**Joseph Tierney**

**Computer Science 2**

**OOP Assignment 2**

**Contents**

Introduction ………………………….. 3

Steps …………………………………….. 4

Program Walkthrough ………….. 10

Test Bed ……………………………….. 30

Conclusion ……………………………. 36

**Introduction**

This assignment will cover the concept of inheritance and polymorphism, using multiple classes and using base classes. The task for this is assignment is to write a system for a college application process. The system will be made up of a number of different classes. The first class is the Application class. Application class has a course name and course number. The second class is an Applicant class, this class has a name and an id. The third class will be Student class, this has an application and an address, and is an applicant – this class will extend the Applicant class. MatureStudent class is a student and has a date of birth. The test class will have an array of students which will hold student objects and mature student objects, which the user will enter the information for. When the user enters the information, there will be two menus, the first one displayed gives the user a choice to either View Course details, Edit Course/Personal details or Quit. The second menu (which the user can access by clicking Edit), will give the user 5 options – View application details, View personal details, Change application details, Change personal details and Quit. When the user ends the program, all objects will be written to a file.

**Steps**

**Application Class**

1. Start Application class
2. Declare variables as private
3. Create constructor with variables String course\_name and int course\_number
   1. Validate variables in constructor using this.course\_name etc.
4. End constructor
5. Start toString method
   1. Return course name and course number
6. End toString method
7. Set Course Name method
   1. Again, validate variables using this.course\_name
8. End Set Course Name method
9. Get Course Name method
   1. Return course\_name
10. End Get Course Name method
11. Set Course Number method
    1. Validate variables using this.course\_number
12. End Set Course Number method
13. Get Course Number method
    1. Return course\_number
14. End Get Course Number method
15. End Application class

**Applicant Class**

1. Start Applicant class
2. Declare variables as private
3. Create constructor with variables String name and int id
   1. Validate variables in constructor using this.name etc.
4. End constructor
5. Start toString method
   1. Return name and id
6. End toString method
7. Set Name method
   1. Validate variable using this.name
8. End Set Name method
9. Get Name Method
   1. Return name
10. End Get Name Method
11. Set ID method
    1. Validate variable using this.id
12. End Set ID method
13. Get ID method
    1. Return ID
14. End Get ID method
15. End Applicant class

**Student Class**

1. Start Student class extends applicant
2. Declare variables as private
3. Create constructor with variables from the super class String name, int id and its own variables String address, application a1
   1. Call super class constructor using super(name,id)
   2. Validate variables using this.address etc.
4. End constructor
5. Start toString method
   1. Return super.toString() + address + a1
6. End toString method
7. Set Address method
   1. Validate variables using this.address
8. End Set Address method
9. Get Address method
   1. Return address
10. End Get Address method
11. Set Application a1 method
    1. Validate variable using this.a1
12. End Set Application a1 method
13. Get Application a1 method
    1. Return a1.toString()
14. End Get Application a1 method
15. Get name method
    1. Return super.getname()
16. End get name method
17. Get ID method
    1. Return super.getid()
18. End Get ID method
19. End student class

**Mature Student Class**

1. Start maturestudent extends student class
2. Declare variables as private
3. Create constructor with variables from the super class String name, int id, String address, application a1 and its own variable String dob
   1. Call super class constructor using super(name,id,address,a1)
   2. Validate variable using this.dob
4. End constructor
5. Start toString method
   1. Return super.toString() + dob
6. End toString method
7. End maturestudent class

**Application Test**

1. Import Scanner
2. Import Date
3. Import IO
4. Start Class
5. Declare variables
6. Start loop to determine how many students there are from user input
7. Prompt the user to enter number of students
   1. While the input doesn’t match digits
      1. Print “Error: Digits Only”
      2. Beep
      3. Prompt the user again
   2. End validation
8. Parse the int
9. Create the array of students
10. Start the for loop to take in the students details
    1. Prompt the user to enter a name
    2. Prompt the user to enter an ID
       1. While the input ID doesn’t match digits
          1. Print “Error: Digits Only”
          2. Beep
          3. Prompt the user again
       2. End validation
    3. Parse the int
    4. Prompt the user to enter an Address
    5. Prompt the user to enter a Course Name
    6. Prompt the user to enter a Course Number
       1. While the input Course Number doesn’t match digits
          1. Print “Error: Digits Only”
          2. Beep
          3. Prompt the user again
       2. End validation
    7. Parse the int
    8. Ask the user if the student is a mature student
       1. While the input y/n doesn’t match y or n
          1. Beep
          2. Print “Error: Y/N only”
          3. Prompt the user again
       2. End validation
    9. If the input matches Y
       1. While the flag is false
          1. Prompt the user to enter the date of birth
          2. If the dob doesn’t equal to length 10
             1. Print “Invalid format” and prompt user again
          3. Else
             1. Check1 and Check2 are equal to ‘/’
          4. If check1 and check to are equal to ‘/’
             1. Change the flag to true
          5. Else prompt the user again
    10. Create application object
    11. Create maturestudent object
    12. Else
        1. Create application object
        2. Crate student object
    13. While the user input is not 3
        1. Display the first menu
        2. Take in input from user
        3. While the input doesn’t match 1-3
           1. Print “Error: 1-3 Only”
           2. Beep
           3. Prompt user again
        4. Parse the int
        5. If the user inputs 1
           1. Print “Enter ID of Applicant”
           2. Take in input from user
           3. While the user input doesn’t equal digits
              1. Print “Error: Digits Only”
              2. Beep
              3. Prompt the user again
           4. Parse the int
           5. Check if the id is in the array
              1. If the id is in the array
              2. Sub = i
              3. Flag2 is true
              4. Print arraystudents[i].toString()
           6. If flag2 is still false
              1. Print “Student Not found”
           7. Reset the flag2 to false
        6. If the user inputs 2
           1. Print “Enter ID of Applicant to Edit”
           2. Take in input from user
           3. While the user input doesn’t equal digits
              1. Print “Error: Digits Only”
              2. Beep
              3. Prompt user again
           4. Parse the int
           5. Check if the id is in the array
              1. If the id is in the array
              2. Sub = i
              3. Flag2 is true
              4. Print arraystudents[i].toString()
           6. If flag2 is still false
              1. Print “Student Not found”
           7. Display the second menu
           8. Take input in from user
           9. While the input doesn’t match 1-5
              1. Print “Error: 1-5 Only”
              2. Prompt the user again
           10. Parse the int
           11. If the input is 1
               1. Print arraystudents[sub].getapplication()
           12. If the input is 2
               1. Print arraystudents[sub].getname()
               2. Print arraystudents[sub].getid()
               3. Print arraystudents[sub].getaddress()
           13. If the input is 3
               1. Print “Enter 1 to Change Course name, Enter 2 to change course ID”
               2. Take input in from user
               3. While the input doesn’t match 1-2

Print “Error: 1-2 only”

Prompt the user again

* + - * 1. Parse the int
        2. If the user enters 1

Create application

Print “Enter course name:”

A1.setcname(course\_name = input.next())

