Homework 2: Basic C++

CS16 - Winter 2021

Due:	Thursday, January 21, 2021 (11:59 PM PST)
Points:	60
Name:	Tyler_Pruitt______________________________________
Homework buddy:	____

- You may collaborate on this homework with **at most** one person, an optional "homework buddy."
- **Submission instructions:** All questions are to be written (either by hand or typed) *in the provided spaces* and turned in as a single PDF on Gradescope. If you submit handwritten solutions write legibly. We reserve the right to give 0 points to answers we cannot read. When you submit your answer on Gradescope, **be sure to select which portions of your answer correspond to which problem** and clearly mark on the page itself which problem you are answering. We reserve the right to give 0 points to submissions that fail to do this.

1. (3 points) Is this variable declaration OK to do in C++? Why or why not?

```
double const = 30;
```

No this variable declaration is not OK because "const" is a reserved keyword in C++.

2. (4 points) Show 2 different ways to initialize variables in C++.

```
int number = 3;
int number(3);
These two ways are equivalent.
```

3. (12 points) Mark up this code to show (at least) 4 things that are missing.

Marked up code:

```
#include <iostream>
// missing <string> library
// missing namespace std;

int main() // missing first { for main() function
    int a(0), b(0), c(0);
    string quote;
    cout << "Enter 3 numbers separated by spaces ";
    cin >> a, b, c; // cannot assign variables a,b,c this way, need >> not commas
    sum = a + b + c; // missing initialization for sum
    quote = "The sum of these 3 numbers is " // missing ; at end of statement
    cout << quote << sum;
    return 0;
}
```

Correct code:

```
#include <iostream>
#include <string> // EDIT 1
using namespace std; // EDIT 2

int main() { // EDIT 3
    int a(0), b(0), c(0);
    string quote;
    cout << "Enter 3 numbers separated by spaces ";
    cin >> a >> b >> c; // EDIT 4
    int sum; // EDIT 5
    sum = a + b + c;
    quote = "The sum of these 3 numbers is "; // EDIT 6
    cout << quote << sum;
    return 0;
}
```

4. (15 points) The program below intends to do the following: Repeatedly prompt the user to input an integer number. When the user no longer wants to continue, output the sum of all the positive numbers, followed by the sum of all the negative numbers. However, the given program has errors. Mark all logical and syntax errors in the program and provide corrections in the space provided to the right. Add missing statements if any.

Marked up code:

```
#include <iostream>
// missing <string> library
using namespace std;

int main() {

    int a, sumPositive, sumNegative; // sumPositive, sumNegative should be set to 0

    string promptContinue = "\nTo continue, enter Y/y\n";

    string promptNum = "\nEnter a number: ";

    char response; // set response to 'y' initially

    while (reponse = 'y' || 'Y') { // logic error: need response == 'Y'
    //syntax error: response is spelled wrong

        cout << promptNum;

        cin >> a;

        if (a) // logic error: need (a>0)
            sumPositive += a;
        else
            sumNegative += a;

        cout << promptContinue;

        // need input for response again
    }

    cout << "Sum of all positive numbers: " << sumPositive << endl;

    cout << "Sum of all negative numbers: " << sumNegative << endl;

    return 0;
}
```

Correct code:

```
#include <iostream>
#include <string> // EDIT 1
using namespace std;

int main() {

    int a, sumPositive(0), sumNegative(0); // EDIT 2

    string promptContinue = "\nTo continue, enter Y/y\n";

    string promptNum = "\nEnter a number: ";

    char response = ’y’; // EDIT 3

    while (response == ’y’ || response == ’Y’) { // EDIT 4

        cout << promptNum;

        cin >> a;

        if (a>0) // EDIT 5
            sumPositive += a;
        else
            sumNegative += a;

        cout << promptContinue;

        cin >> response; // EDIT 6


    }

    cout << "Sum of all positive numbers: " << sumPositive << endl;

    cout << "Sum of all negative numbers: " << sumNegative << endl;

    return 0;
}
```

5. (4 points) What is the resulting output from the following C++ statements? Explain why.

```
int x(35), y(5);
bool v, w;
v = (x < y * y);
w = ((x/y) == 7);
cout << (v && w) << endl;
```

```
The output is 0 because v is assigned as (false * 5) = (0 * 5) = 0, so v is
false and w is true because 35/5 is 7. Since in C++ booleans are converted
into 0 for false and 1 for true, (v && w) is false because v is false so the
value of (v && w) is converted into 0 and then printed out to terminal.
```

6. (6 points) Write code with **one** `while` loop that prints out these 5 lines like this (remember a newline after each line):

```
COUNTDOWN TO ZERO: 4
COUNTDOWN TO ZERO: 3
COUNTDOWN TO ZERO: 2
COUNTDOWN TO ZERO: 1
COUNTDOWN TO ZERO: LIFT OFF!
```

```
#include <iostream>
using namespace std;

int main()
{
    int iteration = 4;
    while (iteration>=0)
    {
        if (iteration > 0)
        {
            cout << "COUNTDOWN TO ZERO: " << iteration << endl;
        }
        else
        {
            cout << "COUNTDOWN TO ZERO: LIFT OFF!" << endl;
        }
        iteration--;
    }

    return 0;
}
```

7. (4 points) What is the exact output of the following statements?

```
int s = 1;
do {
    cout << s << "+";
}
while (s++ < 5);
```

```
                      Output is: 1+2+3+4+5+
```

8. (4 points) Same question as (7.) but the last statement now reads:

```
while (++s <= 5);
```

```
                      Output is: 1+2+3+4+5+
```

9. (4 points) Explain via an example what a "type mismatch" is. Also, explain how compilers handle C++ statements that have a type mismatch.

A type mismatch is when the types of two objects are not in agreement. For example, this occurs in the following code.

```
char letter;
letter = "hello";
```

In this example, the character letter is trying to be assigned as the string "hello" which is a type mismatch because the types of the variable and the value are different and incompatible.

Different compilers react to these type mismatches differently and the same compiler might differ on a case-by-case basis. Some compilers will throw an error message, some will throw a warning message, and some compilers will not complain whatsoever. Compilers may allow the user to mix types such as int and double or char and int but these are not recommended. Even if a compiler does not complain for a type mismatch, it is not wise nor best practice to let these assignments occur.

10. (4 points) Write an if-else statement that outputs the string "Grade is B" if the variable `score` is between 80 and 90 (both limits included). Otherwise, the if-else statement should output "Grade is not B".

```
if (score >= 80 && score <= 90)
{
    cout << "Grade is B" << endl;
}
else
{
    cout << "Grade is not B" << endl;
}
```