|  |
| --- |
| City of Glasgow College |
| Technical Manual |
| OOP: Centralia College Project |
|  |
| **Niall Ferguson** |
| **4/19/2013** |

|  |
| --- |
|  |

Contents

[1. Main Menu 2](#_Toc354751766)

[2. Adding New Student 3](#_Toc354751767)

[3. Deleting Student 5](#_Toc354751768)

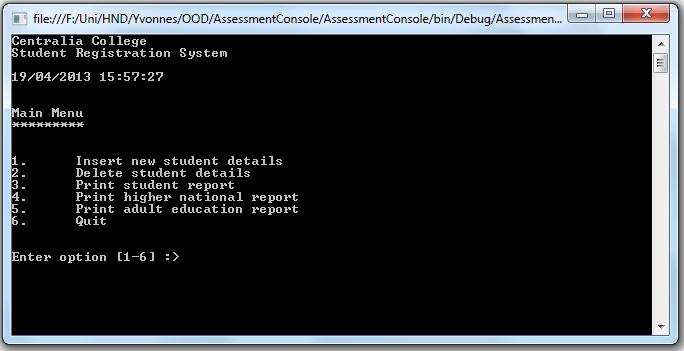
[4. Printed report of all Students 6](#_Toc354751769)

[5. Report of all Higher National Students 7](#_Toc354751770)

[6. Report of all Adult Students 8](#_Toc354751771)

[7. Quit 9](#_Toc354751772)

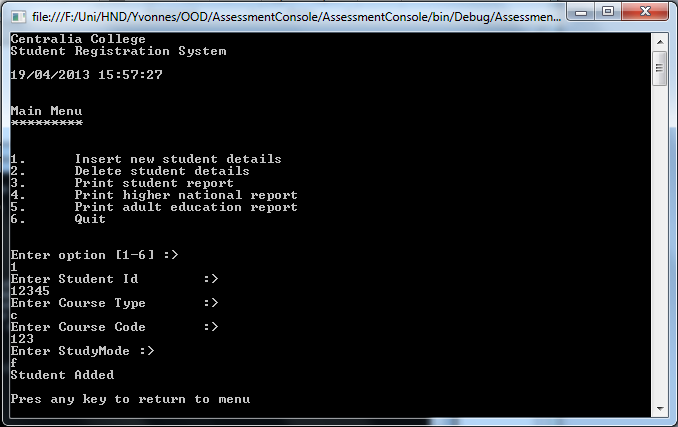
## Main Menu



* 1. On loading of application Main() is called
     1. Method creates new instance of College class called college1
     2. FileManager Class calls readstudent() and passes in college1object
        1. Readstudent() opens file stream reader and takes in students details from the CSV file called stdtList.txt. This file is parsed and Student information is added to the properties of either an instance of AdultStudent Class or HNStudent Class depending on the value of coursecode. These HNStudent objects or AdultStudent objects are added to a dictionary property in the college1 object.
     3. The College1 mainMenu() is called
        1. mainMenu method writes to console the menu options displayed to the user. It then waits for user input from 1-6. A switch statement will call the relevant method as chosen by the user.

