Homework #1 - C++ Intro

Name:

1 Intro

Welcome to the homework! Today we learned a little bit about a **computer language** called **C++**. The C++ language is very popular for writing games and scientific programs.

A **computer language** or **programming language** is a language that people use to talk to computers. Writing programs is called **programming** or **coding**, and a **programmer** or **coder** is a person who writes programs. Programmers write **source code**, also just called **code**, which is a list of steps for the computer to follow. Every program starts as source code, written by one or more programmers.

For the computer to understand our source code, it has to be converted into an even simpler form called **binary code**. Everything on the computer is stored as binary code. All your pictures, movies, and music...they are all stored as binary code that looks like this:

```
0011101011010101
```

Programs are just binary code too. The binary code that programs are made of is sometimes called **object code**. Each pattern of 1's and 0's in object code tells the computer to do something, like "add", "multiply", or "compare". To run our C++ programs, we have to first translate them into object code. We do this using a program called a **compiler** which takes our source code and converts it into object code for us. We did this today in class:

```
$ g++ hello.c -o hello
```

The compiler we used was called "g++", and it turns C++ source code into object code for us. The very first computers didn't have compilers, so you had to write your programs directly using object code. It was slow and difficult, and it was very easy to make mistakes.

After the compiler translates your source code into object code, it will save the object code in a file. This file is called an **executable** or sometimes just a **program**. On a Mac or Linux computer, you can run your programs like this:

```
$ ./hello
```

We also did this today in class. After we typed this into the terminal and hit enter, the computer printed out "Hello, world!".

2 Input and Output in C++

Most programs do more than just say "Hello, world!". To be really useful, a program should be able to take **input** (read data), and produce **output** (write data). In C++, we "read" and "write" data using special "streams". There is one stream for input, called "cin", and another stream for output called "cout". You can think of streams as being kind of like "pipes" that data can travel through. Have a look at the program below, which can take **input** from the computer's keyboard, and show **output** on the computer's screen:

```
// A demonstration of I/O

#include <iostream>
using namespace std;
int main(){
    int x = 0;
    cout << "Enter a number: ";
    cin >> x;
    cout << "Your number was: " << x << endl;
    return 0;
}
```

ioTest.cpp

When this program runs, anything the user types in will be sent to **cin**. You can get the data that was typed in using the >> **operator**. Operators are the part of a programming language that help us work with dada. Operators do things like compare numbers, add, and subtract.You can output (print out) data using the << operator. Look at the code above and pay attention to how << and >> are used. You can try running this program on your own computer.

3 Variables and Data Types

Most programs need to have a place to save the data they are working on. We do this using **variables**. You can think of variables as being like boxes where we can store the data our program is working on.

To the computer, all data we put into a variable looks the same: it is all just binary code. We need a way to tell the computer what kind of data is saved in our variables. We do that using **data types**. Whenever we create a new variable, we give it a **type**. Here are some common types in C++:

```
unsigned int a = 2; // A positive number without a "." like 36
int b = -3; // A number without a "." like -10 or 1205
float c = 2.5; // A number with a "." like 12.055
double d = 3.28; // Like a float, but can hold bigger numbers
char e = 'x'; // Holds letters, like 'a' or 'b'
string f = "Hello bob!"; // Groups of letters (words), like "Steven"
```

It is important to remember the limits of each data type. Trying to save a number that is the wrong type (or too big, or too small) inside a variable can make your program do surprising things. See an example of this on the next page.

```
// Demonstrates the concept of rollover (tested on 64-bit Intel machine)

#include <iostream>
#include <stdio.h>

using namespace std;

int main(){
    unsigned char x = 0;
    int y = 0;

    cout << "int has size " << sizeof(y) << " byte(s)" << endl;
    cout << "unsigned char has size " << sizeof(x) << " byte(s)" << endl;

    for (y = 0; y < 500; y++) {
        printf("y = %d, x = %u\n", y, x);
        x++;
    }
}
```

rollover.cpp

If you try running this on your computer, you'll get a surprising result: the variable y can count up to 500, but the variable x counts up to 255 and then goes back to zero! The reason that this happens is that "int" variables are bigger than "char" variables. The "char" variable can only hold binary numbers that have 8 digits in them, like "11111111" (255). If you try to put "100000000" (256) into a "char" variable, the computer throws away the extra "1" and just saves "00000000" (0) instead. This is why you must be careful to choose the right variable types!

4 If/else

Why is a computer smarter than a calculator? A calculator can only do math, but a computer can *make decisions*. In C++, your program can make decisions using "if" and "else".

```
// A demonstration of branching

#include <iostream>

using namespace std;

int main(){
    int x = 5;

    if (x > 1) {
        cout << "STUFF" << endl;
    }
    else {
        cout << "THINGS" << endl;
    }
}
```

conditionals.cpp

You can make very complex decisions by putting if/else pairs inside of each other, or by having multiple different **conditions** (choices) in your if/else.

5 Now You Try

Doing Simple Math

Write a program that can ask for two numbers, add them, and print the answer. It should work like this:

```
Enter two numbers: 2 5
Sum: 7
```

Making Choices

Write a program that asks you to enter a number. If the number is bigger than 100 it says "That's big!". If the number is smaller than or equal to 100, it says "Wow, that's small!"

It should work like this:

```
Enter a number: 100
Wow, that's small!
```

Or...

```
Enter a number: 101
Wow, that's big!
```