# Lab Exercise #7

**Assignment Overview**

This lab exercise provides practice with functions and exceptions in Python.

You will work with a partner on this exercise during your lab session. Two people should work at one computer. Occasionally switch the person who is typing. Talk to each other about what you are doing and why so that both of you understand each step.

**Part A: Defining and Calling Functions**

1. Examine the Python program below and predict the values which will be displayed.

```
def total( start, end ):

    tot = 0

    n = start
    while n <= end:
        tot += n
        n += 1

    return tot

A = total( 0, 10 )

print( "Value of A:", A )            # Value of A: ____________

B = total( 8, 12 )

print( "Value of B:", B )            # Value of B: ____________

C = total( 15, 15 )

print( "Value of C:", C )            # Value of C: ____________

D = 3 * total( 4, 7 ) + 10

print( "Value of D:", D )            # Value of D: ____________

X = 5
E = total( 2*X, 12 )

print( "Value of E:", E )            # Value of E: ____________
```

2. After completing (1) above, download and execute the program (“lab07.parta.py”) to check your predictions. If any of your answers are incorrect, re-work the appropriate questions.

**Part B: Practice with Parameters**

1. Examine the Python program below and predict the values which will be displayed

```
def test( a, b ):

    a = 7

    print( "In test, value of a:", a )    # Value of a: _____________

    print( "In test, value of b:", b )    # Value of b: _____________

    print( "In test, value of x:", x )    # Value of x: _____________


a = 1
b = 2
x = 5
y = test( a, b )

print( "In main, value of a:", a )        # Value of a: _____________

print( "In main, value of b:", b )        # Value of b: _____________

print( "In main, value of x:", x )        # Value of x: _____________

print( "In main, value of y:", y )        # Value of y: _____________
```

2. After completing (1) above, download and execute the program ("lab07.partb.py") to check your predictions. If any of your answers are incorrect, re-work the appropriate questions.

**Part C: Exceptions**

1. Consider the following Python program:

```
def get_integer( prompt ):
    value_str = input( prompt )
    try:
        value_int = int( value_str )
    except ValueError:
        print ( "** Invalid input, assuming 0 **" )
        value_int = 0
    return value_int

def main():
    try:
        numer = get_integer( "Enter the numerator: " )
        denom = get_integer( "Enter the denominator: " )
        print( numer, "divided by", denom, end=" " )
        result = numer/denom
        print( "yields", result )

    except ZeroDivisionError:
        print( "** Invalid: attempted to divide by zero **" )

    print( "Program halted" )

main()
```

Predict what will be displayed when the user enters the following pairs at the prompts:

```
30    10
0     10
30     0
3d    10
30    e0
```

2. Download the program ("lab07.partc.py"), then repeatedly execute the program to check your predictions. Note that you can re-execute the program by simply typing "`main()`" at the shell prompt.

3. Revise the program to move the "`except ValueError`" suite from `get_integer()` into the main function's `except` suite. You will have to delete "`try`" from `get_integer()`. Describe how the program behaves differently.

★ **Demonstrate your completed program to your TA and (important!) describe how the behavior changes. On-line students should submit the completed program (named "lab07.partc.py") for grading via the CSE handin system, and include a description of how the behavior changed as a comment in the program.**

## Part D: Programming with Functions

Develop a Python program which will calculate the sum of an integer series, as described below.

The program will consist of function `squares`, function `cubes`, and the main block of code.

Function `squares` has two parameters: the initial integer number in the series and the number of terms in the series. It will use repetition to compute the sum of the series, and then return the results. For example, if the first parameter is 2 and the second parameter is 4, the function will return 54.

$$\text{Sum} = 2^2 + 3^2 + 4^2 + 5^2 = 54$$

Function `cubes` has two parameters: the initial integer number in the series and the number of terms in the series. It will use repetition to compute the sum of the series, and then return the results. For example, if the first parameter is 3 and the second parameter is 2, the function will return 91.

$$\text{Sum} = 3^3 + 4^3 = 91$$

The program will repeatedly prompt the user to enter a command (a character string). The program will halt and display the message "Program halted normally" when the command is "exit".

When the command is "squares", the program will prompt the user to enter the initial integer number in the series, and then the number of terms in the series. It will call function `squares`, and then display the results.

When the command is "cubes", the program will prompt the user to enter the initial integer number in the series, and then the number of terms in the series. It will call function `cubes`, and then display the results.

The program will display the message "*** Invalid choice ***" if the user enters an invalid command.

Function `squares` and function `cubes` will use repetition to compute the sums (rather than using a closed form equation).

You may assume that the user enters integer numbers when prompted for numeric values.

A suggestion: use incremental development (start with a simple version of the program, then extend it). When developing functions, it is often useful to start with a stub (a definition of the function which is syntactically complete, but logically incomplete). The following is a stub for function `squares`:

```
def squares( start, num ):
     return 0
```

The stub is a complete function definition, but it doesn't do the calculations yet.

★ **Demonstrate your completed program to your TA. On-line students should submit the completed program (named "lab07.partd.py") for grading via the CSE handin system.**