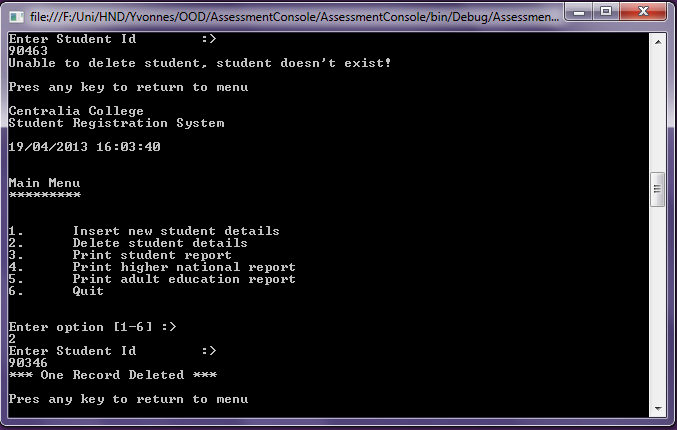
## Adding New Student

* 1. Switch statement of the mainMenu() calls addStudentMenu() if input is 1
     1. addStudentMenu() writes to console prompt for user to enter Student ID. Input is passed to string variable menuStudentId. menuStudentId is checked to see if it already exists in the students dictionary by using the ContainsKey(). If it does exist user is informed that “Unable to add student, studend already exists” and prompted to press any key to return to main menu. Else user is prompted to enter Course Type and input is passed to string variable menuCourseType. User is then prompted to enter Course Code and the input is parsed and passed to int variable coursecode. If coursecode is greater than 100 user is prompted to input StudyMode and input is passed to string variable studyMode. A new instance of HNStudent called HNStudent1 is created by passing menuStudentId, menuCourseType, coursecode and studymode to the Class constructor. addStudent() is called and HNStudent1 object is passed to it.
        1. addStudent() takes in student object and checks the studentId property against the students dictionary key list. If it doesn’t exist already the StudentId property is used as the Key and the student object as the value and is added to the dictionary using the add(). addStudent returns Boolean variable as true, else returns it as false. If Boolean return is true message displayed is “Student Added”.
     2. If coursecode is less than 100 a new instance of AdultStudent called AdultStudent1 is created by passing menuStudentId, menuCourseType, coursecode and studymode to the Class constructor. setFee() is called
        1. setFee() takes in courseType and using a switch statement sets the property fee accordingly.
        2. addStudent() takes in student object and checks the studentId property against the students dictionary key list. If it doesn’t exist already the StudentId property is used as the Key and the student object as the value and is added to the dictionary using the add(). addStudent returns Boolean variable as true, else returns it as false. If Boolean return is true message displayed is “Student Added”.
  2. User is prompted to press any key to return to menu. mainMenu() is called.



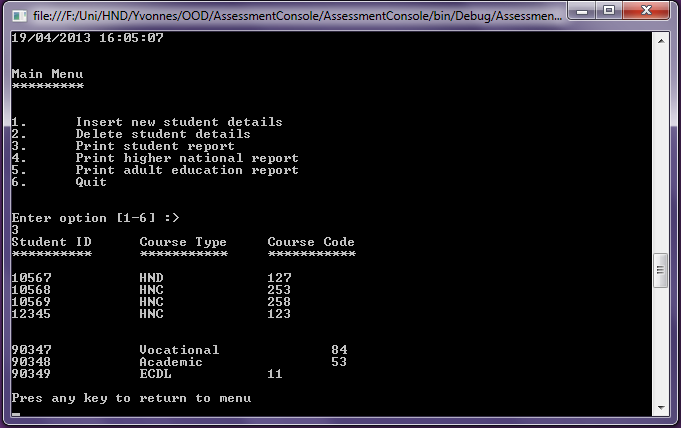
## Deleting Student

* 1. Switch statement of the mainMenu() calls deleteMenu() if input is 2
     1. deleteMenu() writes to console prompt for user to enter studentId. Input is passed to string variable menuStudentId. deleteStudent() is called and menuStudentId is passed to it.
        1. deleteStudent() checks the studentId against the key list of the students dictionary. If the key does exist the Remove() is called with studentKey as parameter. deleteStudent() returns Boolean true. Else returns Boolean false.
     2. If deleteStudent() returns true message display to user "\*\*\* One Record Deleted \*\*\*" and user is prompted to press any key to return to menu. If returns false message display to user "Unable to delete student, student doesn't exist!" and user is prompted to press any key to return to menu.
  2. mainMenu() is called.



