

TCES 203
Programming Practicum
Assignment 1 – C++ Basics
15 Points

This assignment tests your understanding of concepts covered in the first six chapters of the course dealing with C++ Language, Expressions and Decision Structures, Loops and Functions. You are to write a program that prompts the user for information about two applicants and that computes an overall score for each applicant. This is a simplified version of a program that might be used for admissions purposes.

For each applicant, we prompt for exam scores (either SAT or ACT) and actual GPA. The exam information is turned into a number between 0 and 100 and the GPA information is turned into a number between 0 and 100 and these two scores are added together to get an overall score between 0 and 200. After obtaining scores for each applicant, the program reports which one looks better or whether they look equal.

Notice that the program asks for each applicant whether to enter SAT scores or ACT scores (SAT scores are integers that vary between 200 and 800, ACT scores are integers that vary between 1 and 36). In the case of SAT scores, the user is prompted for SAT verbal and SAT math subscores. In the case of ACT scores, the user is prompted for English, math, reading and science subscores. These scores are turned into a number between 0 and 100 using the following formulas:

$$\text{For SAT Scores: } \frac{2 \cdot \text{verbal} + \text{math}}{24}$$

$$\text{For ACT Scores: } \frac{2 \cdot \text{reading} + \text{English} + \text{math} + \text{science}}{1.8}$$

These formulas produce numbers in the range of 0 to 100. After computing this exam score, we compute a number between 0 and 100 based on the GPA. You will notice that the program prompts for the GPA and the maximum GPA. Both the GPA and maximum GPA are real values (i.e., they can have a decimal part). You should turn this into a score between 0 and 100 using the following formula:

$$\frac{\text{actual_gpa}}{\text{max_gpa}} \cdot 100$$

At this point your program has two scores that vary from 0 to 100, one from their test score and one from their GPA. The overall score for the applicant is computed as the sum of these two numbers (exam result + gpa result). Because each of these numbers is between 0 and 100, the overall score for an applicant ranges from 0 to 200.

Your program does not need to perform any error checking. Assume that the user will input the right values.

In terms of program style you should use functions to eliminate redundant code and to break the problem up into logical subtasks. Your main function should be short so that a person can easily see the overall structure of the program. You are to introduce at least five functions other than main to break this problem up into smaller subtasks and you should make sure that no single function is doing too much work. In this program, none of your functions should have more than 15 lines of code in the body of the function (not counting blank lines or lines with just curly braces on them). Be sure to once again include a short comment at the beginning of your program as well as a short comment for each function describing what it does. Also remember that because this program involves both integer data and real data, you need to use appropriate type declarations (type int for integer data, type float for real-valued data).

Use C++ functions that you learned in this course as much as possible. Your functions must be declared above main. In other words, provide function prototypes.

First log of execution (user input underlined)

This program compares two applicants to determine which one seems like the stronger applicant. For each candidate I will need either SAT or ACT scores plus a weighted GPA.

Information for the first applicant:

```
do you have 1) SAT scores or 2) ACT scores? 1
SAT math? 450
SAT verbal? 530
actual GPA? 3.4
max GPA? 4.0
```

Information for the second applicant:

```
do you have 1) SAT scores or 2) ACT scores? 2
ACT English? 25
ACT math? 20
ACT reading? 18
ACT science? 15
actual GPA? 3.3
max GPA? 4.0
```

```
First applicant overall score = 147.917
Second applicant overall score = 135.833
The first applicant seems to be better
```

Second log of execution (user input underlined)

This program compares two applicants to determine which one seems like the stronger applicant. For each candidate I will need either SAT or ACT scores plus a weighted GPA.

Information for the first applicant:

do you have 1) SAT scores or 2) ACT scores? 2
ACT English? 20
ACT math? 19
ACT reading? 21
ACT science? 30
actual GPA? 3.5
max GPA? 4.0

Information for the second applicant:

do you have 1) SAT scores or 2) ACT scores? 1
SAT math? 610
SAT verbal? 640
actual GPA? 4.3
max GPA? 5.0

First applicant overall score = 149.167

Second applicant overall score = 164.75

The second applicant seems to be better

Third log of execution (user input underlined)

This program compares two applicants to determine which one seems like the stronger applicant. For each candidate I will need either SAT or ACT scores plus a weighted GPA.

Information for the first applicant:

do you have 1) SAT scores or 2) ACT scores? 1
SAT math? 510
SAT verbal? 530
actual GPA? 3.4
max GPA? 4.0

Information for the second applicant:

do you have 1) SAT scores or 2) ACT scores? 1
SAT math? 570
SAT verbal? 500
actual GPA? 3.4
max GPA? 4.0

First applicant overall score = 150.417

Second applicant overall score = 150.417

The two applicants seem to be equal

Submission and Grading:

Submit admit.cpp under the Assignments section of the course web page.

Your outputs must *exactly* match the ones above and shouldn't include more tests or fewer tests. You should definitely test your code for other scenarios but those tests should not be included in submission.

There will be points taken off for not following the conventions listed in this document regarding submissions, outputs and naming conventions.

You are required to properly indent your code and will lose points if you make significant indentation mistakes. See the textbook for an explanation and examples of proper indentation.

Give meaningful names to functions and variables in your code. Localize variables whenever possible -- that is, declare them in the smallest scope in which they are needed.

Include a comment at the beginning of your program with basic information and a description of the program **and include a comment at the start of each function**. Your comments should be written in your own words and not taken directly from this document. Write comments within functions to explain the flow or any obscure code.

You should include a comment at the beginning of your program with some basic information and a description of the program, as in:

```
// Menaka Abraham
// 3/30/15
// 203
// Assignment #1
//
// This program will...
```

You should name your file **admit.cpp** and you should turn it in electronically on the class web page.