HomeWork 1 - CS5007

Thierry Petit

TPetit@WPI.edu

Learning objectives

- Learn about variables and expressions.
- Understand basic mathematical and logical operators.
- Define a function and call it to perform a calculation.
- Print information on the output window.
- First use of If/else statement.

Dates

- Assigned: May 25, 2020, 6PM.
- Due: June 2, 2020, 6PM.

The basic assignment is worth 100 points and the extra credit option is worth up to 5 additional points. Details are provided in this document. This is an individual assignment. You may discuss any aspect of this assignment with anyone, but you must type everything into an IDLE window yourself. Except where specified, you may never copy and paste from any electronic source.

Write your code and comments on the template `HW1.py` and save your file as `FIRSTNAME-LASTNAME-HW1.py`. Upload it on the course website (no delay accepted).

1 Arithmetic expressions (15 points)

Convert each of the following mathematical expression in an equivalent Python expression. Write the expression in the designated space in your template module a line of Python code.

The values of the three rational variables a, b and c are assigned at the beginning of the template module and should not be modified.

a. $4ab + bc$

b. b^3

c. $\sqrt{ab}$

d. $(ac + ba)/bc$

e. $a^2 - c^4 + 2ab$

2 Expressions, Python operators (5 points)

Write the following Python expressions into the designated places in the template module. Run the module.

a. $(8//6) * 6$

b. $(8/6) * 6$

Explain in a comment line why the value of expressions **a.** and **b.** are different.

3 Boolean expressions (20 points)

Convert each of the following logical expression in an equivalent Python expression. Write the expression in the designated space in your template module a line of Python code.

The values of the Boolean variables a, b and c are assigned at the beginning of the template module and should not be modified. Recall that $\vee$ is "or", $\wedge$ is "and", $\neg$ is "not" and $\rightarrow$ is implication.

a. $a \wedge (b \vee c)$

b. $b \rightarrow (a \vee c)$

c. $(a \wedge b) \vee (\neg c)$

d. $a \rightarrow (b \rightarrow (\neg c))$

4 Function (30 points)

Wind chill is defined by the following formula:

$$\text{Wind Chill} = 13.13 + 0.628 \times T - 12.1 \times V^{0.15} + 0.3967 \times T \times V^{0.155}$$

In this formula, T is the temperature in Celsius and V is the wind velocity in kilometers per hour.

1) Define a function `WindChill` in your template module to calculate the wind chill for any temperature and velocity. For instance, a call to `WindChill`$(-20, 30)$ prints the following result:

```
At -20C and 30 kph winds it feels like -33.02508884265753C
```

Your function should NOT return anything, but PRINT the result on IDLE. Call your `windChill` function with arguments -20 for T and 30 for V.

2) Write a similar function, named `WindChill2`, that does not print but returns the result. Write an appropriate statement using this second function, so as to print this returned result.

5 Errors in a Python program (30 points)

```
def euc(x1,y1,x2,y1):
   lambda = (x1-x2)**2
   gamma = (y1-y2)**2
return((lambda+gamma)**(1//2))
result = euc(1,3,2,7)
print("result")
```

Function `euc` was written to calculate a distance with following formula:

$$\sqrt{((x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Unfortunately, the function definition, call and print statement contain 5 errors. Explain in a command line each error, within the template module. Re-write correctly the function and the call/print statement in the template module.

6 Extra credit

This part is not mandatory but may provide extra credits, up to 5 points. Consider the following Python function.

```
def myFct(value):
   if(value==1):
      return value
   else:
      return value+myFct(value-1)
```

Explain in a comment line the result returned by the function (2 points). You should not paraphrase the code (this will not give you points), but rather explain with one simple sentence what is the result of a call to this function.

▷ *You may print the result of the function call with values* 3, 4 *and* 5 *(*i.e., *successive calls), for instance. In addition, observe that the function is recursively called by itself in the else statement. What is the effect of this instruction?*

Write a simpler function `myFct` in the template module, that returns exactly the same result, but encoded without using `if` and `else` and without the *recursion* call `myFct(value-1)`. Your function must NOT use more advanced concepts such as while and for loops. (3 points).