**Task#8**

Void functions are generated and used in the same manner as value-return functions, except they do not return a value after the function executes. A void function executes an operation, and then the power returns to the caller—but does not return a value.

Example:

# Void functions & None

# Void function does not return anything (meaningful)

# Void function is used for its side effects

# Non-void functions return something

>>>x= print(“25”) #void function

Output:25

>>>x

>>>

#x was defined but is does not have any value

#above, returned nothing (also no error is present)

#will try with y (is not defined yet)

>>>y #will occur error

Traceback (most recent call last):

Name Error: name ‘y’ is not defined

#next example

>>>print(x)

None

#it returned None, it means x is not associated with anything meaningful

# from there we should know that print() function is a void function.. It returns None

# void functions return = None = is equivalent to not returning anything

# Non-void Functions & Value returning functions

# Non-void Functions return something other than None.

# May or may not have side effects.

>>>i = int(“25”) #we assigned 25 to i

>>>print(i) # print shows us what i is

25

#abother example

>>>S = input(“Enter any number: “)

Enter any number: 25 #let`s insert 25

>>>print(S)

25 #output of string

# we should also know that empty string is different from None

# string None is different from key word None

>>> S = input(“Enter any number: “)

Enter any number: #this time we will not assign any value (leave it empty)

>>>s

“” #it is empty string now, but it is not None

>>>type(S) #let`s check its type

<class ‘str’> # for sure it is string

>>> S = input(“Enter any number: “)

Enter any number: None #None is user input (not keyword)

>>>S

“None”

#input always returns string

>>>type(S)

<class ‘str’>

A value – returning function not only performs a task, but it also transfers a value back to the code that called it.