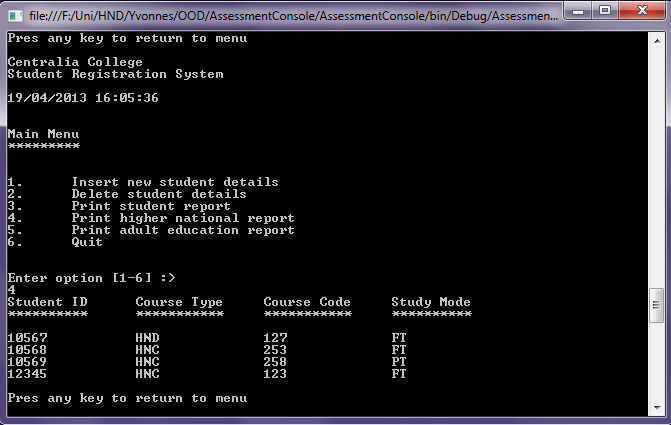
## Printed report of all Students

* 1. Switch statement of the mainMenu() calls printStudents() if input is 3
     1. printStudents() writes to console table headers.
     2. Iterates through all student objects in students dictionary
        1. If student is of type HNStudent calls student’s superclass toString()
           1. toString() returns as a string - the student ID, courseTypeFull and courseCode properties of the Student Object
        2. Writes the returned string to console
     3. Iterates through all student objects in students dictionary
        1. If student is of type AdultStudent calls student’s superclass toString()
           1. toString() returns as a string - the student ID, courseTypeFull and courseCode properties of the Student Object
        2. Writes the returned string to console
     4. Prompts user to press any key to return to main menu
     5. mainMenu() is called



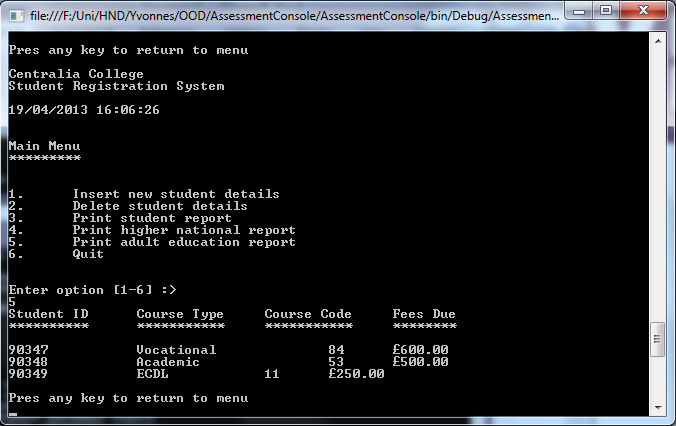
## Report of all Higher National Students

* 1. Switch statement of the mainMenu() calls printHNStudents() if input is 4
     1. printHNStudents() writes to console table headers.
     2. Iterates through all student objects in students dictionary
        1. If student is of type HNStudent student is cast as HNStudent and calls student’s toString()
           1. toString() returns as a string - the student ID, courseTypeFull and courseCode and modeFull properties of the Student Object
        2. Writes the returned string to console
     3. Prompts user to press any key to return to main menu
     4. mainMenu() is called



## Report of all Adult Students

* 1. Switch statement of the mainMenu() calls printAdultStudents() if input is 5
     1. printAdultStudents() writes to console table headers.
     2. Iterates through all student objects in students dictionary
        1. If student is of type AdultStudent student is cast as AdultStudent calls student’s toString()
           1. toString() returns as a string - the student ID, courseTypeFull and courseCode and fee properties of the Student Object
        2. Writes the returned string to console
     3. Prompts user to press any key to return to main menu
     4. mainMenu() is called



## Quit

* 1. Switch statement of the mainMenu() calls FileManger.writeStudent() if input is 6
     1. writeStudent() takes in students dictionary
        1. opens up streamwriter with students.txt as its destination file
           1. iterates through student objects in dictionary and calls Write() to write to file the student properties StudenId, CourseType and CourseCode.
           2. If the student is of type AdultStudent call Write() and write AdultStudent property Fee. Else call Write() and write HNStudent property StudyMode
        2. Close streamwriter
     2. Environment method Exit() called and programme quits returning user to operating system.