CS102L Project Report

**Spring 2018**

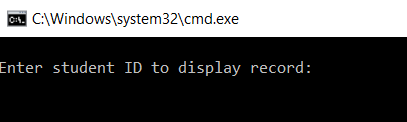
**Section:** C

**Submitted By:** Saim Ul Hassan, 2017404 | Shehr Yar, 2017434

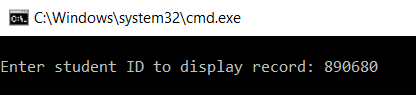
**Date:** May 6 2018

**Operation:**

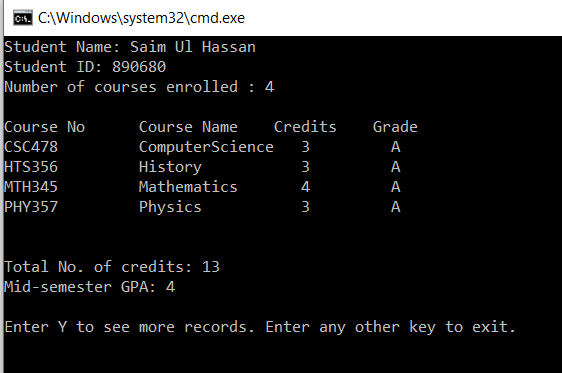
1. Upon running the program, a prompt to enter a Student ID will be displayed.



1. Type in a Student ID and press Enter key. If the ID exists in the record file, the student grade report will be displayed, otherwise an error message will be displayed with another prompt to enter ID.



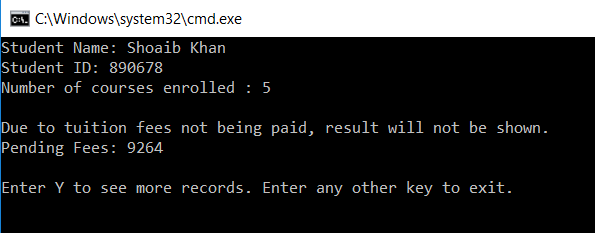
1. The grade report should be displayed after entering valid ID:



The program will prompt the user to enter Y to print a grade report for another student. Typing Y and pressing Enter will clear the screen and display another prompt to enter an ID. Entering any other key will terminate the program.



1. If the student whose ID is entered has not paid the tuition Fees, their grades will not be shown. Instead a message with their pending fees will be shown.



# Functionality

Libraries Included:

**iostream:** For basic cin/cout operations

**fstream:** For file handling operations

**string:** For string

**iomanip:** For formatting output of cout

In order to store the student information in the program, a structure for containing course data and a class for student data is defined. Their member variables include:

**Course** (**Structure**): **Course Name, No, Credit Hours** and **Grade**.

**Student** (**Class**): **Student Name, Student ID, No. of Enrolled Courses**, **tuition paid or not**, and an **array of Course structures** to hold course data for all courses.

The member functions of the class include:

**bool read (int ID):** This function opens the record file and systematically readseach student record from the file and stores it in the class variables until the ID read from the file matches the ID given in the function argument. A while loop is implemented to read each record one by one and then comparing the ID value read from the file to the one passed during function call. If they match, the file is closed and the function **returns true** to signify that a record with the given ID does exist in the file. If such a record is not found, the function **returns** **false.**

In the function, memory is dynamically allocated for the Course array in order to ensure that only as much memory is occupied as needed to hold the data. This memory is then released in the class’ Destructor.

**double GPA ():** Calculates and returns the GPA for the relevant student using the standard formula. Called by the display function. The **gradeValue** is called in this function to convert the alphabetical grades into a numerical value.

**double feesAmount (double tuition):** Calculates and returns the total fees amount for the relevant student.

**void display (double tuition):** Displays the student information and grade report with appropriate formatting. If student has paid tuition, a BubbleSort function is called to sort the Course array according to the Course No. and display grade report showing all grades with the Total No. of Credits and Mid-Semester GPA. If tuition is not paid, only student info is shown and a message along with Pending Fees is displayed.

Functions outside the class include:

**void manager (double tRate):** Function creates a Student object and takes a student ID as input from the user. It then calls the read function through the Student object with the user’s input. If the function returns false, the user is prompted to enter a different ID. If the function returns true, the display function is then called as well as a prompt for the user to enter a different ID to see another student’s information. A while loop keeps taking input until the user enters a terminal value.

**string NameSpace (string name):** Function takes in a string from the Student Class-type object. The string passed in the argument will be in CamelCase and has to be converted into a regular string with spaces for displaying output to the user. A new string is created and each letter from the passed string is added to the new string with a space added before each capital letter. The final string is then returned.

**int gradeValue (char grade):** As the grade read from the file is in character format, it needs to be converted into a numerical value for calculating the GPA in the class. This function takes in that character grade and using a switch selection statement, returns the numerical value associated with the letter grade.

**void BubbleSort (Course c[], int size):** A simple Bubble Sort algorithm is used for sorting the courses array in the Student Class object in ascending order according to the course no. Consecutive elements of the array are compared inside a ***for loop*** and if the earlier element’s course no is bigger than the later element’s, they are swapped. This comparison is carried out for the entire array multiple times inside an outer ***while loop*** until not a single swapping operation takes place over the entire array, meaning the array is fully sorted.

# Execution

In the **main** function, first the record file is opened and the total No. of Students and Tuition Rate is read from the first line of the file. Then the file is closed and the ***manager*** function is called. In the Manager function, an object of class Student is created and a student ID is taken as input from the user inside of a while loop that will ensure further input is taken if a wrong ID is entered or if the user wishes to keep viewing records. This ID is passed to the function ***read*** called through the Student object. This function reads data sequentially from the file and compares the ID of each record from the file to the ID passed into the function. If they match, the function returns true otherwise it returns false if the end of file is reached. The read function also calls the ***NameSpace*** function to convert the name read from the file from CamelCase to regular spaced form.

If the read function returns true, the ***display*** function is then called through the Student object which displays the student information and the grade report if the student had paid tuition, otherwise the grade report is omitted and a message is displayed with the Pending Fee amount.

The display function calls the ***GPA*** and the ***feeAmount*** functions to calculate the Grade Point Average and total pending fee. It also calls the ***BubbleSort*** function to sort the courses array in order of course no. The ***setw*** function is used to help with formatting the output of course information in tabular form.

After the display function is finished, a prompt is displayed to the user to enter the letter ‘Y’ to continue displaying records of more students. If the user enters Y, the while loop in the manager function runs again and the prompt to enter student ID is displayed again. If the user enters any other character, the program terminates.