Set application

Print arraystudents[sub].getapplication()

* + - * 1. If the user enters 2

Create application

Print “Enter course ID”

A1.setcnumber(course\_number = input.nextInt())

Set application

Print arraystudents[sub].getapplication()

* + - 1. If the user inputs 4
         1. Printer “Enter 1 to change name, enter 2 to change id”
         2. Take input in from user
         3. While the input doesn’t match 1-2

Print “Error: 1-2 only”

Prompt the user again

* + - * 1. Parse the int
        2. If the input is 1

Print “Enter name”

Take in the name

Set the name: Arraystudents[sub].setname();

Get the name: Arraystudents[sub].getname();

Print student details

* + - * 1. If the input is 2

Print “Enter id”

Take in the id

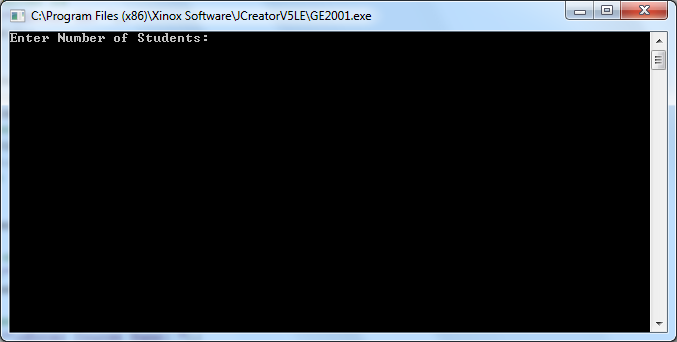
Set the ID: Arraystudents[sub].setid

Get the ID: Arraystudents[sub].getid()

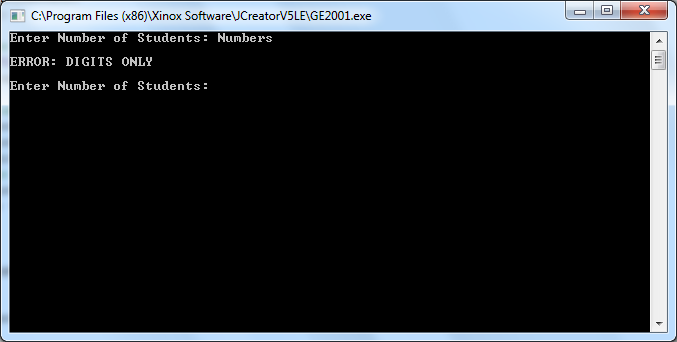
Print student details

1. Write to the file
2. Print “Thank you for using CAO online”

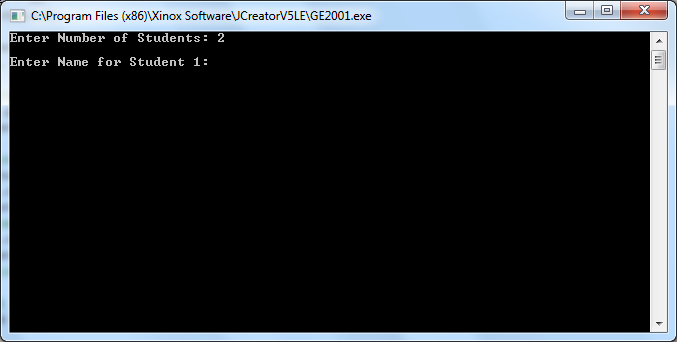
**Program Walkthrough**



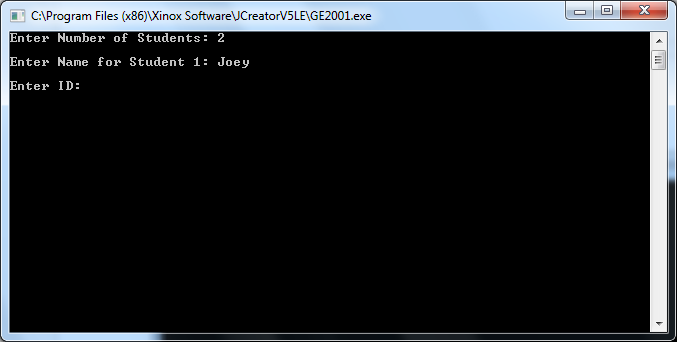
The program starts by prompting the user to enter the number of students they wish to enter.



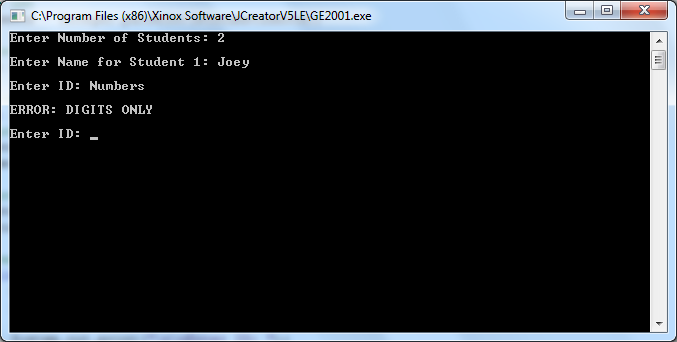
If the user enters in letters instead of digits they are met with an error and are prompted to enter the number of students again.



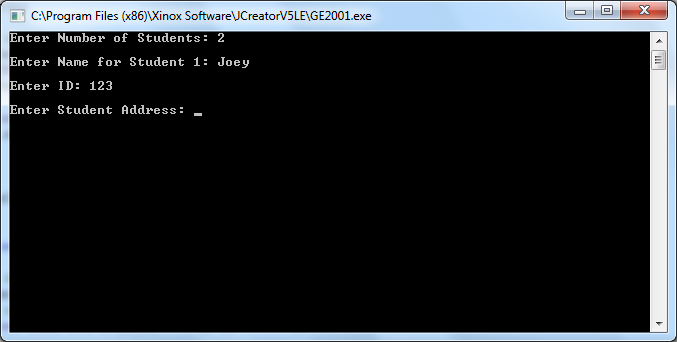
When the user enters the number of students they wish to enter, they are met with another prompt asking them for the name of the first student.



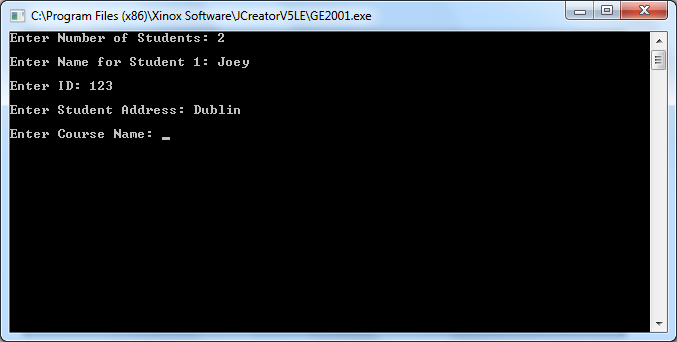
When the user enters a name, they are then prompted to enter the students ID number.



