CSCI 136 Section 05 – Fall 2014

Lab Assignment #3

This week's assignment will have you write programs that introduce you to loops and functions. Feel free to work in pairs and ask for help early.

Using the Terminal command line, create a folder (using the `mkdir` command) called `lab03` and `cd` into that folder. I have added an example of what should happen when the program is run to the end of this document. Your source code file must have a preamble at the top (see the programming rules and guidelines document in Blackboard for more on the preamble).

a. (`lab03a.cpp`) Write a program that has a hard-coded integer valued from 1 to 5 (just pick a value – for example: `int ans = 3;`) and asks the user to guess it. If they get it wrong, the program should ask them again and keep asking until they guess correctly. You should use a `while` or `do-while` loop for this. Don't overthink this one.

b. (`lab03b.cpp`) Write a program that outputs all of the prime numbers between 3 and 100. A prime number is a number that can only be evenly divided by one and itself (i.e., 3, 5, 7, 11, 13, 17, etc…). Solve this problem by using a doubly-nested loop. The outer loop will iterate from 3 to 100 and the inner loop will check to see whether the counter value for the outer loop is prime. One way to decide whether a number `n` is prime is to loop from `2` to `n-1`; if any of these numbers evenly divides `n`, then `n` cannot be prime. If none of the values from `2` to `n-1` evenly divide `n`, then `n` must be prime.

c. (`lab03c.cpp`) Rewrite the program from (b) above, but this time create and use a function called `isPrime`. The function should take an integer as a parameter and return a Boolean true or false. In your `main()`, in the body of the loop from 3 to 100, you'll call this new function to determine if the number is prime and decide to output based on the result.

d. (`lab03d.cpp`) Write a program that outputs the sum of the digits for each number from 0 to 200. You'll need to write and use a function called `sumOfDigits` that takes an integer as a parameter and returns the sum of its digits (also an integer). To calculate the sum, use the division and modulus operators to extract the digits one at a time. For example, if the number is `123`, observe that `(123 % 10)` is `3` and `(123 / 10)` is `12`. You will need to use a loop for the extraction of the digits.

Submitting your work

Make sure you are in your `lab03` folder (use the `pwd` command) with your four source code files (use the `ls` command) and then run the following to create a zip archive of them:

```
$ mkdir lastname_firstname_lab03
$ cp lab03a.cpp lastname_firstname_lab03/lab03a.cpp
$ cp lab03b.cpp lastname_firstname_lab03/lab03b.cpp
$ cp lab03c.cpp lastname_firstname_lab03/lab03c.cpp
$ cp lab03d.cpp lastname_firstname_lab03/lab03d.cpp
$ zip -r lastname_firstname_lab03.zip lastname_firstname_lab03/
```

You'll need to change `lastname` and `firstname` to your actual last and first names in the steps above. Once you have your zipfile, you can use that file as your submission for the assignment in Blackboard.

If you are working in pairs, then you should have both of your names included as comments in the source code file's preamble (see the programming rules and guidelines document in Blackboard for more on the preamble). Also, write both your names to the notes section in the submission form when submitting to Blackboard.

Sample output

```
$ ./lab03a
Guess the number I am thinking of (between 1 and 5): 2
No, that is not correct.  Guess again: 3
You got it right!
```

Output is straightforward for parts b, c and d and thus has been omitted.