CSCI 1300 CS1: Starting Computing
Fleming, Naidu, Quigley 2020
Homework 2
Due: Saturday, September 12, by 6 pm
(5 % bonus on the total score if submitted by 11:59 pm September 11)

Background

Input & Output

The ability to display information to the user and receive input from them is key to making your programs useful. In C++ one way we can achieve this exchange of information is using **cout** (console out) and **cin** (console in) from the iostream library. Whenever you use **cout** or **cin**, you will need to be sure to include the **iostream** (input output stream) library and use the **std** (standard) namespace.

Example Input (User input in bold)

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

int main(){
   int number1;
   int number2;
   cout << "Please enter the first number:" << endl;
   cin >> number1;
   cout << "Please enter the second number:" << endl;
   cin >> number2;
   cout << "Sum of numbers: " << number1 + number2 << endl;
}
```

```
Please enter the first number:
1
Please enter the second number:
2
Sum of numbers: 3
```

Data Types

When programming, we store the variables in our computer's memory, but the computer needs to know what kind of data we want to store in them, since it is not going to occupy the same amount of memory to store a simple number than to store a single letter or a large number, and they are not going to be interpreted the same way. Some commonly used data types in C++ are:

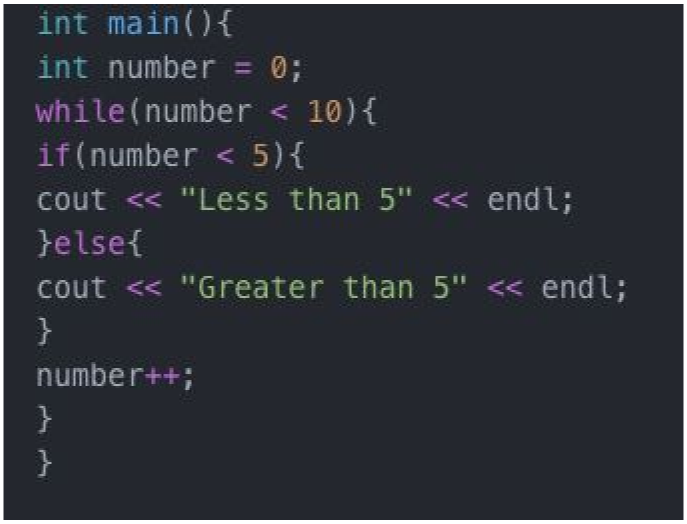
1. `int` (for integers)
   - `int myInt = 5;`
2. `char` (for characters)
   - `char myChar = 'c';`
3. `float` (for floating-point numbers)
   - `float myFloat = 4.4531;`
4. `double` (for double precision floating-point numbers)
   - `double myDouble = 4.4531;`
5. `string` (for strings of characters)
   - `string myString = “Hello World”`

Coding Style

Whitespace and indentation

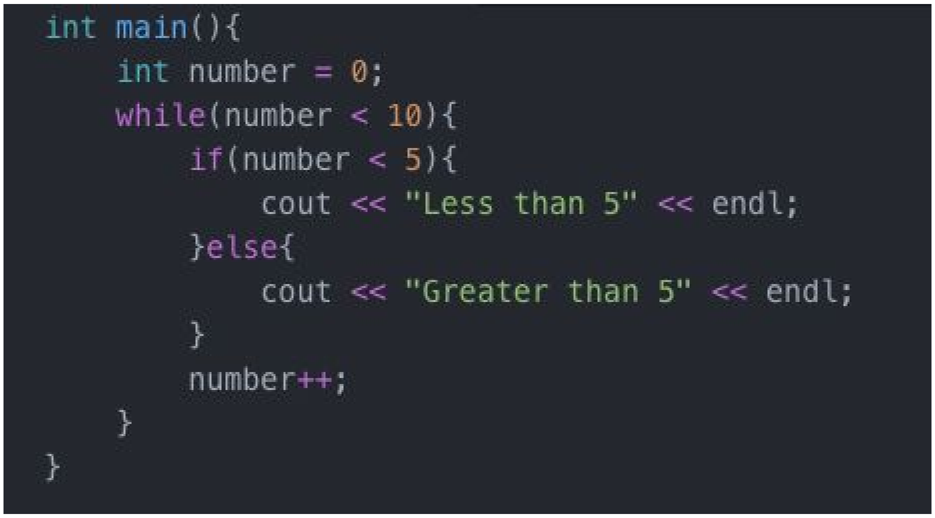
Indenting: increase your indentation by one increment of each brace {, and decrease it once on each closing brace }. Place a line break after every {. Use Tab to increase indent and Shit+Tab to decrease indentation.

Bad Example: There is no indentation and it is difficult to see where different portions of the code start and stop.