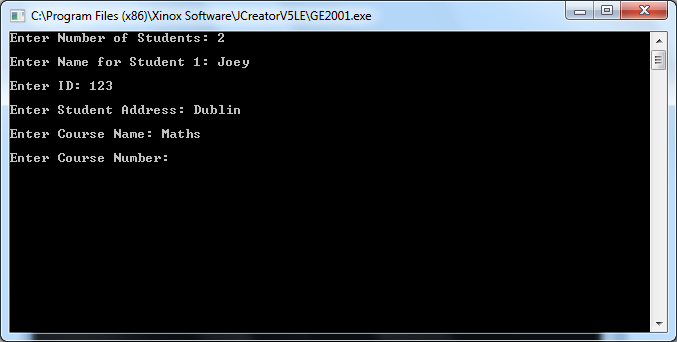
If the user enters letters instead of digits they are met with an error telling them to enter digits only and are prompted to enter the ID again.



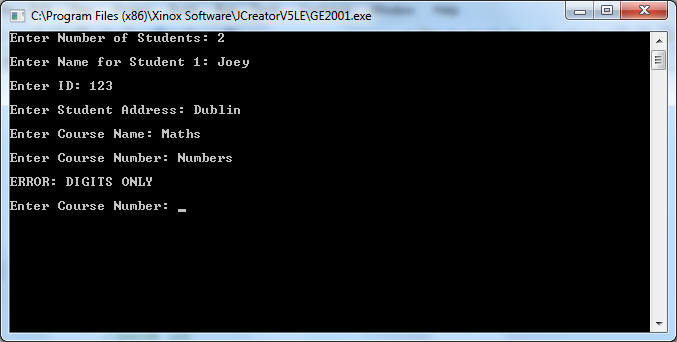
When the user enters a valid ID, they are then prompted to enter the students address.



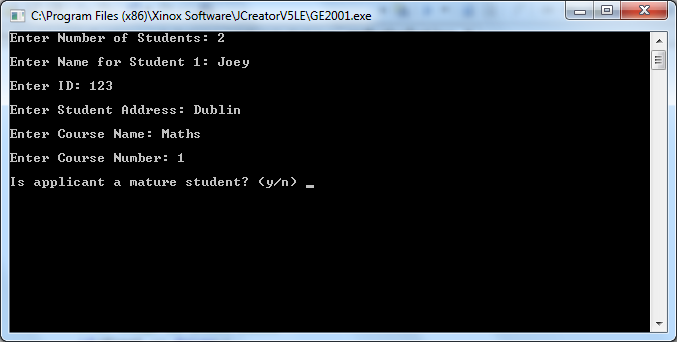
When they enter the students address, the user will then be prompted to enter the course name.



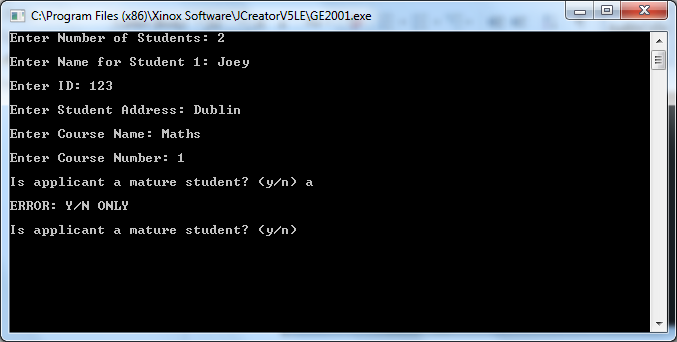
When the user enters a course name, they are then prompted to enter the course ID number.



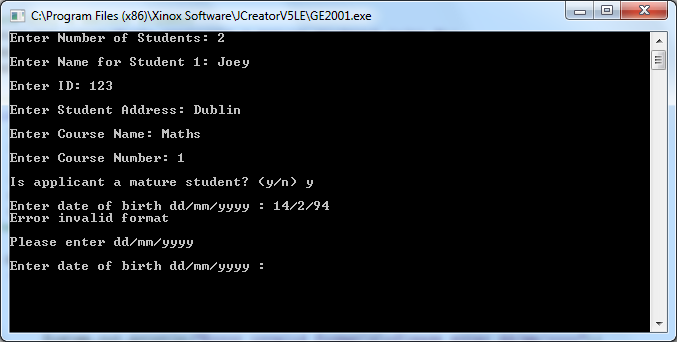
If the user enters letters instead of numbers they are met with an error telling them to enter digits only. The user will then be prompted to enter the course number again.



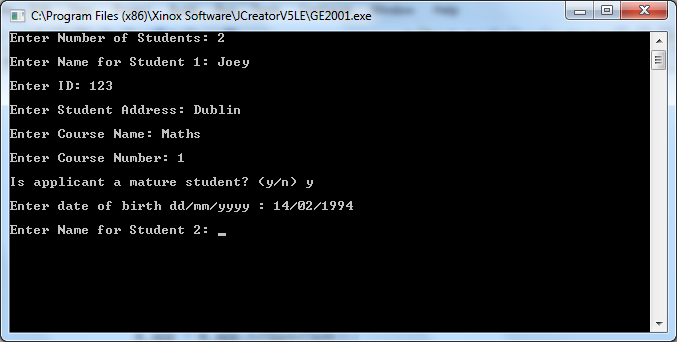
When the user enters the course number, they are then asked if the student is a mature student.



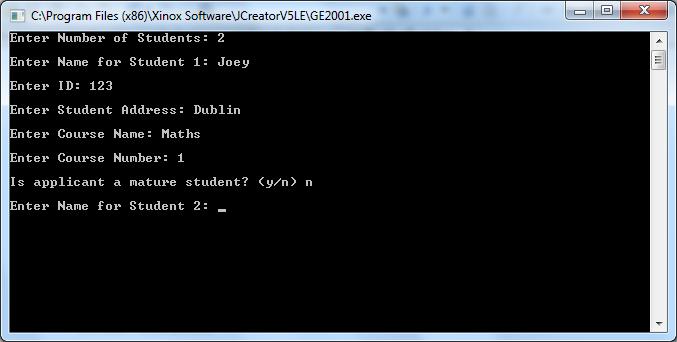
If the user enters anything other than y or n, they are met with an error and are prompted to enter y/n again.



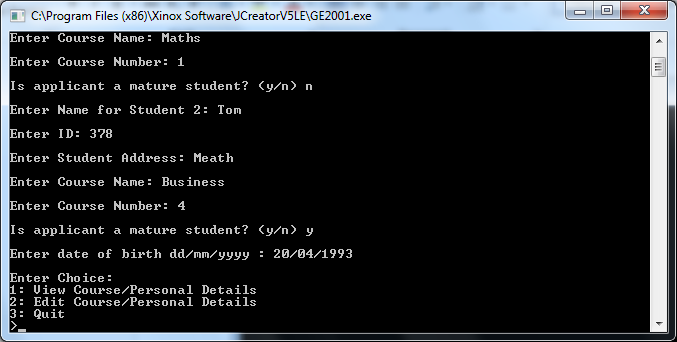
If the user selects y, they are prompted to enter a date of birth in the following format: dd/mm/yyyy. If the user enters an invalid format, they are met with an error telling them that the format was invalid and are prompted again to enter the date of birth.



