Exam: "+Client.courseWorkArray.get(i).getFinalExam()+"%");

}//end of if

}//end of method

public static void toCsv(){

String fileName ="coursework.csv";

PrintWriter outputStream = null;

try{

outputStream = new PrintWriter(fileName); /\* Give the name of the file as .csv format \*/

}

catch(FileNotFoundException e){ /\* Error message when there is a problem opening the files \*/

System.out.println("Error opening the file"+fileName);

System.exit(0);

}

for(int i=0;i<Client.courseWorkArray.size();i++){

outputStream.println(Client.courseWorkArray.get(i).getTitleOfStudent()+","+

Client.courseWorkArray.get(i).getFirstName()+","+

Client.courseWorkArray.get(i).getLastName()+","+

Client.courseWorkArray.get(i).getStudentNumber()+","+

Client.courseWorkArray.get(i).getDay()+"/"+

Client.courseWorkArray.get(i).getMonth()+"/"+

Client.courseWorkArray.get(i).getYear()+","+

Client.courseWorkArray.get(i).getAssignment1()+","+

Client.courseWorkArray.get(i).getAssignment2()+", "+

Client.courseWorkArray.get(i).getWeeklyPracticalWork()+","+

Client.courseWorkArray.get(i).getFinalExam()); /\* Write the information into files \*/

}

outputStream.close(); /\* Close the files \*/

}

}//end of class

**<RESEARCHSTUDENT CLASS>**

/\*\*

\* Title: Client Change

\* @author Ricardo Laner Jacobsen Teixeira

\* Date: 30/10/2020

\* File name: Assignment 2

\* Purpose: This program was created with the intuition of helping the user on manage data about work course and research students.

\*

\*/

package assignement2;

import static assignement2.Client.researchArray;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.PrintWriter;

import java.util.ArrayList;

import java.util.Scanner;

public class ResearchStudent extends Student {

public static Scanner input = new Scanner (System.in);

//declaring the variables

private int proposalComponent ;

private int finalOralPresentation ;

private int finalThesis ;

// Default constructor

public ResearchStudent() {

this.proposalComponent = proposalComponent;

this.finalOralPresentation =finalOralPresentation;

this.finalThesis =finalThesis;

}

//parametrized constructor

public ResearchStudent(int proposalComponent, int finalOralPresentation, int finalThesis) {

this.proposalComponent = proposalComponent;

this.finalOralPresentation = finalOralPresentation;

this.finalThesis = finalThesis;

}

//parametrized constructor with the super class attributes

public ResearchStudent(int proposalComponent, int finalOralPresentation, int finalThesis, String ts, String fn, String ln, long sn,int day, int month, int year) {

super(ts, fn, ln, sn, day, month, year);

this.proposalComponent = proposalComponent;

this.finalOralPresentation = finalOralPresentation;

this.finalThesis = finalThesis;

}

public int getProposalComponent() {

return proposalComponent;

}

public void setProposalComponent(int proposalComponent) {

this.proposalComponent = proposalComponent;

}

public int getFinalOralPresentation() {

return finalOralPresentation;

}

public void setFinalOralPresentation(int finalOralPresentation) {

this.finalOralPresentation = finalOralPresentation;

}

public int getFinalThesis() {

return finalThesis;

}

public void setFinalThesis(int finalThesis) {

this.finalThesis = finalThesis;

}

public int overallMark(){

int overall= (((this.proposalComponent\*30)+(this.finalOralPresentation\*10)+(this.finalThesis\*60))/100);

return overall;

}

public String finalGrade(){

int fg = this.overallMark();

String situation =" ";

if(fg>=80){situation ="The Research Student " +getFirstName()+" was aproved with the OVERALL "+overallMark()+ "%, and the GRADED AS High Distintion(HD)."; }

else if (fg>=70 && fg<80){situation="The Research Student " +getFirstName()+" was aproved with the OVERALL "+overallMark()+ "%, and the GRADED AS Distintion (D)."; }

else if (fg>=60 && fg<70){situation="The Research Student " +getFirstName()+" was aproved with the OVERALL "+overallMark()+ "%, and the GRADED AS Credit (C)."; }

else if (fg>=50 && fg<60){situation="The Research Student " +getFirstName()+" was aproved with the OVERALL "+overallMark()+ "%, and the GRADED AS Pass (P)."; }

else if (fg<50){situation="The Research Student " +getFirstName()+" was not aproved with the OVERALL " + overallMark() + "%, and the GRADED AS Reproved(N)."; }

return situation;

}

public static void loadResearchStundent() throws FileNotFoundException{

if(Client.researchArray.isEmpty()){

File f = new File ("research.txt");

