Lab 1

Course: CSE 165
Section: 02L & 03L
Due: Sunday, September 19, at 11:59 pm

All the exercises below are selected from the textbook: Thinking in C++ (volume 1).

1. [Exercise-2 on Page 120] Using Stream2.cpp and Numconv.cpp as guidelines, create a program that asks for the radius of a circle and prints the area of that circle. You can just use the '*' operator to square the radius. Do not try to print out the value as octal or hex (these only work with integral types). [15 points]

2. [Exercise-6 on Page 120] Change Fillvector.cpp so that it concatenates all the elements in the vector into a single string before printing it out, but don't try to add line numbering. [15 points]

3. [Exercise-1 on Page 226] Create a header file (with an extension of '.h'). In this file, declare a group of functions by varying the argument lists and return values from among the following: void, char, int, and float. Now create a .cpp file that includes your header file and creates definitions for all of these functions. Each definition should simply print out the function name, argument list, and return type so you know it's been called. Create a second .cpp file that includes your header file and defines int main( ), containing calls to all of your functions. Compile and run your program. [20 points]

4. [Exercise-2 on Page 226] Write a program that uses two nested for loops and the modulus operator (%) to detect and print prime numbers (integral numbers that are not evenly divisible by any other numbers except for themselves and 1). [20 points]

5. [Exercise-3 on Page 227] Write a program that uses a while loop to read words from standard input (cin) into a string. This is an "infinite" while loop, which you break out of (and exit the program) using a break statement. For each word that is read, evaluate it by first using a sequence of if statements to "map" an integral value to the word, and then use a switch statement that uses that integral value as its selector (this sequence of events is not meant to be good programming style; it's just supposed to give you exercise with control flow). Inside each case, print something meaningful. You must decide what the "interesting" words are and what the meaning is. You must also decide what word will signal the end of the program. Test the program by redirecting a file into the program's standard input (if you want to save typing, this file can be your program's source file). [30 points]

Requirements:

* Usage of spaces, blank lines, indention, and comments for readability

* Descriptive names of variables, functions, structs, classes, and objects (if any)

* Appropriate usage of structs, classes, and objects (if any)

Late Penalties:

10-point deduction per day late