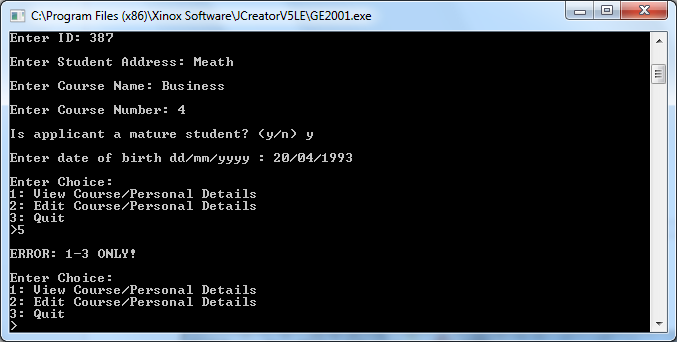
When the user enters a valid format a mature student object will be created, they are then prompted to enter the name of the next student.



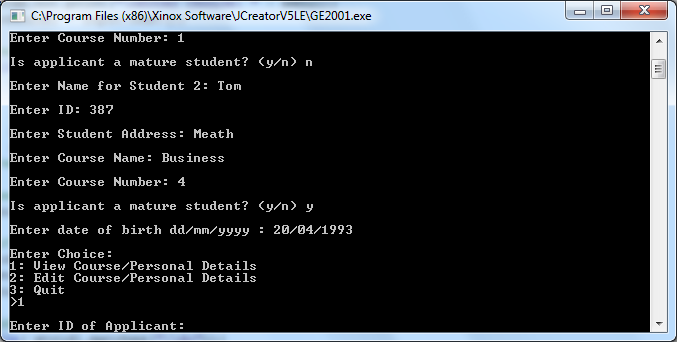
If the user selects n for mature student a regular student object will be created, they are then prompted to enter the name of the next student.



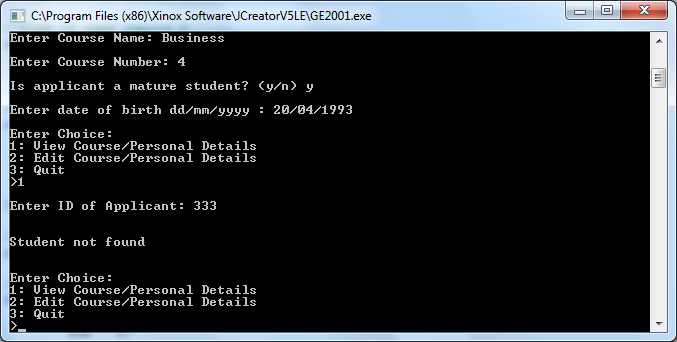
When the user finishes entering all the students’ details, they are met with a menu prompting them with 3 options: 1: View Course/Personal, 2: Edit Course/Personal Details, 3: Quit.



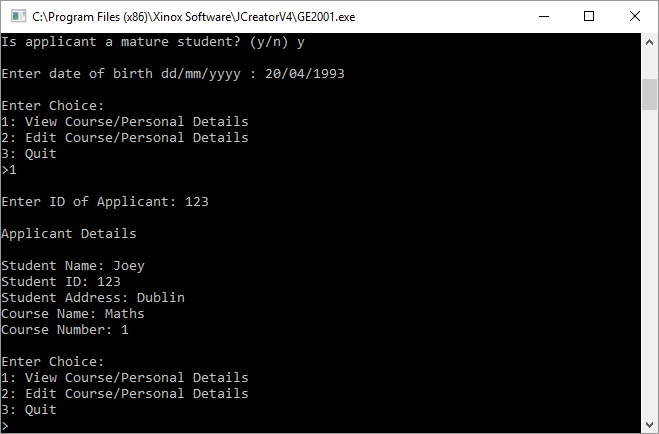
If the user enters letters or an option that is not 1, 2 or 3 they are met with an error telling them that they have to enter 1-3 only, and the menu is then shown to them again.



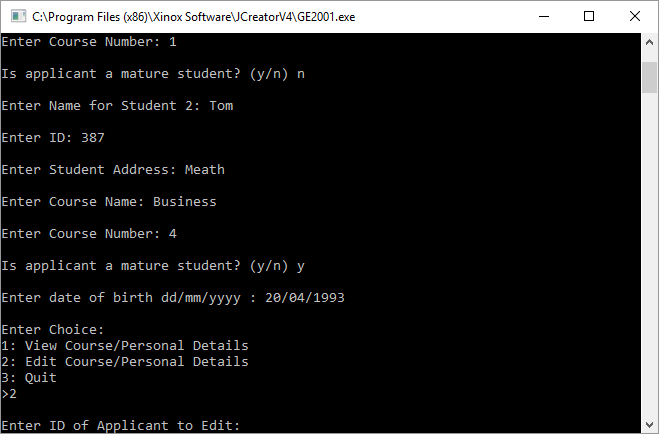
When the user enters 1 to view course/personal details they are prompted to enter the ID of an applicant.



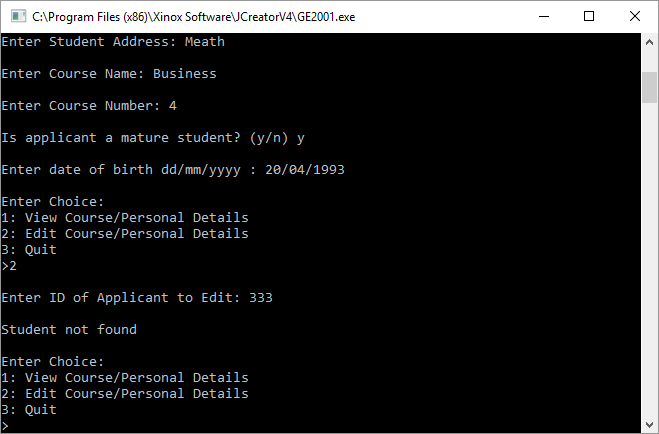
If the user enters an ID that is not in the array, or they enter a value that isn’t digits, they are told that the student couldn’t be found and the menu is displayed again.



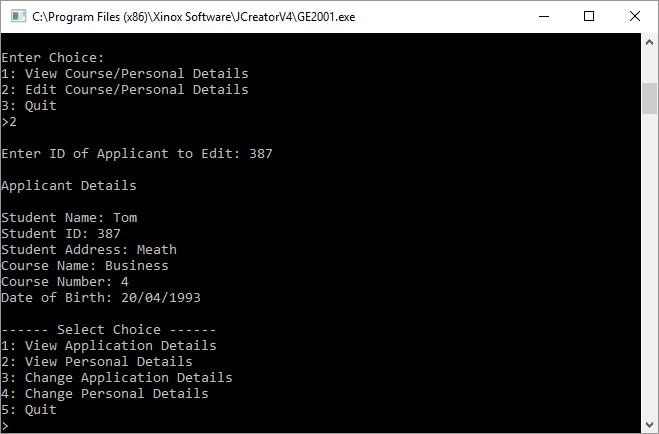
When the user enter a valid ID, the applicant’s details are displayed to them and the menu is redisplayed.



