CIS 236 – Programming in C

Program:	**Week 01**
Points:	**20**
Chapters:	**1, 2**

Description

Copy a C program that converts a temperature from Fahrenheit to Celsius. Execute and debug the program from within the Dev-C++ IDE. Execute the program from Windows File Explorer and a command window.

Follow the steps below!

Learning Objectives

In this assignment, you will practice:

- Starting the C compiler we will use for class
- Typing in a short C program with a variety of statements
- Saving a C source file correctly
- Compiling and executing a C program from within Dev-C++
- Debugging errors, if necessary
- Executing a program using two different techniques: by double-clicking and by using a command window

Step 1 – Start Dev-C++

Start the Dev-C++ IDE. Create a new source code file by clicking the **File** menu, then **New**, then **Source File** (or click the **New icon** in the toolbar).

Step 2 – Type in the Description of the Program

In this step, you will practice creating comments in C.

At the top of the new file, type the following, exactly as it appears below:

```
This program prompts the user for a temperature in
Fahrenheit and displays the result to the screen.
```

After you have finished typing, highlight those lines, and then press **Ctrl + /** on your keyboard. You have made each line a single-line comment.

Hint – It can be difficult to precisely place the cursor at the beginning of a line in Dev-Cpp. If you need to highlight an entire line or a group of lines, it's easier to start at the end of a line and drag to the left and up, instead of starting at the beginning and dragging to the right and down.

Copy those lines and paste them below, leaving a few blank lines in between. Highlight the lines you just pasted, and then press **Ctrl + /** on your keyboard. The **//** at the beginning of each line should be toggled off.

Now make this paragraph a multi-line comment, by typing **/*** in the blank line above the paragraph. Then add ***/** in the blank line below the paragraph.

Add in the part about Celsius, per the picture below:

```
This program prompts the user for a temperature in
Fahrenheit, converts the temperature to Celsius, and
displays the result to the screen.
```

Step 3 – Add a #include Preprocessor Directive

Type the following, which will bring in the definitions for the input and output functions (scanf and printf), when this program is compiled:

```
#include <stdio.h>    //library for input and output
```

Hint – When you type the less than symbol **<**, Dev-Cpp will add the greater than symbol **>** automagically!

Note – The name of the library is **stdio.h**, which stands for "Standard Input Output." It is not "studio.h".

Step 4 –Type the First Line & Curly Braces of the main Function

Type in the first line of the main function, exactly as it appears below. Be sure it is all lowercase. After typing this line, press the Enter key, so that your cursor is on the line below.

```
int main (void)
```

On the line below, type the left curly brace **{**. You should see that Dev-Cpp has automatically added the closing right curly brace **}**:

```
int main (void)
{
}
```

At this point, just press Enter.

You should see that Dev-Cpp has automatically added a blank line in between the curly braces and has indented the cursor, as in the picture below. This is what you want. Indenting code between curly braces is a best practice in C programming.

If this technique didn't work for you, it's ok. Just manually add the blank line and press the Tab key to indent.

```
int main (void)
{
    
}
```

Step 5 –Type in the Rest of the main Function

Type in all remaining statements from the picture below. It's ok if your line numbers don't match up exactly with the picture.

Note – In the picture below, that is a **lowercase L** in the text **"%lf"** on line 24, not the number 1.

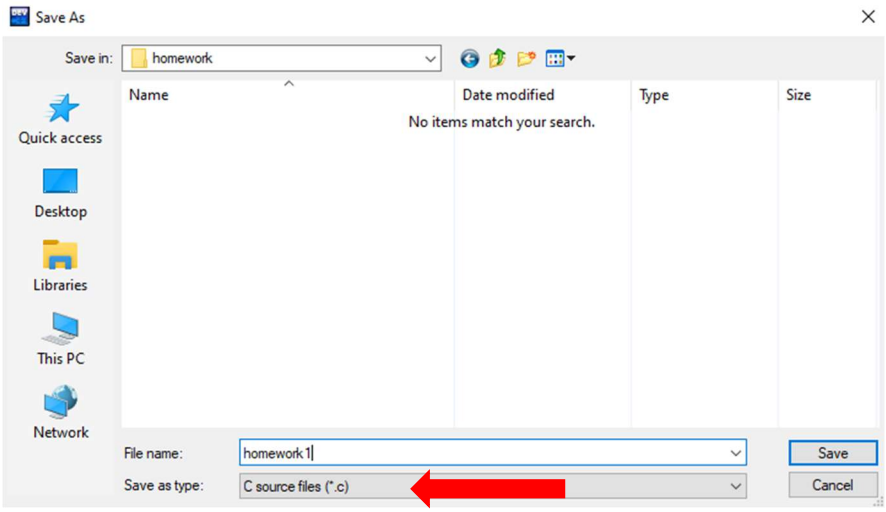
```
12  // main function required by all C programs
13  int main (void)
14  {
15      // Variable declarations
16      double tempInFahrenheit;
17      double tempInCelsius;
18
19      // Displays a prompt so the user knows what to do
20      printf("Enter the temperature in Fahrenheit: ");
21
22      // Reads the number typed by the user, and saves it into
23      // the variable tempInFahrenheit
24      scanf("%lf", &tempInFahrenheit);
25
26      // Computes the temperature in Celsius
27      tempInCelsius = (tempInFahrenheit - 32) / 1.8;
28
29      //(T(F) - 32) / 1.8  // conversion formula for reference
30
31      // Displays the temperature in Celsius
32      printf("%.2f F converts to %.2f degrees C", tempInFahrenheit, tempInCelsius);
33
34      // All C programs need this at the end of main
35      return (0);
36  }
```

Step 6 – Save the Program

In Dev-C++, click File > Save As.

