**Week -3:**

1. A function object is a value you can assign to a variable or pass as an argument. For

example, do\_twice is a function that takes a function object as an argument and calls it twice.

defdo\_twice(f):

f()

f()

Here’s an example that uses do\_twice to call a function named print\_spam twice.

defprint\_spam():

print 'spam'

do\_twice(print\_spam)

1. Type this example into a script and test it.

'''def do\_twice(f):

f()

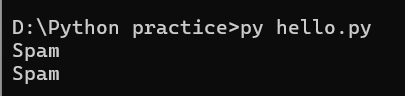
f()

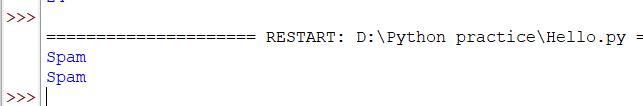
def print\_spam():

print('Spam')

do\_twice(print\_spam)'''

**Output**

 (or)



b. Modify do\_twice so that it takes two arguments, a function object and a value, and   
 calls the function twice, passing the value as an argument.

word = input('What word to repeat?\n')

def do\_twice(f, word):

f(word)

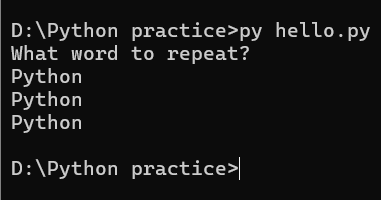
f(word)

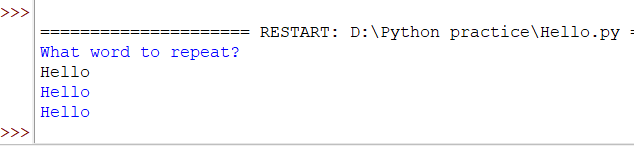
def print\_spam(word):

print(word)

do\_twice(print\_spam, word)

**Output**

 (or)



c. Write a more general version of print\_spam, called print\_twice, that takes a string

as a parameter and prints it twice.

string = "Rama"

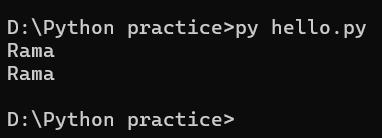
defprint\_twice(string):

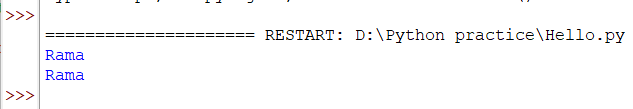
print(string)

print(string)

print\_twice(string)

* OUTPUT

 (or)



d. Use the modified version of do\_twice to call print\_twice twice, passing 'spam' as

an argument.

def do\_twice(f, word):

f(word)

f(word)

defprint\_twice(string):

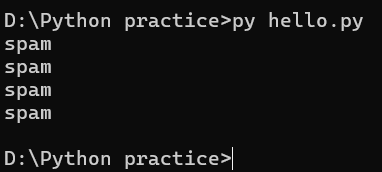
print(string)

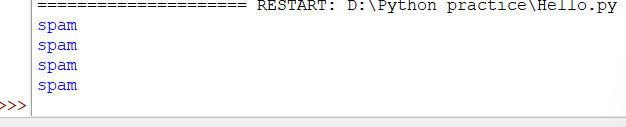
print(string)

newWord = "spam"

do\_twice(print\_twice, newWord)

**Output**

 (or)



1. Write a function that draws a grid like the following:

+ - - - - + - - - - +

| | |

| | |

| | |

| | |

+ - - - - + - - - - +

| | |

| | |

| | |

| | |

+ - - - - + - - - - +

Hint: to print more than one value on a line, you can print a comma-separated

sequence.

def my\_func1():

print("+", 4\*'-', '+', 4\*'-', '+')

def my\_func2():

for x in range(4):

print('|', 4\*' ', '|', 4\*' ', '|')

def total():

my\_func1()

my\_func2()

my\_func1()

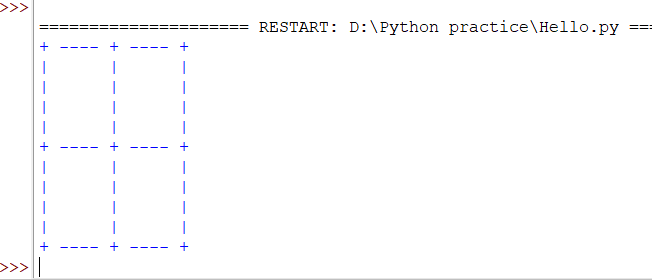
my\_func2()

my\_func1()

total()

**Output:**

 (or)



1. Write a function called gcd that takes parameters a and b and returns their

greatest common divisor.

def gcd(a, b):

     if(b == 0):

         return a

     else:

         return gcd(b, a % b)

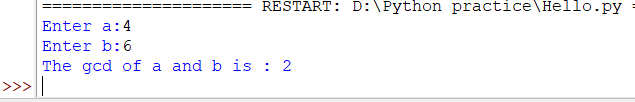
a = int(input(“Enter a:”))

b = int(input(“Enter b:”))

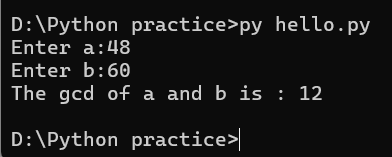
print("The gcd of a and b is : ", end="")

print(gcd(a,b))

**Output**



(or)



1. Write a function called is\_palindrome that takes a string argument and returns True if it is a palindrome and False otherwise. Remember that you can use the built-in function len to check the length of a string.

defisPalindrome(str):

# Run loop from 0 to len/2

for i in range(0, int(len(str)/2)):

if str[i] != str[len(str)-i-1]:

return False

return True

# main function

s = input("Enter a String to check for Palindrome:")

ans = isPalindrome(s)

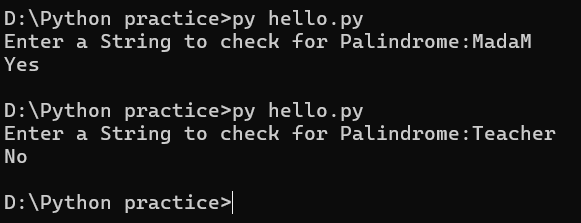
if (ans):

print("Yes")

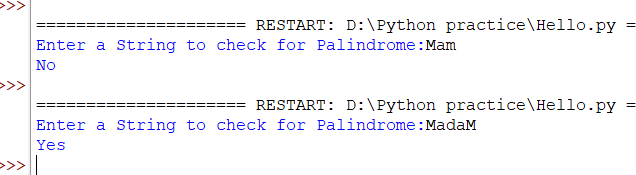
else:

print("No")

**Output:**



(or)



**For Practice**

1. Write a Python program to find the area of a triangle.
2. Write a Python program to swap two variables.
3. Write a Python program to convert Celsius to Fahrenheit.
4. Write a Python program to convert kilometers to miles.
5. Write a Python program to swap two variables without temp variable.
6. Write a Python Program to Check if a Number is Positive, Negative or Zero.