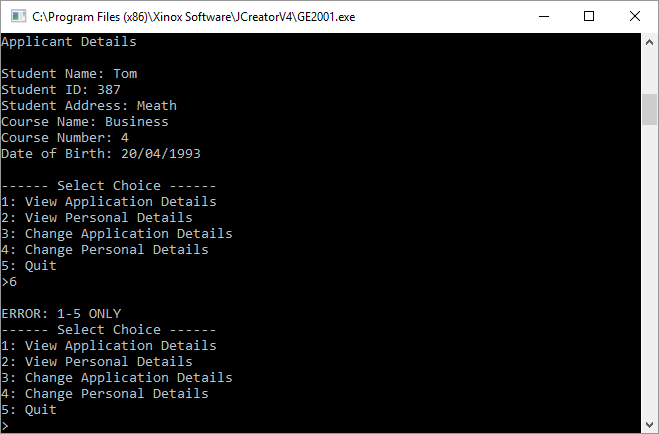
When the user enters inputs 2 to edit course or personal details, they are once again asked to enter the ID of the applicant.



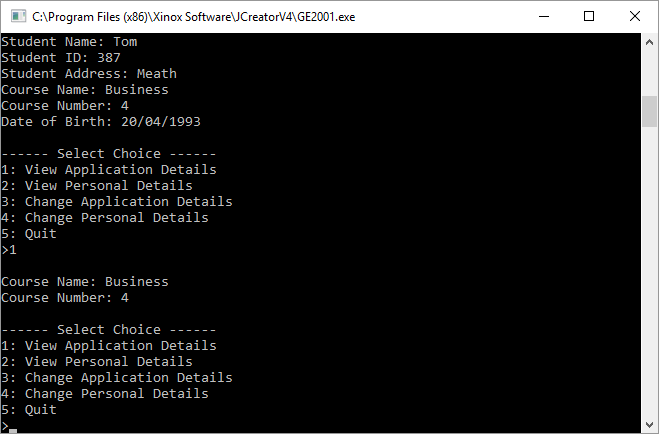
If the ID is not in the array, the user is told that the student could not be found and the main menu is displayed again.



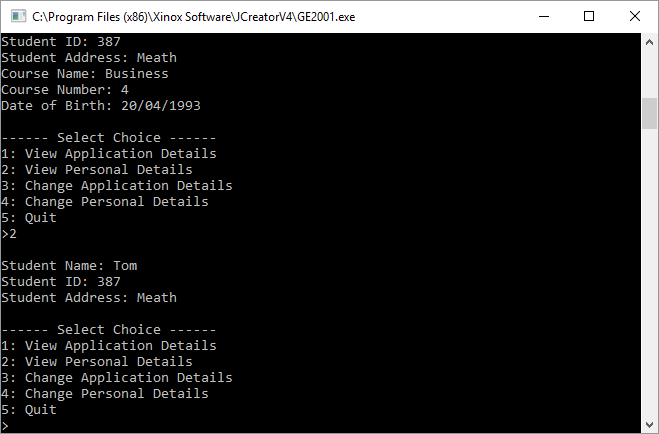
If the ID is in the array, the student is shown the details of the applicant they have chosen to edit, and a new menu will appear prompting the user to enter a choice of 5 options.



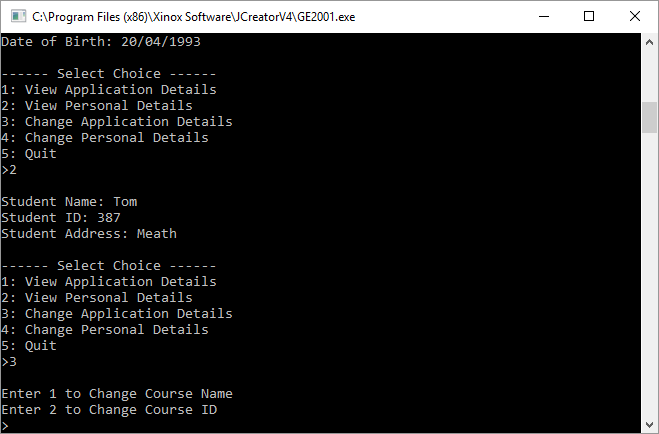
If the user enters an option greater than 5 or one that isn’t digits, they are met with an error telling them 1-5 only, and the menu is redisplayed.



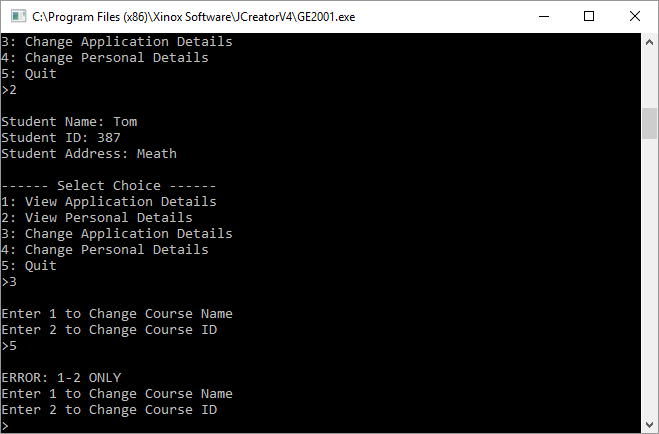
If the user enters 1, the applicants application details are displayed, and the menu will be redisplayed for another prompt.



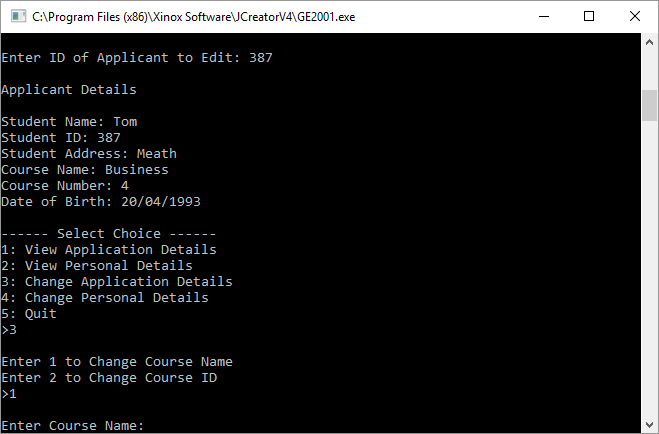
When the user inputs 2, the applicant’s personal details are displayed, and the menu is also redisplayed for another prompt.



