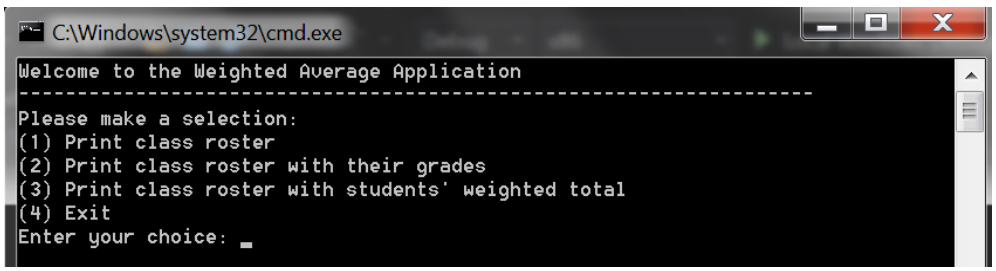
Weighted Average Score

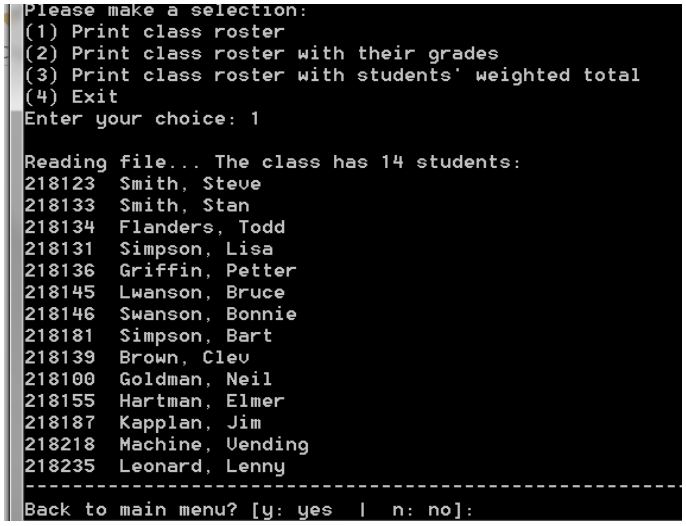
The assignment problem:

You are asked to develop an application with a main goal to calculate the weighted average scores of the students enrolled in 201R class. Your application will read input from two files. First file is called "INFO.txt" and it includes a list of current students with their student IDs and first and last names. Second file is called "Grades.txt" and it includes a list of student IDs and their grades in this course. The grades are listed in the following order "quiz1 quiz2 midterm Final." (quizzes are graded out of 10 points each, exams are graded out of 100 points each) Note that not all students have grades (some are auditing). Also note that the students' IDs in the second file are not necessary consistent (in the same order) with ID's in the first file. Your application starts by providing the following menu to the user:



The main menu options are as follows:

1. **Print Class Roster:** this option reads the data from the first file and prints the students' IDs and full names (Last name, First name) –see following figure:



After each menu selection, your application must ask the user if he wants to go back to the main menu or not. If the user inputs "y" for "yes", then your application must print the main menu again.

2. **Print Class Roster with Grades:** this option will (1) read first file to get students IDs and their names, (2) read the second file to get students IDs with the grades, and (3) print a list of students (IDs, names, and grades) for those who have grads. See following figure:

```
Enter your choice: 2

Reading the grades file...
Simpson, Lisa:  10 10 98 99
Griffin, Petter:        2 2 7 2
Brown, Clev:    10 10 80 90
Kapplan, Jim:   10 5 95 85
---------------------------------------
Back to main menu? [y: yes  |  n: no]:
```

3. **Print Class Roster with Weighted Grades:** this option will find the weighted averages of those students who have grades and print them on the screen. See the following figure:

```
-----------------------------------------------------------------
Back to main menu? [y: yes  |  n: no]: y
-----------------------------------------------------------------
Please make a selection:
(1) Print class roster
(2) Print class roster with their grades
(3) Print class roster with students' weighted total
(4) Exit
Enter your choice: 3
Weighted average score of       Simpson, Lisa is        99    out of 100%
Weighted average score of       Griffin, Petter is      7.75    out of 100%
Weighted average score of       Brown, Clev is  90    out of 100%
Weighted average score of       Kapplan, Jim is 85    out of 100%
-----------------------------------------------------------------
Back to main menu? [y: yes  |  n: no]: _
```

4. **Exit:** exists the application

Development Instructions

Your application will implement its functionalities using a **class** called "Student" which has the following:

- A constructor that takes parameters for the Student ID and student name. The student ID must be bigger than 0, otherwise the student ID must be set to -1
- Getters and Setters for the student ID, First name, Last name, and others
- A function called *FullName()* that returns a string consisting of the last name, a comma and space, then the first name; a Student with first name "A" and last name "B" would return "B, A" from this function
- A method called *ReadData()*. It takes an *istream* passed by reference as its only parameter. It reads in an integer (Student ID), first, and last name (in this order). This method returns a boolean value: true if all data was read successfully, false otherwise. Use this member function to read data from the first file.
- A method called *FindAverage()*that returns the weighted score of the students. It first calculates the average of the two quizzes (Hint: each quiz is worth 10 points maximum, so you might want to consider converting the quizzes to 100 point scale by multiplying each quiz grade by 10 and then finding their average -mean). Then, the function calculates the weighted score of the: quizzes average, midterm exam, and final exam. The weight of these exams are: quizzes 25%, midterm 25%, and final 50%.

Submission:

- Your application must be implemented as a OOP (uses a class, header file, and a main program). Otherwise, it will receive no score.
- Zip up your entire project folder and submit the zip file to Blackboard by the deadline.