Langara College

# Department of Computing Science & Information Systems

# CPSC1160 – Algorithms and Data Structures I

###### **Lab2: Functions and Arrays**

**Problems [35 marks]**

**Instructions:**

1. Create a folder named **Lab2**
2. For each problem create a folder inside the folder Lab2 to store the files from each problem
3. Download **scitestart.zip** from D2L. Unzip it to your desktop and run **scitestart.bat** to start SciTE so that SciTE can fully function.
4. For each problem create a make file to do compilation and linking. Name the make file makefile and save it inside the same folder as the problem. The make file contains one or more rules. The syntax of a rule is as follows:

targetList: dependencyList

commandList

**Notes:** The commandList must start with a tab character. Each new rule must start from a new line.

**For example:**

CheckBMI: CheckBMI.o

g++ CheckBMI.o -o CheckBMI

CheckBMI.o: CheckBMI.cpp

g++ -c CheckBMI.cpp

To run the make file on SciTE, click Tools->Build or on the command line enter the command *make*

**Problem 1: [10 marks] Find the average of an array of numbers**

(filename: **FindAverage.cpp**)

Write two overloaded functions with the following headers to find the average of an array of double values and an array of integer values:

double average(const double num[] , int size);

double average(const int num[] , int size);

Write a test program that invokes these two functions and displays the results on the console.

**Problem 2: [15 marks] Palindromic prime** (filename: **CheckPalinPrime.cpp**)

A palindromic prime is a prime number and also a palindromic. For example, 131 is a prime number and is also palindromic. So are 313 and 757. Write a program that displays the first 100 palindromic prime numbers. Display 10 numbers per line and align the numbers properly, as follows:

2 3 5 7 11 101 131 151 181 191

313 353 373 383 727 757 787 797 919 929

Use functions wherever possible.

**Problem 3: [10 marks] Count occurrence of each letter in a string**

(filename: **CountLetters.cpp**)

Write a function that counts the occurrence of each letter in a string by using the following function header:

void count(const string& s, int counts[]);

where counts is an array of 26 letters. counts[0], counts[1], …, and counts[25] count the occurrence of a, b, …, and z, respectively. Letters are not case-sensitive, i.e., letter A and a are counted the same as a.

Write a test program that reads a string from the console, invokes the count function, and displays the non-zero counts. Here is a sample run of the program:

Enter a string: There are 40 students in the class

a: 2 times

c: 1 times

d: 1 times

e: 5 times

h: 2 times

i: 1 times

l: 1 times

n: 2 times

r: 2 times

s: 4 times

t: 4 times

u: 1 times

**When to hand in**

By the end of the lab time, demo **Problem 1** to the instructor.

By 11:59pm, Thursday, September 21, 2016, zip the folder **Lab2** and submit it to D2L.