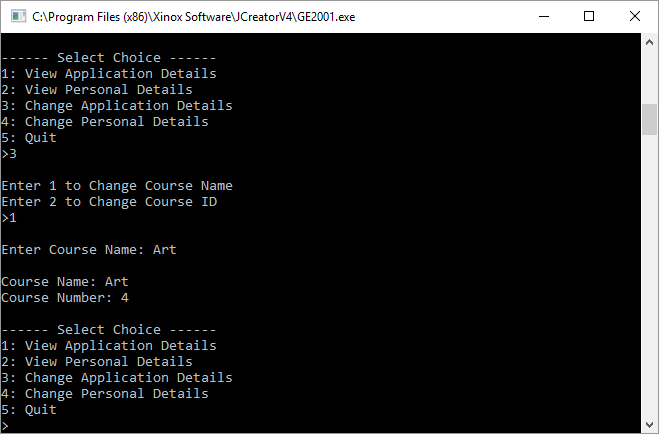
When the user inputs 3, they are met with an option to enter 1 to change course name, or 2 to change course ID.



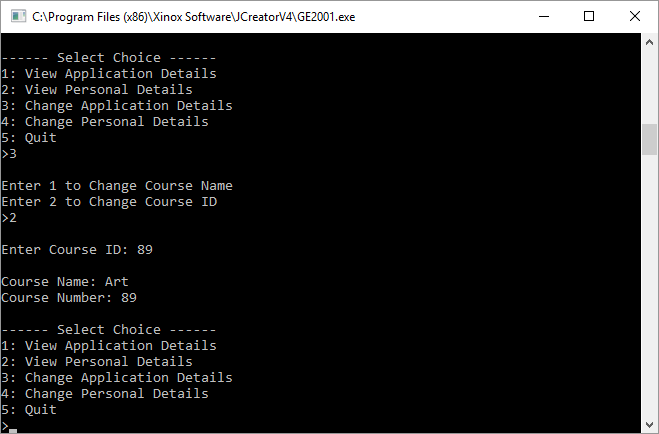
If the user inputs anything other than 1-2, they are met with an error telling them to enter 1-2 only, and the options are displayed again.



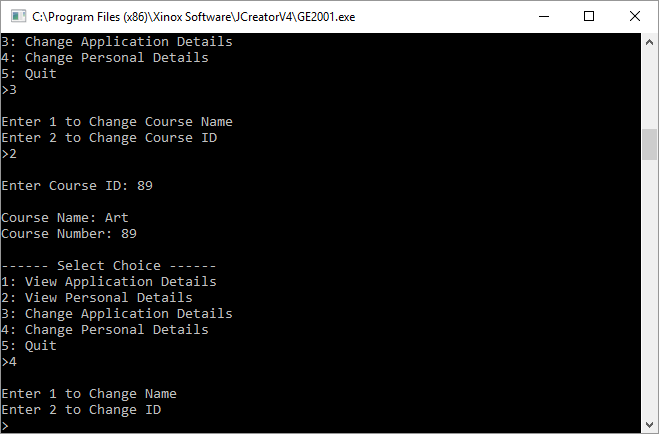
If the user enters 1, they are prompted to enter a new course name.



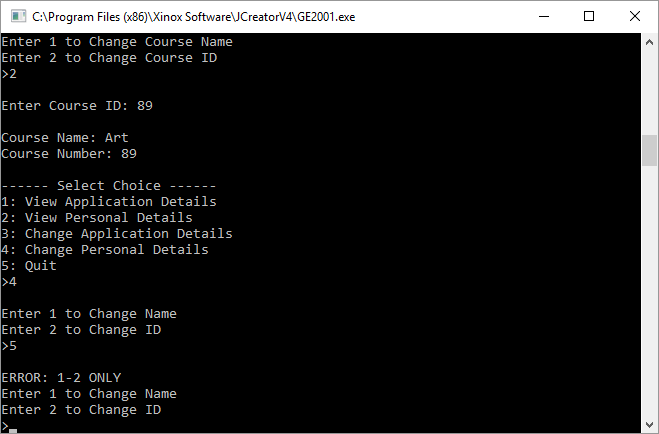
When the user enters a new course name and hits enter, the updated applicant course details will be displayed along with the menu.



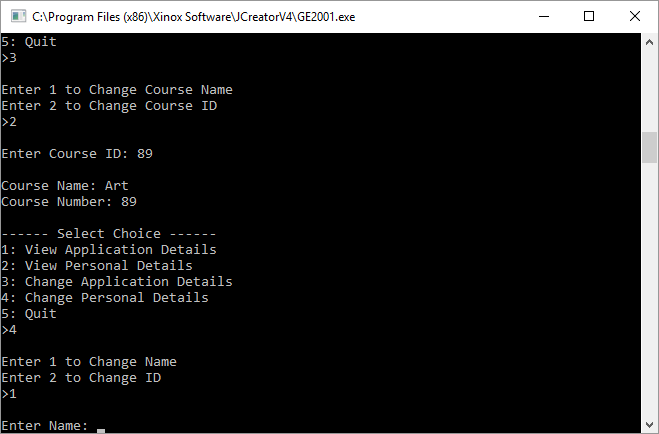
If the user selects 2, they are prompted to enter a new course ID. When they hit enter, the applicant’s course details will be updated and displayed along with the menu.



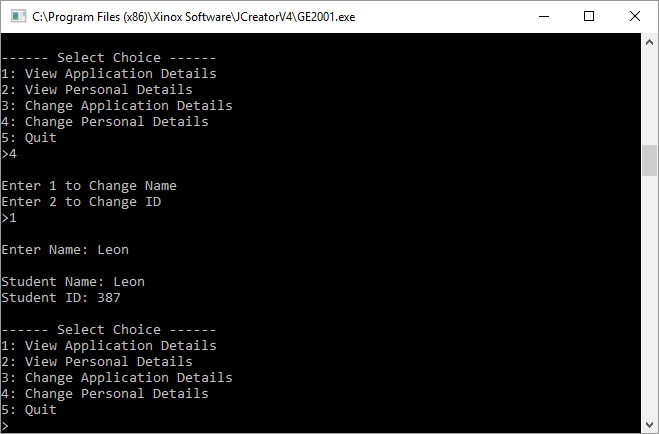
When the user selects 4, they are met with 2 options to enter 1 to change name, and 2 to change ID.



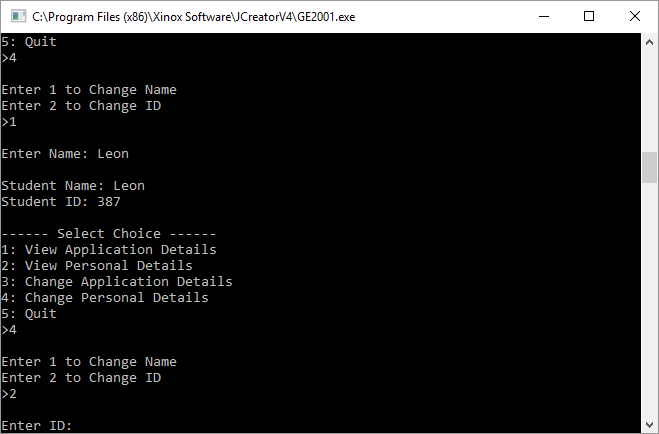
Entering anything other than 1-2 will display an error message to the user informing them to enter 1-2 only, and the options will be redisplayed.



