Department of Computer Science

Final Exam Spring 2020

NAME: M OMAR MASOOD
ID: 14305
SUBJECT: DATA SCIENCE 5th semester
SUBMITTED TO: SIR AYUB

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

ANSWER(a):

FUNCTION USES:

You use functions in programming to bundle a set of instructions that you want to use 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 specified task. To carry out that specific task, the function might or might not need multiple inputs. When the task is carried out, the function can or can not return one or more values.

There are three types of functions in Python:

Built-in functions, such as help() to ask for help, min() to get the minimum value, print() to print an object to the terminal,… You can find an overview with more of these functions here

- User-Defined Functions (UDFs), which are functions that users create to help them out; And
- Anonymous functions, which are also called lambda functions because they are not declared with the standard def keyword.

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

ANSWER(b): Program:

```
In [50]: my_subject = ("subject name is data science")
         print(my_subject)

         subject name is data science
```

Explanation:

. in this we take a string value and assign in my_subject and then print it.

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

ANSWER(a):

We used these functions to return new values. These functions formats the given string into a nicer output.They allows multiple substitutions and value formatting. This methods let us concatenate elements with a string through positional formatting

1).upper()

The string upper() method converts all lowercase character in a string into upper cases.

Syntax is: string.lower()

2).lower()

The string lower method converts all uppercase character in a string into lowercase characters.

Syntax is: string.upper()

3).capitalize()

The capitalize() method converts first character of a string to uppercase letter and the remaining character in lowercase.

Syntax is: string.capitalize()

4). .swapcase()

The string swapcase() method convert all uppercase characters to lowercase and all lowercase into uppercase characters.

Syntax is: string.swapcase()

b. Write a program in which the discussed functions are used.
Note : Q2 part a functions.

ANSWER(b): Program:

```
In [51]: my_language="pushto"
         print(my_language.upper())

PUSHTO
```

```
In [53]: my_semester="Fifth"
         print(my_semester.lower())

fifth
```

```
In [55]: final_paper="datascience"
         print(final_paper.capitalize())

Datascience
```

```
In [59]: my_name="omar"
         print(my_name.swapcase())

OMAR
```

Explanation:

- In first line we use .upper() it convert all lowercase character into upper.
- In second line we use .lower() it convert all uppercase character into lowercase.
- In third line we use .capitalize() in this method converts first character of a string to uppercase letter and the remaining character in lowercase.
- In fourth line we use .swapcase()this method convert all uppercase characters to lowercase and all lowercase into uppercase characters

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

ANSWER(a):

A function is defined by using the `def` keyword, followed by a name of your choosing, followed by a set of parentheses

which hold any parameters the function will take (they can be empty), and ending with a colon.

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.

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

ANSWER(b):

Program:

```
In [8]: def this_funct():
            a = 20
            print("number inside the function:",a)

        a = 28
        this_funct()
        print("number outside the function:",a)
```

```
number inside the function: 20
number outside the function: 28
```

Explanation:

- In this program we are defining a function so I write the function ((def this_funct)). given a variable with a value and we print that variable into a function (a) ,again given variable with different value and print that value out of function with same variable.

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

ANSWER(a):

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.

Rules:

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.

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

ANSWER(b):

Program:

```
In [61]: def this_Functn(words):
             print(words + "!")
         this_Functn("have a nice day")

have a nice day!
```

Explanation:

- In first line we define a function
- In second "+"this notation is used for concatenation in the above program I am going to concatenate and joint two strings among themselvesand to make them or represent them as a single sentence.

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

ANSWER:

Return values to a Function:

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.
We use the Function Returns edit combo box in the General page of the New Script dialog to tell JAWS the type of value the function returns. We also type the description of the return in the Return description edit box. Adding a description for the return, helps us and anyone else use your function to determine exactly what the value should be used for within the calling script or function.
When we create a new function that returns a string value, the Script Manager places the following function beginning line into our script file:
String Function MyFunction ()

The "string" key word that precedes the "function" key word tells us that the MyFunction function returns a string value to the calling script or user-defined function.

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

ANSWER: **Program:**

```
In [3]: def add (a,b):
            c= a+b
            return c
        x= int (input("enter first number"))
        y= int (input("enter second number"))
        z= add (x,y)
        print("Addition=",z)

        enter first number10
        enter second number15
        Addition= 25
```

Explanation:

- in this program we declare a function for addition, and we take input from user
- in a given program in “x” store the value of “a” and “y” store the value of “b” and it will add and return c and store in “z” and at last it will display the z value.