Name the file Homework1 (no spaces!)

Note that you will have to **CHANGE THE FILE TYPE** in the Save As dialog pictured below. Changing the file type using the drop down will automatically add the **.c** extension when it saves. Remember where you've saved the file.



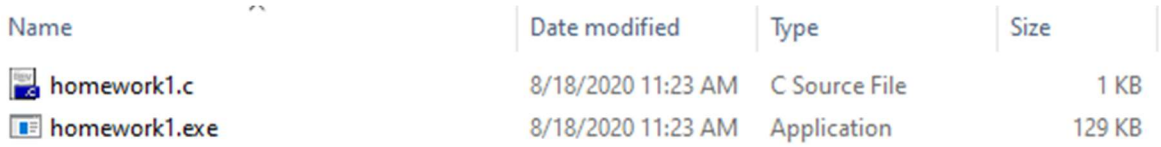
Step 7 – Compile and Execute the Program from Dev-C++

Compile and run the program by clicking the **Compile & Run icon** in Dev-C++'s menu bar, or by pressing **F11** on your keyboard. If it does not compile, fix any errors until it runs. Do not continue to Step 5 until the program is working.

Step 8 – Execute the Program Using the Executable File from Windows File Explorer

Keep the file open in Dev-C++ while you perform the next steps!

Outside of Dev-C++ (using Windows File Explorer), look in the folder where you saved the C source code file. After successfully compiling a program in C, you will see another file with the same name as the original, but with the **.exe** file extension. This is your **executable file**. It may be called an Application file, as in the picture below:

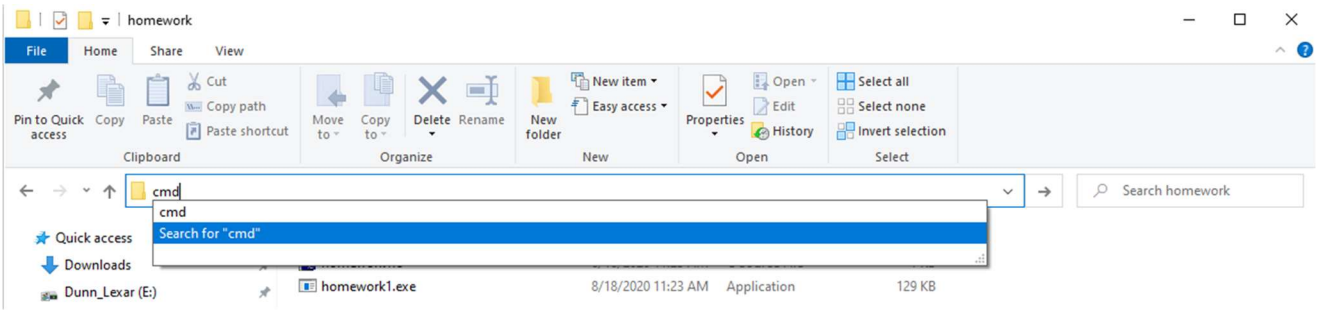


Double-click on this executable file to execute your program again. What happens when you press enter after typing in a value for Fahrenheit? **Write your answer as a comment at the top of your program.**

Step 9 – Execute the Program Using the Executable File from a Command Window

You will open a **command window** for this folder and execute your program from the command line (aka DOS prompt). Be sure you are still looking at the folder where you've saved the .c and .exe files.

At the top of the folder, in the Address Bar, type **cmd** as in the picture below, and then press the Enter key.



A black command window should display.

To execute the program from the command line:

1. Type the name of the exe file. Do not include the file extension ".exe".
2. Press Enter.

For example, if you named the program **Homework1** as instructed above, just type **Homework1** at the prompt. If you used spaces in the name, surround the name with double-quotes: "homework 1"

What happens this time when you press enter after typing in a value for Fahrenheit? **Write your answer as a comment at the top of your program.**

Step 10 – Add system("pause") to the Program

Go back to the C source code in Dev-C++, and add the following code on the **line above the return(0); statement**:

system("pause");

Be careful about copying and pasting from a Word document!! Word converts straight double-quotes to "Smart Quotes" which Dev-C++ doesn't recognize. It's safer to just retype.

Compile and run the program again from within Dev-C++. Then execute it by double-clicking the executable and running it from the command window. What is the difference now? **Write your answer as a comment at the top of your program.**

Step 11 – Submit to the Dropbox on Canvas

Submit **only the C source code file** to the dropbox on Canvas. Do not submit the .exe file.

For complete credit on this project:

All lines must be included as shown.

The program must compile, execute, and give accurate output.

The questions from steps 8, 9 & 10 must be answered as comments at the top of your program. Please label them appropriately.

The C source code must be submitted before the due date and time shown in Canvas.