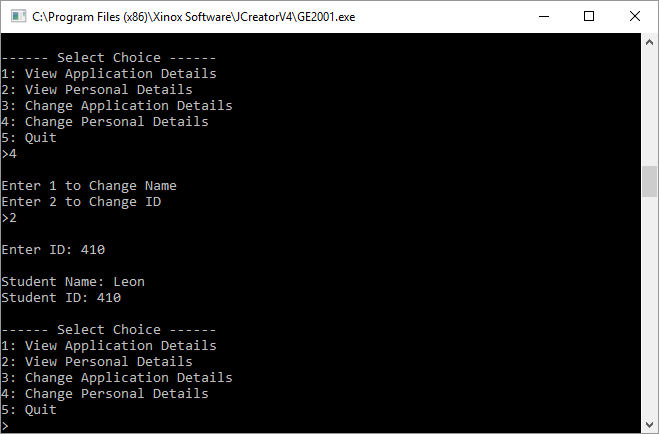
When the user enters 1, they are prompted to change the student’s name.



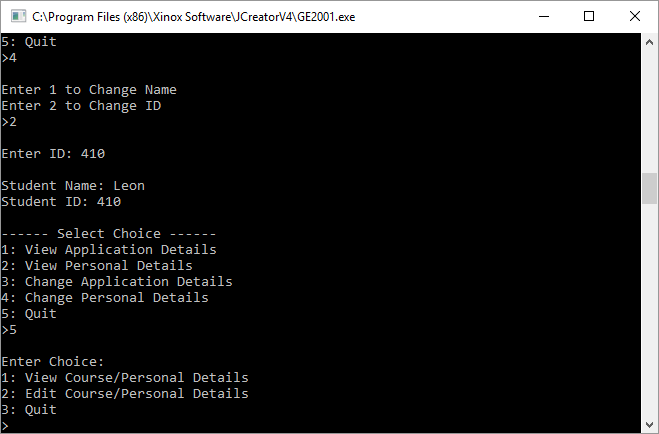
When the user enters a new name and hits enter, the updated name and their current ID will be displayed, and the menu will be redisplayed.



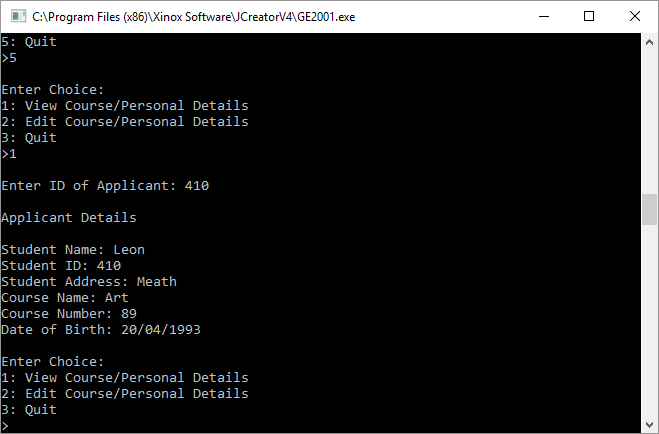
When the user enters 2, they are prompted to enter the student’s new ID.



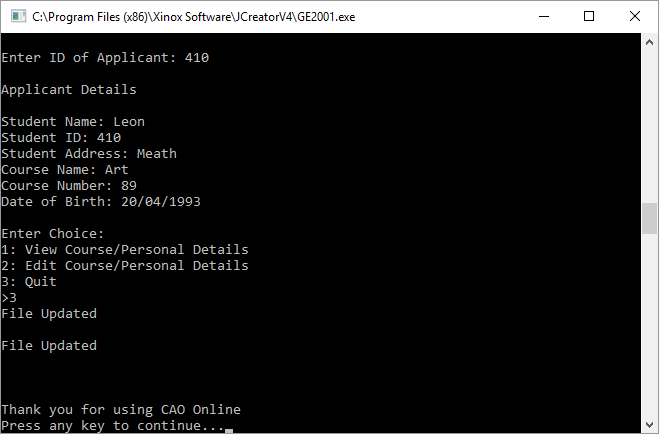
When the user enters a new ID, the student’s personal details will be displayed and the menu will be redisplayed.



When the user inputs 5, they are brought back to the main menu.



When the user selects 1 and enters the updated ID, the updated Students details are displayed, and the main menu is redisplayed.



When the user inputs 3 to quit the program, the system will write all currently entered students into a text file, and the program ends.

