Assignment 1

Re-submit Assignment

Due Sep 28, 2016 by 7pm **Points** 120 **Submitting** a file upload **File Types** pdf

Project 1.a

In this assignment you'll get some practice with some of the tools we'll be using throughout the course. These instructions will assume that you have followed the tutorials in the "Tools You Will Need" page.

Use a terminal emulator to log into the school server (flip). Type **ls** to list the contents of the directory you're in. Type **ls -l** for more information about each file. Create a new directory called "week1" by typing **mkdir week1**. Go into that directory by typing **cd week1** (you can get back out of it by typing **cd ..** which moves you up one level from your current directory). Use vim (or emacs or nano) to make a file named animal.cpp. Type the following code into the file:

```
#include <iostream>
#include <string>

// a simple example program
int main()
{
    std::string faveAnimal;
    std::cout << "Please enter your favorite animal." << std::endl;
    std::cin >> faveAnimal;
    std::cout << "Your favorite animal is the " << faveAnimal;
    std::cout << "." << std::endl;

    return 0;
}
```

Add a comment block at the top as discussed in the Code Style Guidelines. Now save the file. Type **ls** to verify that this directory now contains a file named animal.cpp. Next type **g++ animal.cpp -o animal**. This will compile your source file and create an executable file named "animal". The **-o** flag lets you choose a name for the executable file. If you instead just type **g++ animal.cpp**, the name of the executable file will be "a.out". It is important that you do not accidentally give your executable file the same name as the source file. If you do that, then your executable file will replace your source file and your source file will be gone (and you will be sad). Now type in ./**animal**. The program should now ask you to enter the name of your favorite animal, and after you do, it will print out "Your favorite animal is the <whatever you typed>." Notice that this program only reads the first word of the input, so if the animal name contains any spaces, the full name won't be printed out.

Once you have verified that your program compiles and runs correctly, you need to transfer your source file, "animal.cpp", from the school server to your home computer using FileZilla (or scp).

You will submit your code on TEACH (together with the code for parts 1.b-d). To do so, open up the TEACH website, find submission "week1" for your section of the course, and submit your source files there.

Lastly, create a pdf document with your name and your ONID. Submit your pdf here in Canvas by clicking the "Submit Assignment" link and selecting it for upload.

For all programming projects in this course you can assume that the user (the person using your program) will not try to do so in a way that violates the specifications. For example, if the specifications indicate that the program should expect an int, then you don't have to worry about getting something else, like a double or bool. Some specifications are implicit rather than explicit. For example in Project 2.a the description doesn't tell you which numeric type to use, but you can infer it from the given example. **There are several numeric types available, but for the programming projects you should stick with *int* for integers and *double* for floating-point numbers.**

Remember to not use any material we haven't covered yet (as stated in the syllabus).

Project 1.b

Write a program that asks the user for five numbers and then prints out the average of those numbers. When you run your program, it should match the following format:

```
Please enter five numbers.
-2.4
5.1
6.0
123.8
-19.0
The average of those numbers is:
22.7
```

The file must be named: **average.cpp**. To compile it into an executable file named "average", enter "g++ average.cpp -o average". "g++" is the name of the compiler, "average.cpp" is the name of your program, "-o average" says that you want the executable file to be named "average" (you do not have to match the name of your .cpp file). If you don't use the -o option, then the executable file will be named "a.out".

Project 1.c

Write a program that converts Celsius temperatures to Fahrenheit temperatures. The formula is:

$$F = \frac{9}{5}C + 32$$

where F is the Fahrenheit temperature and C is the Celsius temperature. The program should prompt the user to input a Celsius temperature and should display the corresponding Fahrenheit temperature. It should display **only** the converted temperature without any text (not even a 'F'). When you run your program, it should match the following format:

```
Please enter a Celsius temperature.
-10.5
The equivalent Fahrenheit temperature is:
13.1
```

The file must be named **tempConvert.cpp**.

Project 1.d

Write a program that asks the user for a (integer) number of cents, from 0 to 99, and outputs how many of each type of coin would represent that amount with the fewest total number of coins. When you run your program, it should match the following format:

```
Please enter an amount in cents less than a dollar.
87
Your change will be:
Q: 3
D: 1
N: 0
P: 2
```

Hint: You will find the mod operator (%) and integer division useful.

The file must be named: **change.cpp**.