CSCI 1300 - Intro to Computer Programming
Instructor: Knox
Assignment 2

Due Sunday, Sep 17th, by 6 pm (remember 5% bonus if submitted by Friday Sept 15th 6pm)

For this assignment, write all of your code in one file, called main.cpp, which is the default source file name in a CodeBlocks project. You can find the template of this file uploaded in moodle itself. You just need to insert your code in this file at appropriate places. The template file has all the function definitions including main, you need not modify that part. Once you have your code running on your virtual machine (VM), you must submit it to the autograder (COG). You must also submit your code to Moodle to get full credit for the assignment. You can modify your code and resubmit as many times as you need to, up until the assignment due date. Remember to include comments and a description of the algorithm you use for each function. Please also include a comment at the top of your file with the following format:

```
// Author: CS1300 Fall 2017
// Recitation: 123 - Favorite TA
//
// Assignment 2
```

TAs will be checking that your code has the required comments and explains the algorithms you have used.

- **10 points for comments at the header of the file**
- **10 points for algorithm description**

What to put in your algorithm description:

This is an example C++ solution to problem 5 of Assignment 1. Look at the code and the algorithm description for an example of what is expected.

```
/**
 * Algorithm that checks what range a given MPG falls into.
 * Outputs different string based on three categories of
 * MPG: 50+, 25-49, and less than 25.
 * Returns nothing
 */

void checkMPG(float mpg) {
      if(mpg > 50) {
            cout << “Nice job” << endl;
      }
      else if(mpg > 25) {
            cout << “Not great, but okay.” << endl;
      }
      else {
            cout << “So bad, so very, very bad” << endl;
      }
}
```

CSCI 1300 - Intro to Computer Programming
Instructor: Knox
Assignment 2

The following algorithm description does not mention in detail what the algorithm does and does not mention what value the function returns. This would not receive full credit.

```
/**
 * Checks mpg
 */

void checkMPG(float mpg) {
      if(mpg > 50) {
            cout << "Nice job" << endl;
      }
      else if(mpg > 25) {
            cout << "Not great, but okay." << endl;
      }
      else {
            cout << "So bad, so very, very bad" << endl;
      }
}
```

Submitting Your Code to the Autograder

The computer science autograder, known as COG, can be found here:

https://web-cog-csci1300.cs.colorado.edu

Login to COG using your identikey and password.

- **Select Assignment 2 from the assignment drop-down box.**
- **Upload your .zip file and click Submit.**

Your main.cpp file must be compressed into a zip folder. You may name this zip folder any name of your choosing, but the .cpp *must* be named main.cpp.

COG will run its test cases and display the results in the window below the Submit button. If your code doesn't run correctly on COG, read the error messages carefully, correct the mistakes in your code, and upload a new file. You can modify your code and resubmit *as many times* as you need to, up until the assignment due date.

Before you submit your code to COG, make sure it runs on your computer. If it doesn't run on the VM, it won't run on COG. If your code does not compile, you will receive a 0.

What to do if you have questions

There are several ways to get help on assignments in 1300, and depending on your question, some sources are better than others.

- **Piazza is a good place to post technical questions, such as how to get user input, or treat that input as an integer. When you answer other students' questions, please do not post your code.**
- **The Course Assistants (CAs) are also a good source of technical information.**

CSCI 1300 - Intro to Computer Programming
Instructor: Knox
Assignment 2

- **If,** *after reading the assignment write-up***, you need clarification on what you're being asked to do in the assignment, the TAs and the course instructors are better sources of information than Piazza or the CAs.**

You must name the functions as indicated in each section below. Importantly, the cout formats provided for each problem are not suggestions – they MUST be followed precisely, word-for-word and including all punctuation marks, otherwise the autograder will not recognize your results and you will not receive credit.

Problems Set:

Write a function for each of the following problems within your main.cpp file and make the appropriate calls to each function from within your main function.

Problem 1

The U.S. Census provides information about the current U.S. population as well as approximate rates of change. Using those rates and the current US population, write an algorithm to calculate the U.S. population in exactly one year (365 days). Your algorithm should output the result of your calculations.

Three rates of change are provided:

a. There is a birth every 8 seconds
b. There is a death every 12 seconds
c. There is a new immigrant every 33 seconds

- Your function should take current population as an **integer parameter**.
- Your function should **return** the population in a year.
- **Your function** *MUST* **be named howMany.**

For example, given an initial population of 1,000,000, your function would return 3,269,636.

Problem 2

A day has 86,400 seconds (24*60*60). Given a number of seconds in the range of 0 to 1,000,000 seconds, output the time as days, hours, minutes, and seconds for a 24- hour clock.
E.g., 70,000 seconds is 0 days, 19 hours, 26 minutes, and 40 seconds.

Your **function** should output:
Time is W days, X hours, Y minutes, and Z seconds.

- Your function should take the number of seconds as an **integer parameter**
- **Your function** *MUST* **be named howLong.**

CSCI 1300 - Intro to Computer Programming
Instructor: Knox
Assignment 2

Problem 3

In science, temperature is always described in Celsius, but in the U.S. we tend to use Fahrenheit temperatures. Write an algorithm to convert a Celsius temperature into Fahrenheit.

- Your function should take the temperature in Celsius as an **integer parameter**
- **Your function *MUST* be named howHot.**
- Your function should **return** the temperature in fahrenheit .

For example, given an initial temperature of 20 degrees C , your function would return 68.

Points to remember when running your code in COG:

These are some things in which COG is very rigid:

1. Students *must* include *void* for the input parameter of functions that don't take input, rather than leaving it blank.

For example,

```
*correct*      int some_function (void) {}
*incorrect*    int some_function () {}
```

2. Answers printed from *within* a user-defined function must end with a period.

```
// Inside some_function()
*correct*      cout << "The answer is 42.";
*incorrect*    cout << "The answer is 42!";
*incorrect*    cout << "The answer is 42";
```

3. In addition to Point #2, answers printed from *within* the user-defined function must exactly follow the cout string template given in the writeup.

```
// Inside some_function()
*correct*      cout << "The time is 1 days, 2 hours, 10 minutes, and 5 seconds."  << endl;
*incorrect*    cout << "The tume is 1 days, 2 hours, 10 minutes, and 5 seconds." << endl;
*incorrect*    cout << "The time is 1 days, 2   hours, 10 minutes, and 5 seconds." << endl;
*incorrect*    cout << "The time is 1 day, 2 hours, 10 minutes, and 5 seconds." << endl;
```