

## Lab 9 – Functions

Chapter:	6. Functions
Time:	80 Minutes

### Lab 9

### Objectives

- To define functions with formal parameters.
- To invoke functions with actual parameters.
- To distinguish between functions that return and do not return a value.
- To invoke a function using positional arguments or keyword arguments.
- To develop reusable code that is modular and is easy to read, debug, and maintain.
- To define functions with default arguments.

### Current Lab Learning Outcomes (LLO)

By completion of the lab the students should be able to

- Use `def` keyword to define functions.
- Write value-returning functions (functions with return values).
- Write void (None) functions (functions without return values).
- Invoke (call) value-returning functions.
- Invoke (call) void functions.
- Pass arguments to a function using positional arguments or keyword arguments.
- Develop reusable code.
- Define functions with default arguments.

### Lab Requirements

- PyCharm (IDE).



## Practice Activities with Lab Instructor (30 minutes)

### Problem 1

Programming Exercises (6.5)

Write the following function to display three numbers in increasing order:

Function Header	def displaySortedNumbers(num1, num2, num3)
Parameters	The three parameters ( <b>num1</b> , <b>num2</b> , <b>num3</b> ) are numbers.
Return Value	None

Write a test program that prompts the user to enter three numbers and invokes the function to display them in increasing order.

Here are some sample runs:



```
Enter three integers: 3, 2.4, 5 <enter>
The sorted numbers are 2.4 3 5
```



```
Enter three integers: 100, 300, 200 <enter>
The sorted numbers are 100 200 300
```

### Solution

#### Phase 1: Problem-Solving Phase:

See comments on the code below.

#### Phase 2: Implementation Phase:

1. Create a new project and name it "Lab 9".
2. Create a new file and name it "activity\_1.py".
3. Write the following code in the file:

activity\_1.py

```
1 def main():
2     num1, num2, num3 = eval(input("Enter three integers: "))
3
4     # Invoke the displaySortedNumbers method to display the
5     # numbers in increasing order
6     displaySortedNumbers(num1, num2, num3)
7
8
9 def displaySortedNumbers(num1, num2, num3):
10    # Write the code to implement this method
11    if num1 > num2:
12        num1, num2 = num2, num1
13
14    if num2 > num3:
15        num2, num3 = num3, num2
16
17    if num1 > num2:
18        num1, num2 = num2, num1
19
20    print("The sorted numbers are", num1, num2, num3)
21
22
23 # Call the main function
24 main()
```

## Problem 2

Programming Exercises (6.13)

Write a function to compute the following series:

$$m(i) = \frac{1}{2} + \frac{2}{3} + \cdots + \frac{i}{i+1}$$

Function Header	def m(i)
Parameters	The parameter ( <b>i</b> ) is a number, and it represents the variable of the above equation.
Return Value	The result of the above equation. <b>Note:</b> This function does not print anything.

Write a test program that displays the following table:

i	m(i)
1	0.5000
2	1.1667
3	1.9167
...	
18	15.4523
19	16.4023
20	17.3546

## Solution

### Phase 1: Problem-Solving Phase:

See comments on the code below.

### Phase 2: Implementation Phase:

1. Open the project "Lab 9" if it was not opened or create it if it was not existing.
2. Create a new file and name it "activity\_2.py".
3. Write the following code in the file:

activity\_2.py

```
1 def main():
2     print(format("i", "<15s"), format("m(i)", "<20s"))
3     for i in range(1, 20 + 1):
4         print(format(i, "<15d"), format(m(i), "<20.4f"))
5
6 def m(n):
7     sum = 0
8
9     for i in range(1, n + 1):
10         sum += i / (i + 1)
11
12     return sum
13
14
15 main()
```



### Problem 3

Programming Exercises (6.17)

Create the following two functions:

Function Header	def isValid(side1, side2, side3):
Parameters	The three parameters (side1, side2, side3) are numbers.
Return Value	It returns True if the sum of any two sides is greater than the third side. Otherwise, it returns False. <b>Note:</b> This function does not print anything.

Function Header	def area(side1, side2, side3):
Parameters	The three parameters (side1, side2, side3) are numbers.
Return Value	It returns the area of the triangle using the following formula: $area = \sqrt{p \times (p - side1) \times (p - side2) \times (p - side3)}$ Where p is half the perimeter: $p = \frac{side1 + side2 + side3}{2}$ <b>Note:</b> This function does not print anything.

Write a test program that reads three sides for a triangle and computes the area if the input is valid. Otherwise, it displays that the input is invalid:



```
Enter three sides in double: 1, 3, 1 <enter>
Input is invalid
```



```
Enter three sides in double: 1, 1, 1 <enter>
The area of the triangle is 0.4330127018922193
```

## Solution

### Phase 1: Problem-Solving Phase:

See comments on the code below.

