Homework 1: Intro to C++

CS16 - Winter 2021

Due:	Tuesday, January 14, 2021 (11:59 PM PST)
Points:	100
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.

1. (5 points) Not including any comments that may appear, what are the first two lines that typically begin a C++ program that is either going to output on the screen and/or read input from the keyboard?

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

2. (5 points) What statement is the recommended way to end a C++ program?

```
return 0;
```

3. (15 points) The textbook author describes the difference between **syntax errors** and **logic errors**, as well as the difference between compiler output that produces **error messages** vs **warning messages**. Briefly explain each of the items below in a way that makes the *differences* among them clear.

   a. (5 points) Syntax error that results in an *error* message:

This syntax error is a direct violation of the syntax/grammar rules of C++ and cannot be ignored. An example is missing a semicolon at the end of a statement.

b. (5 points) Syntax error that results in a *warning* message:

This syntax error occurs when something is typed that technically obeys the grammar/syntax rules of C++ but is interpreted by the compiler as not best practice and it think you should probably not put that statement there or it should be modified. In these cases, the compiler will throw out a warning message to make sure that this statement or operation is intended.

c. (5 points) Logic errors:

Logic errors are not errors of C++ syntax/grammar rules, but instead they are errors that allow your program to compile and run successfully but prevents the program from being correct (i.e. doing what the programmer wanted it to do). For example, a logic error will might prevent your program from outputting the correct answer to terminal, however this program will still be syntactically correct and approved by the compiler.

4. (5 points) Assuming the variable `age` has already been declared as `int age;` what single statement of code will read in a value for `age` from the user?

```
cin >> age;
```

5. (10 points) Assuming the variable `balance` has already been declared as `int balance;` write two code statements that will ask (prompt) the user for a value for `balance`, and then read in the value of `balance`.

```
cout << "Enter a value for balance: ";
cin >> balance;
```

6. (5 points) The textbook describes **C++11** on page 27. Briefly, what is C++11? (A one sentence answer is good enough.)

C++11 is the most recently approved version of the standard C++ programming language. Essentially, C++11 is the latest version of C++.

7. (10 points) The book talks about the 5 important components of a computer: (1) processor, (2) input devices, (3) output devices, (4) main memory, (5) secondary memory. It also talks about two important pieces of software: compilers and operating systems. What of the above is primarily responsible for each of the following tasks? Write "none" if none of the options apply.

   a. (2 points) Executes a program stored in main memory.

   `processor`

   b. (2 points) Allocates the computer's resources to different tasks.

   `operating system`

   c. (2 points) Stores a program while it is being executed.

   `main memory`

   d. (2 points) Stores a program when it is not being executed.

   `secondary memory`

   e. (2 points) Converts a program written in a high-level language to another high-level language.

   `none`

8. (5 points) In one sentence, what is the role of a *compiler*?

```
A compiler's role is to convert high-level language (i.e. Python, C, C++,
Java, etc.) into machine language (consisting of 1s and 0s) that is
understandable and executable to the machine.
```

9. (5 points) What is *object code* (and how is it different from C++ code)?

```
Object code is the compiled and translated source code such
that the object code is in machine language and is executable. Object code
is different from C++ code in that it is in machine language (consisting of
1s and 0s) whereas C++ code is written in a high-level language.
```

10. (10 points) If the following statement were in a C++ program, what would it do?

```
cout >> "A penny saved";
```

This statement would raise an error, rendering the program unable to compile and thus the program will not run. The statement "A penny saved" WILL NOT be outputted to terminal.

11. (10 points) If the following statement were in a C++ program, what would it do?

```
cout << "Is a penny earned.";
```

This statement WILL output to terminal "Is a penny earned." without creating a new line after it.

12. (15 points) Complete this C++ program (as indicated by the comments) designed to calculate the area and circumference of a circle. The program gets the *diameter* parameter from the user and then prints out statements that say:

```
The area of this circle is: <RESULT HERE>
The circumference of this circle is: <RESULT HERE>
```

Notes: (1) In the output replace <RESULT HERE> with the appropriate results. (2) Use the C++ `const` keyword to declare a value for pi (π). (3) Your code must be syntactically correct (i.e. it should compile without error).

```
#include <iostream>
using namespace std;
const double pi = 3.14;
int main() {
    // declare the variables here
    double diameter, area, circumference;
    cin >> diameter;

    // calculate the results here
    area = pi*(diameter/2)*(diameter/2);
    circumference = 2*pi*(diameter/2);

    // print statements here
    cout << "The area of this circle is: " << area << endl;
    cout << "The circumference of this circle is: " << circumference << endl;

    // end program
    return 0;
}
```