CS 1428
Fall 2019
Gentry Atkinson
Lab 1

Introduction:

Like mathematical expressions, programs rely on variables. When using C++ you must declare variables as specific types determined by the kind of data you intend to store. These built-in data types are called “primitive”. Later on you will learn to define your own data types.

The primitives we will use are:

int	stores an integer value
float	stores a real number, which is called a “floating point” in programming
char	stores a single character
bool	stores a true or false value

In addition to variables, you need operators in order to write expressions. These operators take some values, process them, and return one value. Operators are also built into the language but can be altered later. Some of the built-in operators are **+, -, *,** /, and similar simple arithmetic signs.

Directions:

1- Launch Code::Blocks and start a new file. Name it your_last_name_lab1.cpp.

2- Include the standard header for this lab:

```
//Your Name
//CS1428 Fall 2019
//Lab 1
```

3- Include the iostream standard library and start your main function:

```
#include <iostream>

using namespace std;
```

```
int main() {
```

4- Declare your first integer type variable:

```
int myFirstInt;
```

This will create space in the computer's memory to store one integer. However, you haven't given myFirstInt any value yet, so right now there might be a default value like 0 or there might be garbage that was left behind by an old program. It's never good practice to reference a variable that hasn't been given a value.

5- Assign values to some variables:

```
myFirstInt = 5;
int mySecondInt = 7;
```

myFirstInt now has a value and is safe to use. Notice that for mySecondInt we chose to declare and assign the variable in the same statement, so mySecondInt is also safe to use. You can assign a variable with a fixed value (eg. myFirstInt = 5) or with another variable that already exists (eg. myFirstInt = mySecondInt) and you can assign a variable a new value as many times as you need to.

6- We can use our new variable to create an expression and display the result with the code:

```
cout << "The sum of my variables is " << myFirstInt + mySecondInt  << endl;
```

This one line of code will first compute the sum of the two integers and will then print the quoted text followed by the sum.

7- Practice declaring and assigning variables by doing the following:

1. Create an integer called myAge and assign it the value of your age.
2. Create an integer called thisYear and assign it the value of the current year.
3. Create a floating point number called pi and give it the value of Pi as accurately as you can remember.
4. Create a floating point number called myFirstFloat and give it the value of myFirstInt.
5. Create a boolean variable called myFirstBool and give it the value false, (e.g. **bool myFirstBool = false**;).
6. Create a boolean variable called mySecondBool and give it the value true.

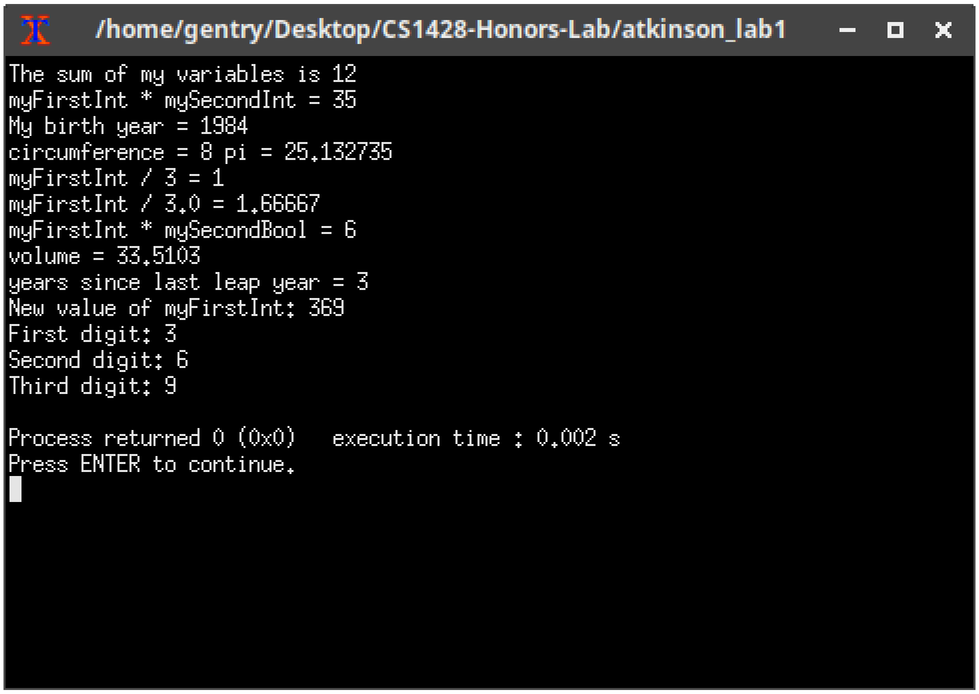
8- Write code to evaluate the following expressions and then output each answer to a separate line using **cout** statements:

1. The product of myFirstInt and mySecondInt.
2. Use thisYear and myAge to roughly calculate your birth year.
3. Use your pi variable to calculate the circumference of a circle with radius 4.
4. Divide myFirstInt by 3.
5. Divide myFirstInt int by 3.0.
6. The sum of myFirstInt and mySecondBool.
7. The volume of a sphere is 4/3 pi times the cube of the radius. Use the pi variable to compute the volume of a sphere with radius 2.
8. The “modulus” operator is written with the % character and returns the remainder when two integers are divided. Use the thisYear variable and the modulus operator to calculate the number of years since the last leap year.
9. Assign myFirstInt a value between 100 and 999. Use the % and / operators to print the three digits of myFirstInt separately.

There are two things that you should notice about this block of questions. The first is that questions 4 and 5 will return different values. This is because integer and floating point division are different in C++. The second is that question 7 will return an answer even though 5 + true doesn’t mean anything mathematically. This is because C++ stores the values true and false as integers “under the hood”. Good naming conventions will help prevent you from accidentally using variables in ways that will give you garbage results.

9- Complete your **main** function by adding **return 0;** as the last line.

10- Save your work. **Build and Run.** Correct any errors. You should see something like this:



```
The sum of my variables is 12
myFirstInt * mySecondInt = 35
My birth year = 1984
circumference = 8 pi = 25.132735
myFirstInt / 3 = 1
myFirstInt / 3.0 = 1.66667
myFirstInt * mySecondBool = 6
volume = 33.5103
years since last leap year = 3
New value of myFirstInt: 369
First digit: 3
Second digit: 6
Third digit: 9

Process returned 0 (0x0)   execution time : 0.002 s
Press ENTER to continue.
```

10- Submit your .cpp file as an attachment through TRACS.