### Phase 2: Implementation Phase:

4. Open the project "Lab 9" if it was not opened or create it if it was not existing.
5. Create a new file and name it "activity\_3.py".
6. Write the following code in the file:

activity\_3.py

```
1  import math
2
3  def main():
4      edge1, edge2, edge3 = eval(input("Enter three sides in double: "))
5
6      if isValid(edge1, edge2, edge3):
7          print("The area of the triangle is", area(edge1, edge2, edge3))
8      else:
9          print("Input is invalid")
10
11     # Returns true if the sum of any two sides is
12     # greater than the third side.
13     def isValid(side1, side2, side3):
14         return (side1 + side2 > side3) and \
15             (side1 + side3 > side2) and (side2 + side3 > side1)
16
17     # Returns the area of the triangle.
18     def area(side1, side2, side3):
19         p = (side1 + side2 + side3) / 2
20         return math.sqrt(p * (p - side1) * (p - side2) * (p - side3))
21
22     main()
```

## Individual Activities (50 minutes)

### Problem 4

Programming Exercises (6.6)

Write a function to display a pattern as follows:



```

          1
        2 1
      3 2 1
    ...
  n  n-1 ... 3 2 1

```

Function Header	def displayPattern(n):
Parameters	The parameter (n) is a number.
Return Value	None. It displays the above pattern.

Write a test program that prompts the user to enter a number n and invokes displayPattern(n) to display the pattern.

Here is a sample run:



```

Enter line number: 6 <enter>
          1
        2 1
      3 2 1
    4 3 2 1
  5 4 3 2 1
6 5 4 3 2 1

```



## Problem 5

Programming Exercises (6.18)

Write a function that displays an n-by-n matrix using the following header:

Function Header	def printMatrix(n):
Parameters	The parameter (n) is a number.
Return Value	None. It displays an n-by-n matrix.

Each element is 0 or 1, which is generated randomly. Write a test program that prompts the user to enter n and displays an n-by-n matrix.

Here are sample runs:

Enter n: 3 <enter>

```
1 0 0
0 1 1
1 1 1
```

Enter n: 10 <enter>

```
0 1 0 0 0 1 0 0 1 0
1 0 1 0 0 0 1 1 1 0
1 0 1 0 0 0 0 0 0 0
0 0 1 1 0 0 0 1 0 0
0 0 1 1 0 0 1 1 1 0
0 1 1 0 1 0 1 1 0 0
0 0 0 0 1 1 1 1 1 0
1 1 1 1 1 1 1 1 1 1
1 0 0 0 1 0 0 0 1 0
0 0 1 1 0 0 0 0 0 1
```

## Problem 6

## Programming Exercises (6.8)

Write the following two functions:

Function Header	def celsiusToFahrenheit(celsius):
Parameters	The parameter (celsius) is a number.
Return Value	It returns the degree in Fahrenheit by using the following formula: $fahrenheit = (9 / 5) * celsius + 32$

Function Header	def fahrenheitToCelsius(fahrenheit):
Parameters	The parameter (fahrenheit) is a number.
Return Value	It returns the degree in Celsius by using the following formula: $celsius = (5 / 9) * (fahrenheit - 32)$

Write a test program that invokes these functions to display the following tables:

Celsius	Fahrenheit		Fahrenheit	Celsius
-----				
40	104.00		120	48.89
39	102.20		110	43.33
38	100.40		100	37.78
37	98.60		90	32.22
36	96.80		80	26.67
35	95.00		70	21.11
34	93.20		60	15.56
33	91.40		50	10.00
32	89.60		40	4.44
31	87.80		30	-1.11



## Extra Exercises (Homework)

### From the Textbook

- Programming Exercises:
  - 6.7
  - 6.9
  - 6.11
  - 6.14
  - 6.20

### From MyProgrammingLab (<https://pearson.turingscraft.com>)

- |         |         |
|---------|---------|
| • 6.2   | • 6.3   |
| ○ 51016 | ○ 51073 |
| ○ 51155 | ○ 51143 |
| ○ 51156 | ○ 51074 |
| ○ 51218 | ○ 51144 |
| ○ 51105 | ○ 51075 |
| ○ 51158 | ○ 51076 |
| ○ 51011 | ○ 51077 |
| ○ 51008 | ○ 51146 |
| ○ 51009 | ○ 51145 |
|         | ○ 51147 |
|         | ○ 51148 |
|         | ○ 51149 |

### Upload Your Solutions



Upload your solutions of the lab activities to Blackboard.