**Test Bed**

|  |  |  |  |
| --- | --- | --- | --- |
| **Scenario** | **Data Entered** | **Expected Result** | **Actual Result** |
| Enter Number of Students: | Number | ERROR: DIGITS ONLY Enter Number of Students: | ERROR: DIGITS ONLY Enter Number of Students: |
| Enter Number of Students: | 2 | Enter Name for Student 1: | Enter Name for Student 1: |
| Enter Name for Student 1: | Joey | Enter ID: | Enter ID: |
| Enter ID: | Number | ERROR: DIGITS ONLY  Enter ID: | ERROR: DIGITS ONLY  Enter ID: |
| Enter ID: | 123 | Enter Student Address: | Enter Student Address: |
| Enter Student Address: | Dublin | Enter Course Name: | Enter Course Name: |
| Enter Course Name: | Maths | Enter Course Number: | Enter Course Number: |
| Enter Course Number: | Number | ERROR: DIGITS ONLY Enter Course Number: | ERROR: DIGITS ONLY Enter Course Number: |
| Enter Course Number: | 1 | Is applicant a mature student? (y/n) | Is applicant a mature student? (y/n) |
| Is applicant a mature student? (y/n) | a | ERROR: Y/N ONLY Is applicant a mature student? (y/n) | ERROR: Y/N ONLY Is applicant a mature student? (y/n) |
| Is applicant a mature student? (y/n) | n | Enter Name for Student 2: | Enter Name for Student 2: |
| Enter Name for Student 2: | Tom | Enter ID: | Enter ID: |
| Enter ID: | Number | ERROR: DIGITS ONLY  Enter ID: | ERROR: DIGITS ONLY  Enter ID: |
| Enter ID: | 387 | Enter Student Address: | Enter Student Address: |
| Enter Student Address: | Meath | Enter Course Name: | Enter Course Name: |
| Enter Course Name: | Business | Enter Course Number: | Enter Course Number: |
| Enter Course Number: | Number | ERROR: DIGITS ONLY Enter Course Number: | ERROR: DIGITS ONLY Enter Course Number: |
| Enter Course Number: | 4 | Is applicant a mature student? (y/n) | Is applicant a mature student? (y/n) |
| Is applicant a mature student? (y/n) | a | ERROR: Y/N ONLY Is applicant a mature student? (y/n) | ERROR: Y/N ONLY Is applicant a mature student? (y/n) |
| Is applicant a mature student? (y/n) | y | Enter date of birth dd/mm/yyyy: | Enter date of birth dd/mm/yyyy: |
| Enter date of birth dd/mm/yyyy: | 20/4/1993 | Error invalid format Please Enter dd/mm/yyyy  Enter date of birth dd/mm/yyyy: | Error invalid format Please Enter dd/mm/yyyy  Enter date of birth dd/mm/yyyy: |
| Enter date of birth dd/mm/yyyy: | 20/04/1993 | Enter Choice:  1: View Course/Personal Details  2: Edit Course/Personal Details  3: Quit | Enter Choice:  1: View Course/Personal Details  2: Edit Course/Personal Details  3: Quit |
| Enter Choice:  1: View Course/Personal Details  2: Edit Course/Personal Details  3: Quit | 4 | ERROR: 1-3 ONLY  Enter Choice:  1: View Course/Personal Details  2: Edit Course/Personal Details  3: Quit | ERROR: 1-3 ONLY  Enter Choice:  1: View Course/Personal Details  2: Edit Course/Personal Details  3: Quit |
| Enter Choice:  1: View Course/Personal Details  2: Edit Course/Personal Details  3: Quit | 1 | Enter ID of Applicant: | Enter ID of Applicant: |
| Enter ID of Applicant: | 333 | Student not found  Enter Choice:  1: View Course/Personal Details  2: Edit Course/Personal Details  3: Quit | Student not found  Enter Choice:  1: View Course/Personal Details  2: Edit Course/Personal Details  3: Quit |
| Enter ID of Applicant: | 387 | Applicant Details  Enter Choice:  1: View Course/Personal Details  2: Edit Course/Personal Details  3: Quit | Applicant Details  Enter Choice:  1: View Course/Personal Details  2: Edit Course/Personal Details  3: Quit |
| Enter Choice:  1: View Course/Personal Details  2: Edit Course/Personal Details  3: Quit | 2 | Enter ID of Applicant to Edit: | Enter ID of Applicant to Edit: |
| Enter ID of Applicant to Edit: | 333 | Student not found  Enter Choice:  1: View Course/Personal Details  2: Edit Course/Personal Details  3: Quit | Student not found  Enter Choice:  1: View Course/Personal Details  2: Edit Course/Personal Details  3: Quit |
| Enter ID of Applicant to Edit: | 387 | Applicant Details  1: View Application Details  2: View Personal Details  3: Change Application Details  4: Change Personal Details  5: Quit | Applicant Details  1: View Application Details  2: View Personal Details  3: Change Application Details  4: Change Personal Details  5: Quit |
| Applicant Details  1: View Application Details  2: View Personal Details  3: Change Application Details  4: Change Personal Details  5: Quit | 6 | ERROR: 1-5 ONLY  1: View Application Details  2: View Personal Details  3: Change Application Details  4: Change Personal Details  5: Quit | ERROR: 1-5 ONLY  1: View Application Details  2: View Personal Details  3: Change Application Details  4: Change Personal Details  5: Quit |
| 1: View Application Details  2: View Personal Details  3: Change Application Details  4: Change Personal Details  5: Quit | 1 | Course Name  Course Number  1: View Application Details  2: View Personal Details  3: Change Application Details  4: Change Personal Details  5: Quit | Course Name  Course Number  1: View Application Details  2: View Personal Details  3: Change Application Details  4: Change Personal Details  5: Quit |
| 1: View Application Details  2: View Personal Details  3: Change Application Details  4: Change Personal Details  5: Quit | 2 | Student Name  Student ID  Student Address  1: View Application Details  2: View Personal Details  3: Change Application Details  4: Change Personal Details  5: Quit | Student Name  Student ID  Student Address  1: View Application Details  2: View Personal Details  3: Change Application Details  4: Change Personal Details  5: Quit |
| 1: View Application Details  2: View Personal Details  3: Change Application Details  4: Change Personal Details  5: Quit | 3 | Enter 1 to Change Course Name  Enter 2 to Change Course Number | Enter 1 to Change Course Name  Enter 2 to Change Course Number |
| Enter 1 to Change Course Name  Enter 2 to Change Course Number | 3 | ERROR: 1-2 ONLY  Enter 1 to Change Course Name  Enter 2 to Change Course Number | ERROR: 1-2 ONLY  Enter 1 to Change Course Name  Enter 2 to Change Course Number |
| Enter 1 to Change Course Name  Enter 2 to Change Course Number | 1 | Enter Course Name | Enter Course Name |
| Enter Course Name | Art | Course Name  Course Number  1: View Application Details  2: View Personal Details  3: Change Application Details  4: Change Personal Details  5: Quit | Course Name  Course Number  1: View Application Details  2: View Personal Details  3: Change Application Details  4: Change Personal Details  5: Quit |
| Enter 1 to Change Course Name  Enter 2 to Change Course Number | 2 | Enter Course ID | Enter Course ID |
| Enter Course ID | 471 | Course Name  Course Number  1: View Application Details  2: View Personal Details  3: Change Application Details  4: Change Personal Details  5: Quit | Course Name  Course Number  1: View Application Details  2: View Personal Details  3: Change Application Details  4: Change Personal Details  5: Quit |
| 1: View Application Details  2: View Personal Details  3: Change Application Details  4: Change Personal Details  5: Quit | 4 | Enter 1 to Change Name  Enter 2 to Change ID | Enter 1 to Change Name  Enter 2 to Change ID |
| Enter 1 to Change Name  Enter 2 to Change ID | 3 | ERROR: 1-2 ONLY  Enter 1 to Change Name  Enter 2 to Change ID | ERROR: 1-2 ONLY  Enter 1 to Change Name  Enter 2 to Change ID |
| Enter 1 to Change Name  Enter 2 to Change ID | 1 | Enter Name | Enter Name |
| Enter Name | Leon | Student Name  Student ID  1: View Application Details  2: View Personal Details  3: Change Application Details  4: Change Personal Details  5: Quit | Student Name  Student ID  1: View Application Details  2: View Personal Details  3: Change Application Details  4: Change Personal Details  5: Quit |
| Enter 1 to Change Name  Enter 2 to Change ID | 2 | Enter ID | Enter ID |
| Enter ID | 471 | Student Name  Student ID  1: View Application Details  2: View Personal Details  3: Change Application Details  4: Change Personal Details  5: Quit | Student Name  Student ID  1: View Application Details  2: View Personal Details  3: Change Application Details  4: Change Personal Details  5: Quit |
| 1: View Application Details  2: View Personal Details  3: Change Application Details  4: Change Personal Details  5: Quit | 5 | Enter Choice:  1: View Course/Personal Details  2: Edit Course/Personal Details  3: Quit | Enter Choice:  1: View Course/Personal Details  2: Edit Course/Personal Details  3: Quit |
| Enter Choice:  1: View Course/Personal Details  2: Edit Course/Personal Details  3: Quit | 3 | File Updated  Program End | File Updated  Program End |

**Conclusion**

Overall, the program works as expected. Early on in development of this assignment, problems occurred with checking if the ID that the user entered was in the array, no matter what ID was entered, the system would always display the first person in the array. This was fixed by changing the for loop to have i<numstudents instead of having i<arraystudents.length and using a subscript to hold onto the value of i throughout the rest of the program.

The test bed results match the outcomes expected of this program.