

**60-140-1 and 60-140-2 ASSIGNMENT #4**  
**Handed Out: Thurs. Oct 22, 2015 for (60-140-01 and 60-140-02)**  
**Due: Thurs Oct 29, 2015 for (60-140-01 and 60-140-02)**  
**Total: 50 marks**

---

**Objective of Assignment :** To continue to write a C program to solve a problem using functions and function calls (including call-by-reference parameters and global variables), but no decision or repetition instructions.

**Scope :** Assignment covers materials up until end of chapter 4.

**Other Things to learn from Assignment :** How to make function calls using **call-by-value** parameters for input parameters and **call-by-reference** parameters for all output parameters. Also, how make function calls with no **parameters where all variables are global**.

**DO NOT USE GLOBAL VARIABLES IN THE FIRST PROBLEM SOLUTION.**

**Important:** Do not forget to type in your *full name, student number, lecture section number, lab section number* and *date* in BOTH the algorithm and source C program files.

**Electronic Assignment Submission:**

**03-60-140-1 students: email script file to [cs140\\_01@cs.uwindsor.ca](mailto:cs140_01@cs.uwindsor.ca) with subject including: Name, student id, lecture section, lab [section], assignment #4 (in the subject of the mail submission of script file).**

**03-60-140-2 students: email script file to [cs140\\_02@cs.uwindsor.ca](mailto:cs140_02@cs.uwindsor.ca) with subject including: Name, student id, lecture section, lab [section], assignment #4 (in the subject of the mail submission of script file).**

\*Only the assignments currently due that are submitted to this site within two days before and by the due date, are retrieved for marking. Others are deleted soon after.

---

**Problem:**

Given the conversion rate between degree Celsius ( $^{\circ}\text{C}$ ) and degree Fahrenheit ( $^{\circ}\text{F}$ ) is  $^{\circ}\text{C} = 5.0/9.0 (^{\circ}\text{F} - 32.0)$ , use two functions **Calc\_Celsius** (which returns the Celsius equivalent of a Fahrenheit temperature) and **Calc\_Fahr** (which returns the Fahrenheit equivalent of a Celsius temperature) to write two programs (one with call-by-reference parameters and the other with global variables), each of which prints

- i) the Celsius temperatures of three Fahrenheit temperatures entered at the keyboard by the user as input data, and
- ii) the Fahrenheit equivalents of three Celsius temperatures entered at the keyboard by the user as input data.

You should print lines of output in a neat tabular form as shown in the sample output, by calling the functions and without using repetition instructions.

Note that the above formula implies that  $^{\circ}\text{F} = (9.0 * ^{\circ}\text{C}/5.0) + 32.0$ .

**Rules to follow:**

You are required to write two C programs using top-down design approach (that is, using functions as specified in the given structure chart) for the problem described below. The requirements of the two programs are as follows:

- 1) The first program **should not use any global variables**, must pass **all input parameters using call-by-value parameters** and must pass all **function result parameters using call-by-reference parameters**.
- 2) The second program should declare all main program input and output variables globally and should not pass any parameters in the function calls. Note that a function can still declare needed local variables.
- 3) You are also required to show tracing (**only step 6 and not all 6 problem solving steps**) of your two programs with test data to test for correctness of your solutions.

All input data must be read in the control or main module and all results printed by another function dedicated to printing. **Use the following structure chart for solving the problem.**

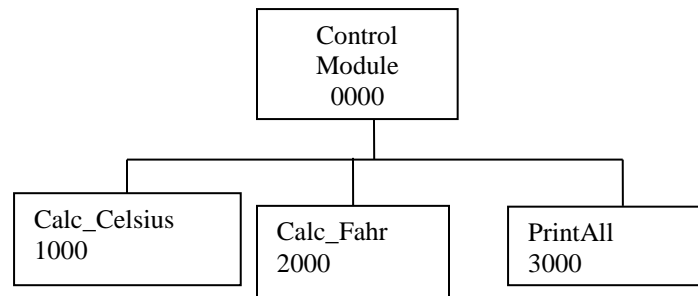
\*\*\*\*\*

All input data should be read in the control module.

Calc\_Fahr module should calculate the Fahrenheit equivalents.

Calc\_Celsius module should be used to calculate the Celsius equivalents.

PrintAll should print all data and tables.



**Sample Input:**

Please type the three fahrenheit temperatures to be converted to Celsius:

68 -40 100

Please type the three Celsius temperatures to be converted to Fahrenheit:

0 -40 20

### Sample Output:

The Temperature Conversions from Fahrenheit to Celsius are

```
Fahrenheit      | Celsius
*****
68.00           | 20.00
-40.00          | -40.00
100.00          | 37.78
*****
```

The Temperature Conversions from Celsius to Fahrenheit are

```
Celsius         | Fahrenheit
*****
0.00            | 32.00
-40.00          | -40.00
20.00           | 68.00
```

Thank You for Using the UWin\_60-140 Temperature Conversion System!!  
Have a Nice Day and Come Again !!

\*\*\*\*\*

Type your tracing of each of your two programs (for testing and verification) for solving this problem in each of the text files called `userid_asn4_prog1.txt` and `userid_asn4_prog2.txt`. Then, type each of the C program solutions into a source file called your `userid_asn4_prog1.c` and `userid_asn4_prog2.c`. Compile your C program with `cc userid_asn4_prog1.c` or `cc userid_asn4_prog2.c`. Then, run (execute it with `./a.out`). When your program is running with no errors, then, create a script file called `userid_script41.txt` and `userid_script42.txt` and hand in your script files for marking. Your script file 1 should show your tracing of source program 1, source program 1, the compilation of your program 1, the running of your program 1 with the input and output data shown and your script file 2 should show your tracing of source program 2, source program 2, the compilation of your program 2, the running of your program 2 with the input and output data shown. The script files can be created as follows:

```
script userid_script41.txt
cat userid_asn4_prog1.txt
cat userid_asn4_prog1.c
cc userid_asn4_prog1.c
./a.out
(when prompted, type in the input data in the correct order)
exit
```

\*\*

```
script userid_script42.txt
cat userid_asn4_prog2.txt
cat userid_asn4_prog2.c
cc userid_asn4_prog2.c
./a.out
(when prompted, type in the input data in the correct order)
exit
```

**Marking Scheme :**

1. Correct Program Solution for program 1 with call-by-reference parameters  
(3 for variable declarations, 2 for function prototypes, 2 for function calls,  
6 for function definitions, 3 for reading and printing, 4 for correct logic) ---(20marks)
2. Correct Program Solution for program 2 with global variables  
(3 for variable declarations, 2 for function prototypes, 2 for function calls,  
6 for function definitions,3 for reading and printing, 4 for correct logic) ----(20marks)
3. Hand tracing the programs for testing and evaluation (step 6)  
(5 marks for correct tracing of program 1, and 5 marks for correct  
tracing of program 2) -----(10 marks)

.....