# Lab – 6 Functions

Tasks:

1. Consider the program segment below. Save the following program as **a.py**:

**a.py**

print ("Module a is loaded")

print (\_\_name\_\_)

def print\_a():

print("Print function in file a.py is executed")

if \_\_name\_\_ == "\_\_main\_\_":

print\_a()

1. Examine the output of the above program.

Answer:

Module a is loaded

\_\_main\_\_

Print function in file a.py is executed

1. Consider the following program. Save the following program as **b.py** and place it in the same folder with **a.py**:

**b.py**

print ("Module b is loaded")

print (\_\_name\_\_)

import a

def print\_b():

print("Print function in file b.py is executed")

if \_\_name\_\_ == "\_\_main\_\_":

print\_b()

Examine the output of the above program

i) What is the value of \_\_name\_\_ variable in **b.py**?

Answer:

\_\_main\_\_

ii) What is the value of \_\_name\_\_ variable in **a.py**?

Answer:

\_\_main\_\_

iii) Had the function print\_a ( ) called while executing **b.py**?

Answer:

Yes

1. Consider the program segment below:

# Complete the function cube\_of\_n below

# Complete the function show\_instruction below

# Complete the function get\_n below

# Output Generation. You are not allowed to modify the following codes

def main():

show\_instruction()

n = get\_n()

cube\_n = cube\_of\_n(n)

print("Cube of", n, "=", cube\_n)

if \_\_name\_\_ == "\_\_main\_\_":

main()

Complete the THREE methods with reference to the comments in the program segment and output below

Output:

This program computes cube of n

Enter an integer N please: 4

Cube of 4 = 64

Answer:

# Complete the function cube\_of\_n below

def cube\_of\_n(n):

cube\_n = n\*n\*n

return cube\_n

# Complete the function show\_instruction below

def show\_instruction():

print("This program computes cube of n")

# Complete the function get\_n below

def get\_n():

n = int(input("Enter an integer N please:"))

return n

# Output Generation. You are not allowed to modify the following codes

def main():

show\_instruction()

n = get\_n()

cube\_n = cube\_of\_n(n)

print("Cube of", n, "=", cube\_n)

if \_\_name\_\_ == "\_\_main\_\_":

main()

1. Consider the program segment below:

# Complete the function rectangle below

def rectangle(...):

...

return area, surface\_area, volume

# Complete the function circle below

def circle(...):

...

return area, surface\_area, volume

# Output Generation. You are not allowed to modify the following codes

def main():

l = float ( input ( "Enter the length of rectangle:" ) )

w = float ( input ( "Enter the width of rectangle:" ) )

h = float ( input ( "Enter the height of rectangle:" ) )

r = float ( input ( "Enter the radius of circle:" ) )

area\_rectangle, surface\_area\_rectangle, volume\_rectangle = rectangle( l, w, h )

area\_circle, surface\_area\_sphere, volume\_sphere = circle( r )

print ( "Area of rectangle is", area\_rectangle )

print ( "Surface area of rectangle is", surface\_area\_rectangle )

print ( "Volume of rectangle is", volume\_rectangle )

print ( "Area of circle is", area\_circle )

print ( "Surface area of sphere is", surface\_area\_sphere )

print ( "Volume of sphere is", volume\_sphere )

if \_\_name\_\_ == "\_\_main\_\_":

main()

Complete the TWO methods with reference to the comments in the program segment and output below

Output:

Enter the length of rectangle:1

Enter the width of rectangle:2

Enter the height of rectangle:3

Enter the radius of circle:2

Area of rectangle is 2.0

Surface area of rectangle is 22.0

Volume of rectangle is 6.0

Area of circle is 12.566370614359172

Surface area of sphere is 50.26548245743669

Volume of sphere is 33.510321638291124

Hints: You may assume the value of PI is 3.14, or make use of the math.pi value in math library.

