

Name:

Sec:

06/05/2008

Programming Assignment #1

Review of C++ Control Structures and Functions

Starting from a simple hello world program, we will be reviewing the programming and use of some of the simple building blocks of the C++ language, including simple control structures, writing functions, declaring variables and programming input/output from your C++ programs.

Objectives

1. During this first week you should get an environment set up that you are comfortable with for writing, developing and debugging C++ programs. I suggest using whichever OS you are most comfortable working in, and getting a good Integrated Development Environment (IDE) such as Visual Studio, Eclipse or NetBeans to use for programming assignments.
2. Review the basic control structures used in C++, including selection statements (such as if/else and switch statements) and looping statements (like for and while loops).
3. Look at creating new functions in C++ and passing parameters to functions as input to the function, as well as returning values from functions as the result of the function.
4. In addition, review simple input and output to C++ programs using the iostream library functions.

Instructions

Part 1: Getting Started with Hello World

You should start by finding an appropriate integrated development environment (IDE) and/or compiler that you can use to help you write, compile and run C++ programs for this course. Some suggestions of appropriate IDEs include Microsoft Visual Studio, NetBeans and Eclipse.

Most IDEs organize your work around the concept of Projects. You will want to learn how to create a C++ project in your IDE of choice, and you will probably end up creating a new project for each of the assignments we will be performing in class this semester.

Once you have a project created for assignment #1, you should add a new source file to your project. Call your source file “assg1.cpp”. Here is a simple hello world program (based on Example 2-1 from the text book):

```
// Author: Joe Student
// Program: Assignment #1
// Class: CSci 152, Summer 2008
// Date: June 5, 2008
//
// Desc: This is a simple Hello World example C++ program to get
// you started. Your first task should be to figure out how to create
// a C++ project and add a new source file in your IDE of choice.
// Then, once you have your source file, you need to learn how to
// build and run your program. You will be using these basic steps
// for all assignments for this course, so figure them out for your
// environment ASAP.
//
// Please include a descriptive header like this with your name,
// assignment, etc. for all assignments for the course.
#include <iostream>

using namespace std;

int main(int argc, char** argv)
{
    cout << "Hello CSci 152!" << endl;

    return 0;
}
```

If you don't remember the purpose of the `#include <iostream>` directive, or how to use iostreams to perform program input and output using the `cin` and `cout` streams, please review the relevant sections of chapters #2 and #3. If you have forgotten the purpose of the `using namespace std;` syntax, there is a brief description of this on pg. 79 (3rd edition) of the text book, and more information in chapter #8 about namespaces.

`main()` is a function in C/C++, just like any other function, with one exception. In any C/C++ program, the `main()` function serves a special purpose. It will be the first function that is invoked when you run your compiled C/C++ program. Thus every C/C++ program that you write will have one and only one `main` function, and it will be the place that execution of your program starts from. If you are unsure of what the `int` before `main()` indicates, or what the two parameters being passed to `main()` (`argc` and `argv`) mean, you should review the relevant material in Chapter #6 about value-returning functions and parameter lists.

Part 2: I/O Streams

The remaining parts of the assignment ask you to write several simple functions that will help you to review some of the basic control structures, syntax and concepts of C++ you should be familiar with.

Write a function called `getName()`. The function should take no parameters as input, and it should return a string as a result of calling it. Inside of your `getName()` function prompt the user to enter 3 pieces of information, their first name, middle initial and last name. You should display an appropriate piece of text on standard

output for each of these, and you should read the response into a string variable. The return result of your getName() function should combine these 3 separate strings into one and return the single string as its result.

Part 3: if/else Selection Control Structure

Write a function called isLeapYear(). The purpose of the function is to determine if an integer representing a calendar year is a leap year or not. Your isLeapYear() function should accept a single integer as a parameter for its input, this will be the year that will be tested. Your isLeapYear() function should return a bool data type as its result (review pg. 183 for the basic bool data type).

The function should return a result of **true** if the year passed is a leap year, and it should return a value of **false** otherwise. The determination of a leap year requires several tests. A year is a leap year if it is divisible by 4, but is not divisible by 100 except when divisible by 400.

