

Name : Muhammad Asif

ID: 13161

Paper : Data Sciences

Submitted To : Sir Ayub

Q1. a. Why Functions are used discuss in detail?

Ans No.1(a):

Functions:

A series of statements that take inputs and perform specific tasks to give output is called a function. Here's what a general syntax of a Python function looks like:

```
def function name (parameter list):  
statements, i.e. the function body { }
```

Now why functions are important:

Functions and procedures are the basic building blocks of programs. They are small sections of code that are used to perform a particular task, and they are used for two main reasons. The first reason is that they can be used to avoid repetition of commands within the program. Functions allow you to break a program into more manageable pieces.

The most basic benefit of having a concept of function is that **(1)** it allows us to use the same function more than once in an expression with different parameter.**(2)** It is needed for better readability of the code as well as it can be used countless times so we do not have code from the scratch.**(3)** It also reduces size of the code significantly and **(4)** set of duplicate code is replaced with functions chunks when the function is called.

b. How arguments are used in function , write a simple program in Python?

Ans No.1(b):

Arguments are passed by value; that is, when a function is called, the parameter receives a copy of the argument's value not its address; however, because arguments can be addresses or pointers, a function can use addresses to modify the values of variables defined in the calling function.

Example:

```
def greet(name):
```

```
    /*This function greets to
    the person passed in as
    a parameter*/
    print("Hello, " + name + ". Good morning!")
```

Q2. a. Why .upper(),.lower(),capitalize() and .swapcase() function are used ?

Ans No.2(a):

Isupper();

This function is used to check whether the given string contains any uppercase letter or not and also if we have one character as an input then it checks whether the character is in uppercase or not.

Islower();

It takes a single argument in the form of an integer and returns a value of type int. Internally, the character is converted to its ASCII value for the check.

Capitalize();

Functions used to convert the first character into Uppercase and remaining characters to Lowercase and return a new string.

Swapcases();

The method swapcase() returns a copy of the string in which all the case-based characters have had their case swapped. This automatically ignores non-alphabetic characters.

b. Write a program in which the discussed functions are used.

Ans No.2(b):

Isupper();

```
string = 'MUHAMMAD JOHAR SHAH'
```

```
print(string.isupper())
```

Output:True

```
string = ' Muhammad johar shah '
```

```
print(string.isupper())
```

Output:False

Islower();

string = 'muhammad johar shah'

Output : True

string = 'Muhammad Johar Shah'

Output: False

Capitalize();

```
name = "muhammad Johar shah"  
print(name.capitalize())  
name1="muhammad"  
name2= "johar"  
name3="shah"  
print((name1 + name2  
      + name3).capitalize())
```

Output:

```
muhammad johar shah  
MuhammadJoharShah
```

Swapcases();

```
string = "MUhaMmAd JoHaR ShAh"  
print(string.swapcase())  
string = "National "  
print(string.swapcase())
```

Output:

```
muHAmMaD jOhAr sHaH  
nATIONAL
```

Q3. a. What are the rules for defining the function?

Ans No.3(a):

Rules for Function Defining:

You can define functions to provide the required functionality. Here are simple rules to define a function in Python.

- Function blocks begin with the keyword **def** followed by the function name and parentheses ().
- Any input parameters or arguments should be placed within these parentheses. You can also define parameters inside these parentheses.
- The first statement of a function can be an optional statement - the documentation string of the function or docstring.
- The code block within every function starts with a colon (:) and is indented.
- The statement `return [expression]` exits a function, optionally passing back an expression to the caller. A return statement with no arguments is the same as `return None`.
- A function name to uniquely identify the function. Function naming follows the same rules of writing identifiers in Python.

b. Write a suitable program of our defined function in Python?