Answer:

# Complete the function rectangle below

def rectangle(l,w,h):

area = l\*w

surface\_area = l\*w\*2 + l\*h\*2 + h\*w\*2

volume = l\*w\*h

return area, surface\_area, volume

# Complete the function circle below

def circle(r):

import math

area = math.pi\*r\*\*2

surface\_area = 4\*math.pi\*r\*\*2

volume = (4/3)\*math.pi\*r\*\*3

return area, surface\_area, volume

# Output Generation. You are not allowed to modify the following codes

def main():

l = float ( input ( "Enter the length of rectangle:" ) )

w = float ( input ( "Enter the width of rectangle:" ) )

h = float ( input ( "Enter the height of rectangle:" ) )

r = float ( input ( "Enter the radius of circle:" ) )

area\_rectangle, surface\_area\_rectangle, volume\_rectangle = rectangle( l, w, h )

area\_circle, surface\_area\_sphere, volume\_sphere = circle( r )

print ( "Area of rectangle is", area\_rectangle )

print ( "Surface area of rectangle is", surface\_area\_rectangle )

print ( "Volume of rectangle is", volume\_rectangle )

print ( "Area of circle is", area\_circle )

print ( "Surface area of sphere is", surface\_area\_sphere )

print ( "Volume of sphere is", volume\_sphere )

if \_\_name\_\_ == "\_\_main\_\_":

main()

1. Write a function **sum\_to\_n** that takes an integer parameter **n**. It then returns the sum of all integers from 1 to n. The following code segment has been given for you.

# Complete the function sum\_to\_n below

# Output Generation. You are not allowed to modify the following codes

def main():

input\_n = int(input("Enter an integer n please:"))

return\_sum = sum\_to\_n(input\_n)

print("Sum of all integers in between 1 to", input\_n, "is", return\_sum)

if \_\_name\_\_ == "\_\_main\_\_":

main()

Output:

Enter an integer n please:9

Sum of all integers in between 1 to 9 is 45

Answer:

# Complete the function sum\_to\_n below

def sum\_to\_n(input\_n):

n = int(input\_n)

sumx = range(n+1)

sumy = 0

for sum in sumx:

sumy = sumy+sum

return sumy

# Output Generation. You are not allowed to modify the following codes

def main():

input\_n = int(input("Enter an integer n please:"))

return\_sum = sum\_to\_n(input\_n)

print("Sum of all integers in between 1 to", input\_n, "is", return\_sum)

if \_\_name\_\_ == "\_\_main\_\_":

main()

1. Write a function **sum\_m\_to\_n** that takes two integer parameter **m** and **n**. It then returns the sum of all integers in between m and n. The following code segment has been given for you.

# Complete the function sum\_m\_to\_n below

# Output Generation. You are not allowed to modify the following codes

def main():

input\_m = int(input("Enter an integer m please:"))

input\_n = int(input("Enter an integer n please:"))

return\_sum = sum\_m\_to\_n(input\_m, input\_n)

print("Sum of all integers in between", input\_m, "and", input\_n, "is", return\_sum)

if \_\_name\_\_ == "\_\_main\_\_":

main()

Output:

Enter an integer m please:10

Enter an integer n please:15

Sum of all integers in between 10 and 15 is 75

Answer:

# Complete the function sum\_m\_to\_n below

def sum\_m\_to\_n(input\_m, input\_n):

m = int(input\_m)

n = int(input\_n)

sumx = range(m,n+1)

sumy = 0

for sum in sumx:

sumy = sumy+sum

return sumy

# Output Generation. You are not allowed to modify the following codes

def main():

input\_m = int(input("Enter an integer m please:"))

input\_n = int(input("Enter an integer n please:"))

return\_sum = sum\_m\_to\_n(input\_m, input\_n)

print("Sum of all integers in between", input\_m, "and", input\_n, "is", return\_sum)

if \_\_name\_\_ == "\_\_main\_\_":

main()

1. Write a function **factorial** that takes an integer parameter **n**. It then returns the factorial n. e.g. 5! = 1 \* 2 \* 3 \* 4 \* 5 = 120.

# Complete the function factorial below

# Output Generation. You are not allowed to modify the following codes

def main():

input\_n = int(input("Enter an integer n please:"))

return\_value = factorial(input\_n)

print(str(input\_n) + "! =", return\_value)

if \_\_name\_\_ == "\_\_main\_\_":

main()

Output:

Enter an integer n please:5

5! = 120

Answer:

# Complete the function factorial below

def factorial(input\_n):

return 1 if (input\_n==1 or input\_n==0) else input\_n \* factorial(input\_n - 1);

#https://www.javatpoint.com/pyhton-factorial-number

# Output Generation. You are not allowed to modify the following codes

def main():

input\_n = int(input("Enter an integer n please:"))

return\_value = factorial(input\_n)

print(str(input\_n) + "! =", return\_value)

if \_\_name\_\_ == "\_\_main\_\_":

main()

1. Write a function **greeting** that can accept a variable length of argument called **names** and print all arguments value. The following code segment has been given for you.

