

CptS 111 — Homework #2

Due Tuesday Jan. 24, 2012 midnight (24:00)

1. Reading Assignment: Read through Chap. 4 of the textbook. **Really!** If you read this material, the rest of this assignment is trivial. If don't read this material, well, just read the material! (Chap. 4 may not be posted until late on Thursday, but the first three chapters should be available now.)

2. Programming Assignment A:

Using IDLE, open a new window. You will write a program in this window and ultimately save your work to a file called `hw2a.py`. At the top of the file/window add comment lines that provide your name, the date, the class, and the assignment. Please call this particular assignment HW 2A. (A skeleton of the code for this program is shown below.)

In this file write a program that prompts the user for a temperature in degrees Fahrenheit and then reports the corresponding temperature in degrees Celsius. A temperature T_f in Fahrenheit can be converted to a temperature T_c in Celsius using the following formula:

$$T_c = \frac{5}{9}(T_f - 32)$$

Your solution to this problem should not use any functions other than built-in Python functions, i.e., it should be a simple collection of statements.

The following demonstrates what should happen when you run the program in IDLE. The user's input is shown in bold. The results of four separate runs are shown.

```
1 >>> ===== RESTART =====
2 >>>
3 Enter temperature in Fahrenheit: 32
4 32.0 Fahrenheit = 0.0 Celsius
5 >>> ===== RESTART =====
6 >>>
7 Enter temperature in Fahrenheit: 212
8 212.0 Fahrenheit = 100.0 Celsius
9 >>> ===== RESTART =====
10 >>>
11 Enter temperature in Fahrenheit: 98.6
12 98.6 Fahrenheit = 37.0 Celsius
13 >>> ===== RESTART =====
14 >>>
15 Enter temperature in Fahrenheit: 70
16 70.0 Fahrenheit = 21.11111111111111 Celsius
```

This implementation is rather cumbersome in that each time we want to rerun the program we have to return to the window that contains the program and type F5. In the next problem we will implement a better solution.

The following is what my solution to this problem looks like except that I have removed all the meaningful statements.

```

1 # John B. Schneider
2 # 1/17/2012
3 # CptS 111
4 # HW 2A
5 #
6 # A program to convert Fahrenheit to Celsius.
7
8 ### LINES TO ACCOMPLISH THIS DELETED! (Three lines deleted.)

```

3. Programming Assignment B:

For this problem you must write another version of the program to convert a temperature in Fahrenheit to a temperature in Celsius. In this version the statements are organized into two functions.

One function is called `temp_f_to_temp_c()`. It takes a single argument which is a temperature in Fahrenheit (it is assumed to be an `int` or a `float`). This function *returns* the corresponding temperature in Celsius. **Note that this function does not generate any output**, i.e., it does not print anything.

The other function is called `main()`. This function takes no arguments and it does not return anything. This function prompts the user for a temperature in Fahrenheit. It then uses the `temp_f_to_temp_c()` function to convert this temperature to Celsius. Finally, it displays the original Fahrenheit temperature and the corresponding Celsius temperature. The output of this program is identical to that of the previous program (nevertheless, we will consider a sample run of this program below).

The last statement in the file should be a call to the `main()` function.

The following provides the framework that I used in my solution to this problem. The bodies of the two functions have been removed, but all the other code is provided.

```

1 # John B. Schneider
2 # 1/17/2012
3 # CptS 111
4 # HW 2B
5 #
6 # A program to convert Fahrenheit to Celsius in which the statements
7 # are organized in terms of functions.
8
9 def temp_f_to_temp_c(tf): # Convert tf from Fahrenheit to Celsius.
10     ##### LINE OR LINES DELETED!
11
12 def main():
13     ##### LINE OR LINES DELETED!

```

```

14
15 # Run the main() function.
16 main()

```

The following demonstrates what should happen when you run this program in IDLE. The user's input is again shown in bold. Note that in this case the interpreter was only restarted once (i.e., F5 was only typed once). See the additional comments below.

```

1 >>> ===== RESTART =====
2 >>>
3 Enter temperature in Fahrenheit: 32
4 32.0 Fahrenheit = 0.0 Celsius
5 >>> main()
6 Enter temperature in Fahrenheit: 0
7 0.0 Fahrenheit = -17.77777777777778 Celsius
8 >>> temp_f_to_temp_c(212)
9 100.0
10 >>> tc = temp_f_to_temp_c(212)
11 >>> print("The return value of the function was:", tc)
12 The return value of the function was: 100.0
13 >>> main()
14 Enter temperature in Fahrenheit: 72
15 72.0 Fahrenheit = 22.22222222222222 Celsius

```

When this program is executed, `temp_f_to_temp_c()` and `main()` are defined. The last statement of the program calls `main()` which immediately starts execution of the program. This is evident in line 3 shown above where the user is prompted for input even though seemingly no command was issued (of course, F5 was typed which got the ball rolling). After one run of the program, the two functions `temp_f_to_temp_c()` and `main()` are available for us to call again if so desired. So, in line 5 `main()` is called and this reruns the program.

In line 8 `temp_f_to_temp_c()` is called with the argument set to the literal integer 212. We see the return value of 100.0 on line 9 solely because the interactive environment echoes the value of an expression if the value is not assigned to anything (again, `temp_f_to_temp_c()` does not produce any output—it merely returns a value). To further illustrate this fact, consider the statement in line 10. Here we again call `temp_f_to_temp_c()` with the literal integer argument of 212, but in this case we do not see the result since the return value is assigned to the variable `tc`. The `print()` statement on the next line is used to display the result.

The call to `main()` on line 13 simply serves to emphasize we can call `main()` as many times as desired.

Submitting Your Work

Submit program `hw2a.py` together with `hw2b.py`. Your submission should be in accordance with

how you submitted your work in the first lab. *Be sure to bundle things together in a zip file before submitting your work.* As a reminder, you can find a link to the submission page at the course Web site.