Ans No.3(b):

Program for defining functions in Python:

General Syntax of Function

```
def function_name(parameters):  
    "docstring"  
    statement(s)
```

Example 1:

```
/* Python program to multiply all values in the  
list using traversal*/  
def multiplyList(myList) :  
    //Multiply elements one by one  
    result = 1  
    for x in myList:  
        result = result * x  
    return result
```

```
# Main code
list1 = [1, 2, 3]
list2 = [3, 2, 4]
print(multiplyList(list1))
print(multiplyList(list2))
```

Output:

```
6
24
```

Example 2:

```
/* Python program to find smallest
number in a list*/

//list of numbers
list1 = [10, 20, 4, 45, 99]

//sorting the list
list1.sort()

//sort() is a built-in function that sorts numbers
//printing the first element
print("Smallest element is:", *list1[:1])
```

Output:

```
smallest element is: 4
```

Q4. a. What are the rules for defining the function and Parameter passing to the function?

Ans No.4(a):

Functions and Parameters:

Arguments are passed by value; that is, when a function is called. The parameter receives a copy of the argument's value, not its address. This rule applies to all scalar values, structures, and unions passed as arguments. Modifying a parameter does not modify the corresponding argument passed by the function call

A function can take parameters which are just values you supply to the function so that the function can do something utilizing those values. Note the terminology used the names given in the function definition are called parameters whereas the values you supply in the function call are called arguments

b. Write a suitable program of our defined function by parameter passing in Python?

Ans No.4(b):

Example 1:

```
/*Python program for implementation of Insertion Sort
Function to do insertion sort */
def insertionSort(arr):

    // Traverse through 1 to len(arr)
    for i in range(1, len(arr)):

        key = arr[i]

        /* Move elements of arr[0..i-1], that are
        greater than key, to one position ahead*/
        # of their current position
        j = i-1
        while j >=0 and key < arr[j] :
            arr[j+1] = arr[j]
```

```
        j -= 1
    arr[j+1] = key
```

```
//main code
arr = [12, 11, 13, 5, 6]
insertionSort(arr)
print ("Sorted array is:")
for i in range(len(arr)):
    print ("%d" %arr[i])
```

Output:

Sorted array is:

5

6

11

12

13

Example 2:

```
def bubbleSort(arr):
    n = len(arr)

    // Traverse through all array elements
    for i in range(n-1):
        // range(n) also work but outer loop will repeat one time more than needed.

        // Last i elements are already in place
        for j in range(0, n-i-1):
```



```

        /* traverse the array from 0 to n-i-1
        Swap if the element found is greater
        than the next element */
        if arr[j] > arr[j+1] :
            arr[j], arr[j+1] = arr[j+1], arr[j]

# main code
arr = [64, 34, 25, 12, 22, 11, 90]

bubbleSort(arr)

print ("Sorted array is:")
for i in range(len(arr)):
    print ("%d" %arr[i]),

```

Output:

```

11
12
22
25
34
64
90

```

Q5. a. What are return values to a Function discuss in detail?

Ans No.5(a):

Return type in Functions:

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

The return statement is used to send a value back to the calling script or user-defined function. This key word tells JAWS to return the specified value to the calling script or function. You can return the value as a literal value or within a variable.

b. Write a suitable program of a Function with returning value?

Ans No.5(b):

Example of Function with Return Type: 1

```
def f(x, y):  
    z = 2 * (x + y)  
    return z  
  
print("Program starts!")  
  
a = 3  
res1 = f(a, 2+a)  
print("Result of function call:", res1)  
  
a = 4  
b = 7  
res2 = f(a, b)  
print("Result of function call:", res2)
```

Output:

16

22

Example of Function with Return Type: 2

```
def fahrenheit(T_in_celsius):  
    " returns the temperature in degrees Fahrenheit "  
    return (T_in_celsius * 9 / 5) + 32
```

```
for t in (22.6, 25.8, 27.3, 29.8):
```

```
    print(t, ": ", fahrenheit(t))
```

22.6 : 72.68

25.8 : 78.44

27.3 : 81.14

29.8 : 85.64

