**CS 153 / 453**
**Lab Assignment #5**

Do your own work. Your grade on exams depends heavily on how well you understand the material in the lab assignments.

Working together is not permitted.

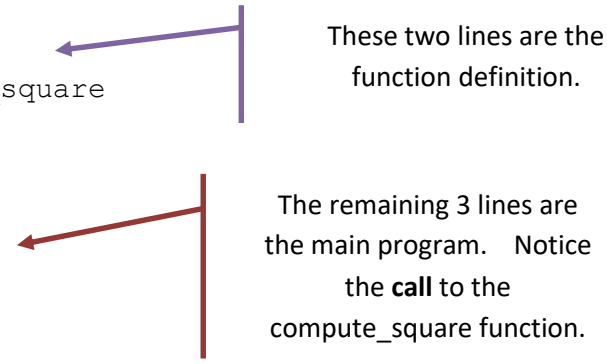
All code you submit must be your own individual work.

From the textbook, chapter 6.

```
def compute_square(num_to_square):
    return num_to_square * num_to_square


num_squared = 0
num_squared = compute_square(7)

print('7 squared is', num_squared)
```




From section 6.4.

A function may accept different types of parameters. Having different behaviors (depending on a parameter's type) is called polymorphism. The book gives the example:

```
def add( x, y ):
   return x + y
```

This function is polymorphic. It behaves one way (adds) if the values of x and y are numeric. It behaves a different way (concatenates) if the values of x and y are strings.

Python uses dynamic typing. The type of parameters, variables, and functions is dynamic (they can change types).

## Create a program named lab5.py.

- All 3 functions and the main program, which tests the functions, go in one file.
- Place the code for all 3 functions first, in the order that they are listed below, then place the code for the main program.
- Header comments and inline comments are required in all programs.
- Programs that do not compile will receive a grade of zero.

1. **Function #1:** Write a function named **print5** that will accept one parameter. The parameter contains a string. The function should print the string on multiple lines, 5 characters per line. The last line of output may not have 5 characters.

   Example, if the string is "extemporaneous", the output would be:

   ```
   extem
   poran
   eous
   ```

   The function does not return any value.

2. **Testing Function #1:** In the main part of the program, print the message "Testing Function #1" followed by a blank line.

   Input a string from the user then **call** the print5 function to display the string, 5 characters per line.

   After the function call, print another blank line.

   Test thoroughly. Debug if needed. When it works correctly, go on to the next function.

3. **Function #2:** Write a function named **triangleType** that accepts three parameters. The parameters represent the lengths of the sides of a triangle.

   The function should return a string. The return values that are possible are: "equilateral" (all sides are equal), "isosceles" (two sides are equal but the third side is different), or "scalene" (all three sides are different).

4. **Testing Function #2:** In the main part of the program, add lines (don't remove the lines that tested function 1).

   Print the message "Testing Function #2" followed by a blank line.

   Call the triangleType function at least 3 times with different values for the parameters. Print the result of each call.

After the function calls, print another blank line.

Test thoroughly. Debug if needed. When it works correctly, go on to the next function.

5. **Function #3:** Write a function named **pumpSchedule** that will help monitor water needs and pump schedules for a large water tank. The parameters for the function are 1) tank capacity (gallons), 2) current level (a value that indicates current water level, as a percentage of the tank capacity. For example, a current level of 100 means that the tank is completely full.), and 3) usage rate (gallons per day).

   A pump that refills the tank must be switched on when the current level drops below 50 (half full).

   The function should calculate and return the number of days (int) that the tank can operate at the current usage rate before the pump must be turned on.

6. **Testing Function #3:** In the main part of the program, add lines (don't remove the lines that tested functions 1 and 2).

   Print the message "Testing Function #3" followed by a blank line.

   Call the pumpSchedule function at least 3 times with different values for the parameters. Print the result of each call.

   Test thoroughly. Debug if needed. When it works correctly, check to make sure that you have documented the program with header comments and inline comments.

## Submit lab5.py on Canvas.

Note: CS 153 and 453 have the same assignment for Lab 5.