Scanner txt = new Scanner (f);

while(txt.hasNext()){

String record = txt.nextLine();

Scanner ScanRecord = new Scanner(record);

ScanRecord.useDelimiter("#");

String titleOfStudent, firstName, lastName;

long studentNumber;

int day, month,year, proposalComponent , finalOralPresentation, finalThesis ;

// needed to be recreated because of scan order;

titleOfStudent = ScanRecord.next();

firstName = ScanRecord.next();

lastName = ScanRecord.next();

studentNumber = ScanRecord.nextLong();

day = ScanRecord.nextInt();

month = ScanRecord.nextInt();

year = ScanRecord.nextInt();

proposalComponent=ScanRecord.nextInt();

finalOralPresentation=ScanRecord.nextInt();

finalThesis= ScanRecord.nextInt();

ResearchStudent rs = new ResearchStudent(proposalComponent,finalOralPresentation,finalThesis,titleOfStudent,firstName,lastName,studentNumber,day,month,year);

Client.researchArray.add(rs);

}//end of while loop

} //end of if

}//end of loading method

//delete method by ID

public static int deleteResearchStudent(long id){

int position = -1;

for (int i = 0; i < Client.researchArray.size(); i++) {

if(Client.researchArray.get(i).getStudentNumber()==id){

position = i;

System.out.println("The STUDENT ID "+id+" "+ Client.researchArray.get(i).getFirstName()+" "+Client.researchArray.get(i).getLastName()+" WAS FOUND");

char confirmation= ' ' ;

while(confirmation !='Y' && confirmation !='N' ){

System.out.println("DO YOU CONFIRM THE EXCLUSION? [Y/N]");

confirmation = input.next().toUpperCase().charAt(0);

}

if(confirmation=='Y'){

position = i;

}else {

position = -1;

}

break;

}

}

return position;

}

public static void displayAllResearchStudents(){

System.out.println("All the INFORMATION about RESEARCH STUDENTS:");

System.out.println("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++");

for (int i = 0; i < Client.researchArray.size(); i++) {

System.out.println("STUDENT "+(i+1)+": ");

System.out.println(Client.researchArray.get(i).getTitleOfStudent()+" "+

Client.researchArray.get(i).getFirstName()+" "+

Client.researchArray.get(i).getLastName()+" "+

Client.researchArray.get(i).getStudentNumber()+" "+

Client.researchArray.get(i).getDay()+"/"+

Client.researchArray.get(i).getMonth()+"/"+

Client.researchArray.get(i).getYear()+" "+

"Propoal Component: "+Client.researchArray.get(i).getProposalComponent()+"% "+

" Final Oral Presentation: "+Client.researchArray.get(i).getFinalOralPresentation()+"% "+

" Final Thesis: "+Client.researchArray.get(i).getFinalThesis()+"% ");

String situation= Client.researchArray.get(i).finalGrade();

System.out.println(situation+"\n");

}//end of for loop

}//end of method

public static void AvarageResearch(){

int counter =0;

int overall;

for (int i = 0; i < Client.researchArray.size(); i++) {

overall=Client.researchArray.get(i).overallMark();

if (overall>=50) {

counter +=1;

}

}

System.out.println("The system found FROM RESEARCH STUDENTS "+counter+" STUDEND(S) which had the overall EQUAL TO OR ABOVE THE AVARAGE.");

counter =0;//cleaning for the next loop

for (int i = 0; i < Client.researchArray.size(); i++) {

overall=Client.researchArray.get(i).overallMark();

if (overall<50) {

counter +=1;

}

}

System.out.println("The system found FROM RESEARCH STUDENTS "+counter+" STUDEND(S) which had the overall BELLOW THE AVARAGE.");

}//end of method

public static int findID(long id ){

int ret=0;

for (int i = 0; i < Client.researchArray.size(); i++) {

if (Client.researchArray.get(i).getStudentNumber()==id) {

System.out.println("The ID WAS FOUND in Research student.");

System.out.println("STUDENT INFORMATION");

System.out.println("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++");

break;

} else{

ret=-1;

}

}//end of for

for (int i = 0; i < Client.researchArray.size(); i++) {

if (Client.researchArray.get(i).getStudentNumber()==id) {

System.out.println(Client.researchArray.get(i).getTitleOfStudent()+" "+

Client.researchArray.get(i).getFirstName()+" "+

Client.researchArray.get(i).getLastName()+" "+

Client.researchArray.get(i).getStudentNumber()+" "+

Client.researchArray.get(i).getDay()+"/"+

Client.researchArray.get(i).getMonth()+"/"+

Client.researchArray.get(i).getYear()+" "+

"Propoal Component: "+Client.researchArray.get(i).getProposalComponent()+"% "+

" Final Oral Presentation: "+Client.researchArray.get(i).getFinalOralPresentation()+"% "+