# Complete the function greeting below

# Output Generation. You are not allowed to modify the following codes

def main():

greeting ("Alice")

greeting ("Sam", "Susan")

greeting ("Kelvin", "Peter", "Mary")

if \_\_name\_\_ == "\_\_main\_\_":

main()

Output:

Hello! Alice

Hello! Sam

Hello! Susan

Hello! Kelvin

Hello! Peter

Hello! Mary

Answer:

# Complete the function greeting below

def greeting(\*name):

for n in name:

print("Hello!"+n)

# Output Generation. You are not allowed to modify the following codes

def main():

greeting ("Alice")

greeting ("Sam", "Susan")

greeting ("Kelvin", "Peter", "Mary")

if \_\_name\_\_ == "\_\_main\_\_":

main()

1. Complete the missing statement below by a lambda function so that the program would generate the given output.

# Complete your power function here

# You are not allowed to modify the following codes

square = power(2)

cube = power(3)

n = int(input("Please input an integer: "))

print ("Square of",n,"is",square(n))

print ("Cube of",n,"is",cube(n))

| Sample Output 1:  Please input an integer: 2  Square of 2 is 4  Cube of 2 is 8 | Sample Output 2:  Please input an integer: 3  Square of 3 is 9  Cube of 3 is 27 |
| --- | --- |

Answer:

# Complete your power function here

from tkinter import N

def power(n):

return lambda a:a\*\*n

# You are not allowed to modify the following codes

square = power(2)

cube = power(3)

n = int(input("Please input an integer: "))

print ("Square of",n,"is",square(n))

print ("Cube of",n,"is",cube(n))

1. With reference to Lab3 Question 6, write a function to calculate the two roots of a quadratic equation () by the formulas and returns an output string with reference to the table below:

The discriminant (Δ), can be used to determine the number of roots. **Your function should return suitable strings** the following cases:

| Cases | Description | Output |
| --- | --- | --- |
| a = 0 | not a quadratic equation | This is not a quadratic equation |
| Δ > 0 | two roots | 2 roots, x1=…, x2=… |
| Δ = 0 | one roots | 1 root, x=… |
| Δ < 0 | no real root | No real roots |

The **default value** for variable a, b, c should be set to **zero** if not specified. The following code segment has been given for you.

# Complete the function quadratic below

# Output Generation. You are not allowed to modify the following codes

def main():

print (quadratic(b=1,c=-1))

print (quadratic(a=1,c=-1))

print (quadratic(a=1,b=-1))

print (quadratic(1,-2,1))

print (quadratic(1,1,1))

print (quadratic(3,-4,1))

if \_\_name\_\_ == "\_\_main\_\_":

main()

Output:

This is not a quadratic equation

2 roots, x1 = 1.0, x2 = -1.0

2 roots, x1 = 1.0, x2 = 0.0

1 root, x =1.0

No real roots

2 roots, x1 = 1.0, x2 = 0.3333333333333333

Answer:

# Complete the function quadratic below

def quadratic(a=0,b=0,c=0):

if a==0:

return "This is not a quadratic equation"

else:

discriminant = (b \* b - 4 \* a \* c)

if discriminant > 0: # two roots

x1= (-b + discriminant \*\* 0.5) / (2 \* a)

x2= (-b - discriminant \*\* 0.5) / (2 \* a)

return f"2 roots, x1 ={x1}, x2 = {x2}"

elif discriminant == 0: # one root:

x= -b / (2 \* a)

return f"1 root, x ={x}"

else: # no roots

return "No real roots"

# Output Generation. You are not allowed to modify the following codes

def main():

print (type(quadratic(b=1,c=-1)))

print (quadratic(a=1,c=-1))

print (quadratic(a=1,b=-1))

print (quadratic(1,-2,1))

print (quadratic(1,1,1))

print (quadratic(3,-4,1))

if \_\_name\_\_ == "\_\_main\_\_":

main()