```
int main(){
int number = 0;
while(number < 10){
if(number < 5){
cout << "Less than 5" << endl;
}else{
cout << "Greater than 5" << endl;
}
number++;
}
}
```

Good Example: There is proper indentation and the code is easier to read.

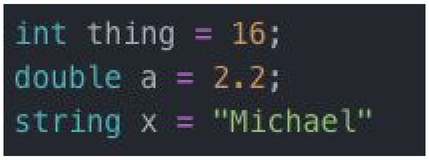


```
int main(){
    int number = 0;
    while(number < 10){
        if(number < 5){
            cout << "Less than 5" << endl;
        }else{
            cout << "Greater than 5" << endl;
        }
        number++;
    }
}
```

Naming and variables

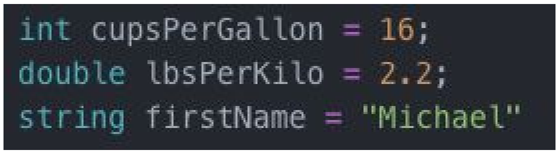
Names: Give variables descriptive names, such as firstName or homeworkScore. Avoid one-letter names like a or x, except for loop counter variables such as i.

Bad Example:



```
int thing = 16;
double a = 2.2;
string x = "Michael"
```

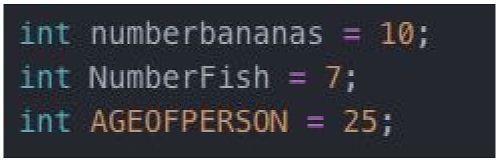
Good Example:



```
int cupsPerGallon = 16;
double lbsPerKilo = 2.2;
string firstName = "Michael"
```

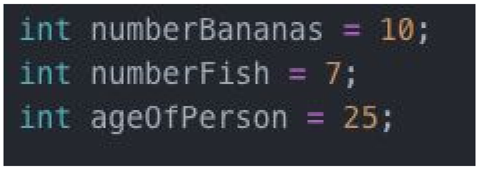
Capitalization: Name variables and functions with camel-casing, where the first letter is lowercase and all subsequent words in the name start with a capital letter.

Bad Example:



```
int numberbananas = 10;
int NumberFish = 7;
int AGEOFPERSON = 25;
```

Good Example:



```
int numberBananas = 10;
int numberFish = 7;
int ageOfPerson = 25;
```

Comments

- Your code should be well-commented. Use comments to explain what you are doing, especially if you have a complex section of code. These comments are intended to help other developers understand how your code works. These comments should begin with two backslashes (`//`) or the multi-line comments (`/* … comments here… */`) .
- Please also include a comment at the top of your solution with the following format:

```
// CS1300 Fall 2020
// Author: my name
// Recitation: 123 – Favorite TA
// Homework X - Problem # …
```

Example submission

```
// CS1300 Fall 2020
// Author: Punith Sandhu
// Recitation: 123 – Favorite TA
// Homework 101 - add two numbers

#include <iostream>
using namespace std;

int main(){

   // asks for two integer numbers
   int number1;
   int number2;
   cout << "Please enter the first number:" << endl;
   cin >> number1;
   cout << "Please enter the second number:" << endl;
   cin >> number2;

   // add numbers
   cout << "Sum of numbers: " << number1 + number2 << endl;
}
```