ID: 14252

NAme: khalid malik.

PH1

QNO 1.

(a) Function in python. You use functions in programming to bundle a set of instruction that you repeatedly or that, because of their complexity are better self-contained in a sub-program and called when needed. that means that a function is a piece of code written to carry out a specific task.

* Reducing duplication of Code.
* Improving clarity of the code
* Decomposing complex problem into simpler pieces.

In mathematics, a function is a relation between sets that associated to every element of a first set exactly one element of the second set.

* It is easy to locate and isolate a faulty function for further investigator.

Q NO 1
part (6)

Arguments. the arguments of a functions are defined With the def statement. like all other variable in python there is no explict type associated with the function arguments. The fact is important to Consider when making assumptions about the types of data that your function will receive.

```
1 # define a function
2 def Func1 ( ):
  print ("I am learning python function")

  Func1()                    → function Call
# print func1()
# Print Func1
```

OutPut

I am learning python function ..... function out put

Q NO 2.
part a.

Upper () uses.

Return value from string upper().

The upper() method returns the uppercased string from the given It. Converts all lower case characters to uppercase. If no lowercase characters exist it. it returns the original string.

Lower().

The lower() return the lowercased string from the given string. It Convert all uppercase character to Lowercase.

Swapcase()

The Swapcase() method returns a String where all the upper case letter are lower case and vice versa.

Converts all uppercase character to lowercase and all character to uppercase.

Capitalize ().

Capitalize () are used to convert first character of a string to uppercase letter and lowercase all other characters, if any.

Sytax.

String. capitalize ()

Part b.

```
String = " This should BE LOWERCASE!"
Print (string. lower())

# String with numbers
# all alphabet whould be lowercase
String = " This should Be Lowercase
print (string. lower ())
```

Output Put.

This should be lowercase
This should Bs lowercase.

Capitilaze ().

```
String = "Python is AWesome.

Capitalized_String = String.capitalize()

Print('Old String =', String)
print('Capitalized =', Capitalized_string)
```

Output.

```
Old String = Python is AWesome.
Capitalized String = Python is awesome.
```

The rules for a function.

* The must start with a letter or an underscore:-
* They should be lowercase.
* They can have numbers.
* They can be any (within reason) but keep them short.
* They can't be the same as a python keyword. they can have the same as an existing function (including a built-in) but avoid this for now.

Simple rule to define a function.

* def keyword is used to start the function in python.
* def keyword is followed by function-name ~~to indicate the start of function~~ which is followed by parenthesis which contains are arguments passed the user (if any).
* Colon is used at the end of the function name to indicate the start of function body from the next line.
* After adding the colon, start the body of the function with an indented block in the new line.

Part (b).

```
def Square (x):
    y = x * x
Print (Square (4))
```

Output.

16

None

```
def Square (x):
    return x * x

Print (Square (4))
```

Outputput

16

```
def multiply (x, y):
    Print (x * y)
multiply (2, 8)
```

Output

16

Q NO 4.

Here are simple rules to define a function in python. function blocks begin with the keyword def followed by the function name and parantheses (()). Any input parameters or argument should be placed within these parentheses. you can also define parameter inside these parentheses.

If you pass immutable arguments like integers, string or tuples to a function, the passing acts like call-by-value The object reference is passed to the function parameters they can't be changed within the function, because they can't be changed at all i.e They are immutable.

Passing Part (b).

Passing function.

```
def shout (text):
    return text.upper()
Print (shout ('Hello'))
    yell = shout
    Print (yell ('Hello'))
```

Output
HELLO
HELLO
* — * — * — *

```
def shout(text):
    return text.upper()
def whisper (text):
    return text.lower()
def greet (Funct):
# storing the funtion in variable
greeting = function ("Hi, I am created by a
  function passed as an argoment
    print (greeting)
    greet (shout)
    greet (wishper)
```

Output.

HI, I AM CREATED BY A FUNCTION PASSED AS AN ARGUMENT.
hi, i am created by A function passed AS AN argument.

Q QNO. 5

Part a.

A return is a value that a function returns to the calling script or function when it completes it task A return value can be any one of the four variable types. handle integer, object, or string the type of value your function return depends largely on the task it performs.

you Can't return two values However you Can return a single value that is a struct that contains two values You Can return only one thing from a function. Either you make a struct which contain all the things you want to return or you pass some function parameters by reference.

Return multiple values from a function Call. Something missing from many other language. In the case the return values should be a Comma-separated list of values and python then constructs a tuple and return this to the Caller

Part (b).

```
def my function (x):
    return  5 * X

Print (my_Function(3))
Print (my-function(6))
print (my-function(7))
print (my-fuction(10))
```

Output.

15
30
35
50.

* * * * *

```
def Square (x,y):
    return x*x, y*y

    t = Square (2,3)
print(t)    # produce (4,9)
```