Assignment 5

Reading and Exercises

Read the online class notes Lesson 5 and Chapter 4, pages 227-234

Questions

1) A tricky question:

Let's say int n = 7. What is the effect of:

a)
```
if ( n = = 3)
        cout << "Hello";
```

b) if (n = 3)

cout << "Hello" ;

Why ? (Hint: the assignment operator on native C++ types returns the value being assigned.)

2) When you call a function, as in the area and perimeter of a past assignment, you include the values to be sent to the function (the arguments) in parentheses. The function must have been defined with parameters of an equivalent type, that will be used to receive those values.

If the function is called using variables, as in cout << area(length, width) for instance, what is passed to the parameters is a copy of the values in those variables. This is called *call-by-value.* In that case, the receiving parameters in the function are initialized to the values of the two argument variables (length and width, in our example). Based on this, given the following function:

```
void f (int i )
{   i =57;
}
```

assuming it is called like this:

f(value) ; where *value* is a variable of type int, defined as int value = 9; what would be the result of printing *variable* value after the call? Would *value* change? **Explain**

3) Change the binary number 10101101 into its decimal representation

4) Based on the sections we covered on output formatting, the output statement

```
cout << left << setw(10) << "Tables" << endl;
```

should display the string “Tables” followed by how many space characters? Explain your answer.

C++ Program :

Write a program that accepts the following input from the keyboard:

-a first name

-a last Name

-The scores on three tests (scores will range from 0 to 10). They type should be *double*

The program will calculate the average of the three scores entered from the keyboard and store the average as a **rounded integer** (ex. If the average is 7.57 then it will be stored as 8, but if it was 6.49, then it will be stored as 6)

Then, using the rounded average calculate the corresponding course letter grade using the scoring system below. **You must use** a *switch* statement to get the letter grade.

The simplified scoring system will be:

9-10 A

8 B

7 B

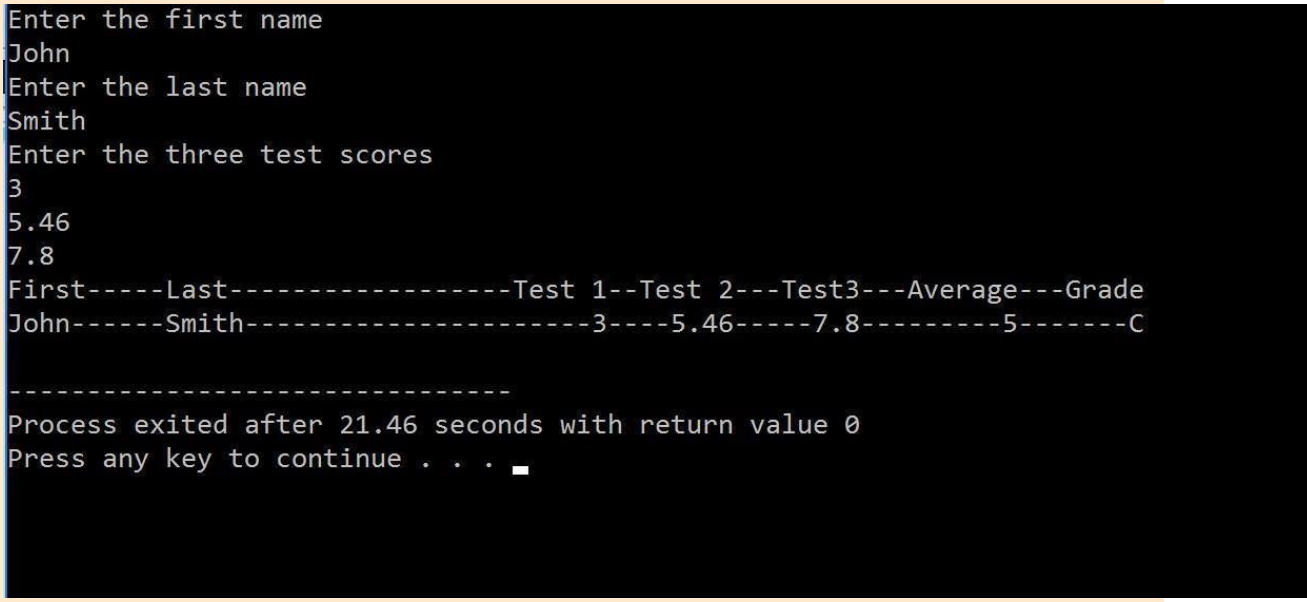
6 C+

5 C

4 D+

0-3 F

Once all of the data has been processed (entered, stored and average and letter grades have computed) display on your monitor (cout) the results in the following way:



The dashes were produced by adding *setfill('-')* to the output.

Once you are satisfied that your output is correct and properly lined up, send the same output to a file. Use "example.txt" as the file name. The file will be created in the same directory as your program.

To find more about how to output to files, review the File Input Output posted in Lesson 4 on Blacbord or read the textbook pages 160-163 (File input output) There you will see how to create a file for output (and input, but this time we will do only output). Once you have created the file stream associated with file *example.txt*, you can output to that file stream the same as you do to *cout*. In fact, you can think of *cout* as a file stream associated with your monitor, instead with a real file on disk.

In order to make all the above work you need to include:

#include <iostream>

#include <fstream>

#include <string>

#include <iomanip>

using namespace std;

You can view your output file from within Visual Studio by going to the File menu entry, clicking on Open and then on the File option. This will open the folder where your file will be saved by default. Look for *example.txt*, as that is the file name I need you to use when you create/open the file.

It is important that you name the output file as I indicated (“example.txt”), since when I run your program I will expect it to create a file of the same name on my computer that I will have to check to make sure your output is correct.