Thus:

1996 **was** a leap year (divisible by 4, not divisible by 100)
1900 **was not** a leap year (divisible by 4, but also divisible
by 100 but not by 400)
2000 **was** a leap year (divisible by 4 and also 100 but also divisible by 400)

Hint 1: you will need nest your tests. For example, as the first test, you know that if the year is not divisible by 4 then the answer is **false**, it is not a leap year. But if the year is divisible by 4 then you are not yet done because you need to see if it is divisible by 100 (etc. if it is divisible by 4 and by 100, there is still a chance it is a leap year depending on if it is divisible by 400 or not).

Hint 2: An easy way to test if an integer is evenly divisible by some other integer is to use C's modulus operator (a review of basic arithmetic operators begins on pg 39, including the modulus operator). The modulus operator '%' returns the remainder of a division operation. Thus to check if something is evenly divisible by, say 4, you can test if the remainder is 0 using the '%' like this:

```
if (2004 % 4 == 0)
    // then the year 2004 must be divisible by 4 since the remainder is 0
else
    // otherwise the remainder was not 0 and thus 2004 must not have
    // been evenly divisible by 4
```

Make sure that your isLeapYear() function returns an appropriate bool answer as a result of calling it.

Part 4: looping Control Structure

Write a function called `displayManyTimes()`. Your function should take two parameters as input. The first parameter should be a string, and the second parameter should be an integer. The `displayManyTimes()` function should not return any value (e.g. it is a void function, see Chapter #7, pg. 346). The purpose of your `displayManyTimes` function is to display the passed in string a number of times on standard output. So if the function is called with the string "Derek S. Harter" and an integer parameter of 3, it should display on standard output the following:

```
Derek S. Harter
Derek S. Harter
Derek S. Harter
```

You should use a for or while loop to display the string the indicated number of times on standard output.

Part 5: Further Examples and Help

To make the requirements of the 3 functions you are required to write, here is an explicit example of how they should work. The following is an example of a `main()` function that calls the 3 functions you were asked to write:

```
// Author: Joe Student
// Program: Assignment #1
// Class: CSci 152, Summer 2008
// Date: June 5, 2008
//
// Desc: An example of a main() function to test your getName(),
//       isLeapYear(), and displayManyTimes() functions for assignment #1.
#include <iostream>

using namespace std;

// function prototypes
string getName();
bool isLeapYear(int year);
void displayManyTimes(string s, int n);

int main(int argc, char** argv)
{
    string fullName;

    fullName = getName();
    cout << "Hello, " << fullName << ", welcome to CSci 152!" << endl;

    cout << "1996 isLeapYear is " << boolalpha << isLeapYear(1996) << endl;
    cout << "1900 isLeapYear is " << boolalpha << isLeapYear(1900) << endl;
    cout << "2000 isLeapYear is " << boolalpha << isLeapYear(2000) << endl;
    cout << "1989 isLeapYear is " << boolalpha << isLeapYear(1989) << endl;
    cout << "4 isLeapYear is " << boolalpha << isLeapYear(4) << endl;
    cout << "400 isLeapYear is " << boolalpha << isLeapYear(400) << endl;
    cout << "300 isLeapYear is " << boolalpha << isLeapYear(300) << endl;
    cout << "304 isLeapYear is " << boolalpha << isLeapYear(304) << endl;

    displayManyTimes(fullName, 5);
}

// The implementation of your 3 functions for assg1 should go below here
```

If you have implemented your 3 functions correctly, and you run the program with the above example main() function, you should get output for example like this:

```
Enter your first name: Derek
Enter your middle initial: S.
Enter your last name: Harter
Hello, Derek S. Harter, welcome to CSci 152!
1996 isLeapYear is true
1900 isLeapYear is false
2000 isLeapYear is true
1989 isLeapYear is false
4   isLeapYear is true
400 isLeapYear is true
300 isLeapYear is false
304 isLeapYear is true
Derek S. Harter
Derek S. Harter
Derek S. Harter
Derek S. Harter
Derek S. Harter
```

Assignment 1 Finished

You have now completed Assignment 1. The final step you should perform when you complete an assignment is to upload your finished assignment to your eCollege account. Go ahead at this point and upload your `assg1.cpp` source file to eCollege. You should do this by selecting the Dropbox tab in eCollege, and choosing the “Week 1: Assignment” drop box, then hit the GO button. This will take you to a page that will allow you to upload your source files to eCollege. I only need your source files. Thus for this week you should only upload “`assg1.cpp`”, not any of your executables or other project files that might be created by your IDE.