" Final Thesis: "+Client.researchArray.get(i).getFinalThesis()+"% \n");

}//end of if

}//end of for

return ret;

}//end of find ID

public static int findByName(String name , String lastName){

int ret=0;

for (int i = 0; i < Client.researchArray.size(); i++) {

//to receive the name from the arrray

String nameinarr =Client.researchArray.get(i).getFirstName();

if (nameinarr.equalsIgnoreCase(name)) {

nameinarr=Client.researchArray.get(i).getLastName();

if (nameinarr.equalsIgnoreCase(lastName)) {

System.out.println("THE NAME WAS FOUND IN RESEARCH ");

System.out.println("STUDENT INFORMATION");

System.out.println("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++");

break;

}

}else {

ret=-1;

}

}

for (int i = 0; i < Client.researchArray.size(); i++) {

//to receive the name from the arrray

String nameinarr =Client.researchArray.get(i).getFirstName();

if (nameinarr.equalsIgnoreCase(name)) {

nameinarr=Client.researchArray.get(i).getLastName();

if (nameinarr.equalsIgnoreCase(lastName)) {

System.out.println(Client.researchArray.get(i).getTitleOfStudent()+" "+

Client.researchArray.get(i).getFirstName()+" "+

Client.researchArray.get(i).getLastName()+" "+

Client.researchArray.get(i).getStudentNumber()+" "+

Client.researchArray.get(i).getDay()+"/"+

Client.researchArray.get(i).getMonth()+"/"+

Client.researchArray.get(i).getYear()+" "+

"Propoal Component: "+Client.researchArray.get(i).getProposalComponent()+"% "+

" Final Oral Presentation: "+Client.researchArray.get(i).getFinalOralPresentation()+"% "+

" Final Thesis: "+Client.researchArray.get(i).getFinalThesis()+"% ");

}

}

}

return ret;

}//end of findByName

public static void sortByID(){

ArrayList<ResearchStudent> temp = new ArrayList();

for (int i = 0; i < Client.researchArray.size(); i++) {

for (int j = Client.researchArray.size()- 1; j > i; j--) {

if (Client.researchArray.get(i).getStudentNumber()> Client.researchArray.get(j).getStudentNumber()) {

temp.add(0, Client.researchArray.get(i));

Client.researchArray.set(i,Client.researchArray.get(j));

Client.researchArray.set(j,temp.get(0));

}

}

}

System.out.println("The sorted Array by ID is: ");

for (int i = 0; i < Client.researchArray.size(); i++) {

System.out.println( "ID: " +Client.researchArray.get(i).getStudentNumber()+" Name: "+

Client.researchArray.get(i).getTitleOfStudent()+" "+

Client.researchArray.get(i).getFirstName()+" "+

Client.researchArray.get(i).getLastName()+" "+

Client.researchArray.get(i).getDay()+"/"+

Client.researchArray.get(i).getMonth()+"/"+

Client.researchArray.get(i).getYear()+" "+

"Propoal Component: "+Client.researchArray.get(i).getProposalComponent()+"% "+

" Final Oral Presentation: "+Client.researchArray.get(i).getFinalOralPresentation()+"% "+

" Final Thesis: "+Client.researchArray.get(i).getFinalThesis()+"% ");

}//end of if

}//end of method

public static void toCsv(){

String fileName ="research.csv";

PrintWriter outputStream = null;

try{

outputStream = new PrintWriter(fileName); /\* Give the name of the file as .csv format \*/

}

catch(FileNotFoundException e){ /\* Error message when there is a problem opening the files \*/

System.out.println("Error opening the file"+fileName);

System.exit(0);

}

for(int i=0;i<Client.researchArray.size();i++){

outputStream.println(researchArray.get(i).getTitleOfStudent()+","+

Client.researchArray.get(i).getFirstName()+","+

Client.researchArray.get(i).getLastName()+","+

Client.researchArray.get(i).getStudentNumber()+", "+

Client.researchArray.get(i).getDay()+"/"+

Client.researchArray.get(i).getMonth()+"/"+

Client.researchArray.get(i).getYear()+","+

Client.researchArray.get(i).getProposalComponent()+","+

Client.researchArray.get(i).getFinalOralPresentation()+","+

Client.researchArray.get(i).getFinalThesis()); /\* Write the information into files \*/

}

outputStream.close(); /\* Close the files \*/

}

}//end of class