

CSC 134

C++ Assignment M1LAB – Apple Orchard

Introduction

We know how to work with string variables so far. For example:

```
string name = "John";  
cout << "Hello, " << name << endl;
```

Will print:

```
Hello, John
```

Today we're going to work with two types of variables – numbers without a decimal point (often called “whole numbers”), which in computer terms are of type **integer**, and numbers with a decimal point, which are of type **double**.

(Double stands for “double precision floating point number”, but we'll use the short name. The important point to remember is that **doubles** have **decimal points**, and integers don't.)

Let's start writing some C++ code. Go to repl.it and create a new C++ repository by clicking “Start Coding”, scrolling down to C++, and clicking “New Repl”.

Start the file off by writing the standard comment block. Today's assignment is M1LAB so it will look like this. (Note, copy/paste usually won't work from Word documents. It's best to type this in by hand. You may want to delete the starter code that's already there, or add to it since some of the lines are the same.)

```
// CSC 134  
// M1LAB  
// Your Name  
// The Date
```

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

```

int main() {
    // This program will simulate an apple orchard.
    // The owner's name
    string name = "Your Name"
    // number of apples owned
    int apples = 100
    // price per apple
    double pricePerApple = 0.25

    // calculate the total price of the apples
    // TODO

    // print all the information about the orchard
    // TODO
    cout << "Welcome to " << name;
    cout << " 's apple orchard." << endl;
}

```

Notice there's comments that says **TODO** up there – this program isn't finished yet. (Lines starting with // are comments written for our benefit, they are ignored by the compiler.) We will test what we have first, then continue.

Run this code and makes sure it works without errors. A successful run will look like this, assuming your name is Jane Smith:

```
Welcome to Jane Smith 's apple orchard.
```

Common Errors and How To Fix Them

(If you didn't get any errors, congratulations! Skip to the next bolded section.)

If you get errors in the right hand window, double check your spelling, as it's usually a typo. If that doesn't work, check these options below.

C++ error messages are not exactly user friendly to new programmers. Don't worry – you'll get the hang of it. Usually just knowing where the error is is enough to fix it.

The general rule is, look at the line number listed after the filename like this:
`main.cpp:7:18: error: something` , that's saying that line 7, at 18 characters across, is where it got lost. The error might actually be before this point. (Kind of like how when driving, you usually realize "Oops, I'm lost" **after** you make a wrong turn.)

If you see: `error: expected ';' after expression`, look for the line number listed, and make sure that line (or the line above it) aren't missing a semicolon at the end. C++ lines end with a semicolon much like English sentences end with a period.

If you see: `error: use of undeclared identifier 'cout'; did you mean 'std::cout'?`, you are missing the line using namespace `std`; near the top of the file. Add that.

If you see: `error: non-ASCII characters are not allowed outside of literals and identifiers`, then you probably copy pasted. That's because Word uses special open quote and close quote symbols. In repl.it, you replace them with the regular double quote character `"`.

Printing More Information

So far the program doesn't actually do much. Let's print out all the information we have so far.

Remove the second TODO line, and add the following code so this section looks like this.

```
// print all the information about the orchard
cout << "Welcome to " << name;
cout << " 's apple orchard." << endl;
cout << "We have " << apples;
cout << " apples in stock" << endl;
cout << "Apples are currently $";
cout << pricePerApple << " each." << endl;
```

That's a little hard to read, so let's talk about what's going on.

The **cout** command starts printing information to the console (the text window on the right). **cout** uses what's known as a "stream" based output method. It's like a conveyer belt – every time something comes after a << statement, it's added in order to the list of things that will be printed out.

Every line of C++ code ends with a semicolon. But starting a new line of C++ doesn't start a new line in the console, you have to add the **endl** command to the stream, which is the equivalent of pressing the Enter key to start a new line.

This means we could have put all of that code in one line of C++ code like this:

```
// an example of code that works but is a pain to read
// (don't actually type this in to your program)
cout << "Welcome to " << name << " 's apple orchard." << endl << "We
have " << apples << " apples in stock" << endl << "Apples are
currently $"; << pricePerApple << " each." << endl;
```

That is really hard to read, so we break it up into more lines of code. Both of them print the same text, though.

Running Your Program Again

Running the program now should give output like this:

```
Welcome to Your Name 's apple orchard.
We have 100 apples in stock
Apples are currently $0.25 each.
```

Again, if you have errors, proofread your code and try to track them down.

(One reason we write our program step by step, and run it after each step, is that if you wait to the end, you're likely to get a lot of errors at once. It is much easier to fix bugs if there's only a few.)

If you see some odd formatting like `We have100apples in stock`, note that the code has spaces inside of the quotes. When you add something to the output stream with <<, it doesn't automatically space it the way you might like, you have to do that manually.

Adding Calculations

Next we need to multiply **apples** by **pricePerApples** to get the total cost.

Replace that remaining `// TODO` line with this code, so that section looks like this:

```
// calculate the total price of the apples
double totalPrice = apples * pricePerApple;
```

Try running the program. It should run without any errors. But it's still not showing us the total price.

Finish the program by writing a line of code to also print the total price if someone were to buy all of your apples. This involves adding another **cout** statement similar to the ones you used before. See if you can write this line yourself, using **cout** and the variable **totalPrice**, before checking your code against the finished version below.

The Finished Program

Here's what my code looks like. Double check your file against this.

```
// Header comments go here

#include <iostream>
using namespace std;

int main() {
    // This program will simulate an apple orchard.
    // The owner's name
    string name = "Your Name"
    // number of apples owned
    int apples = 100
    // price per apple
    double pricePerApple = 0.25

    // calculate the total price of the apples
    double totalPrice = (double) apples * pricePerApple

    // print all the information about the orchard
    cout << "Welcome to " << name;
```

```

    cout << " 's apple orchard." << endl;
    cout << "We have " << apples;
    cout << " apples in stock" << endl;
    cout << "Apples are currently $";
    cout << totalPrice << " each." << endl;

    // Final line, to print totalPrice
    cout << "If you want them all, that will be $" <<
    totalPrice << endl;
}

```

The output should look something like this when the program runs:

```

Welcome to Bob Jones 's apple orchard.
We have 100 apples in stock
Apples are currently $0.25 each.
If you want them all, that will be $25

```

Before you turn in your lab:

Is it displaying your actual name in the printout, and in the comment header?
 Does the program run without errors?
 Is the math correct?

Submitting Your Work

Like last time, you'll want to save your code as a .cpp (text) file by copying it out of the left hand window in your browser. Then you'll rename the filename in the format M1LAB_Lastname.cpp, and upload it to Blackboard.

You'll also want to include a screenshot of your compiler window showing your code in the middle and your program output on the right. Snipping Tool is the easiest way to do this, your browser may also have its own screenshot function. Name this something like M1LAB_Lastname.png (or whatever extension your image has). I demonstrate how to do this with Snipping Tool in the video linked in Blackboard.