**10th June Assignment**

1. In Python, what is the difference between a built-in function and a user-defined function? Provide an example of each.

**Ans: -** A built in function is a function that is pre-defined in a programming language or software application. These functions are usually a part of the core functionality of the language or application and are available for use without the need to define them. Some examples of built in functions in python are len ( ) :- Which Shows the length of the object, sum ( ): - it means its calculates the sum of numbers in program.

A user defined function is a function that is defined by the user in their code. These functions are created to perform specific tasks and can be called upon whenever the need arises in the code. User defined functions can also be referred to as custom functions or homemade functions. Some examples of user defined functions in python are greet ( ) function takes a name as an argument and prints a greeting message, square( ) function takes a number as an argument and returns the square of that number.

2. How can you pass arguments to a function in Python? Explain the difference between positional arguments and keyword arguments.

**Ans: -** In Python information can be passed into functions as an argument. An Arguments are specified after the function name, inside the parentheses & can be add as many arguments as you want, just separate them with a comma. There are two types of arguments positional arguments & keyword arguments.

**Positional arguments: -**

Positional arguments are arguments that are passed to a function based on their position or order. The order in which the arguments are passed is important, as it determines which parameter each argument corresponds to. The number of positional arguments passed must match the number of parameters defined in the function. Here’s an example of positional argument:

def greet (name, standard):

print(f” Hello {name}, you are in { standard } Standard.”)

greet( “ Akshay”,” 10th”)

In the above example, the function greet takes two arguments name & standard. When calling the function, the arguments are passed in the same order as Shown in example. The output will be: **“ Hello Akshay, you are in 10th standard.”**

**Keyword arguments:-**

Keyword arguments are arguments that are passed to a function using the parameter name as keyword. The order in which the arguments are passed is not important, as long as the parameter names are specified. Keyword arguments allow you to specify only the arguments you want to pass, skipping the ones you don’t need. Here’s an example of keyword arguments:

def greet (name, standard):

print(f” Hello {name}, you are in { standard } Standard.”)

greet( standard=”10th “, name=“ Akshay”)

In the above example, the function greet takes two arguments name & standard. When calling the function, the argument are passed using the parameter names. The output will be:

“**Hello Akshay, you are in 10th standard.”**

3. What is the purpose of the return statement in a function? Can a function have multiple return statements? Explain with an example.

**Ans: -** A return statement is used to end the execution of the function call and “returns” the result (value of the expression following the return keyword) to the caller. The statements after the return statements are not executed. If the return statement is without any expression, then the special value None is returned.

Yes, a function has multiple return statements.

For example:

def test\_list ():

return ['abc', 100]

result = test\_list ()

print(result)

print(type(result))

4. What are lambda functions in Python? How are they different from regular functions? Provide an example where a lambda function can be useful.

**Ans: -** Lambda functions are similar to user-defined functions but without a name. They’re commonly referred to as anonymous functions. Lambda functions are efficient whenever you want to create a function that will only contain simple expressions- that is, expressions that are usually a single line of a statement.

A lambda function is an anonymous function (i.e., defined without a name) that can take any number of arguments but, unlike normal functions, evaluates and returns only one expression. Note that, unlike a normal function, we don’t surround the parameters of a lambda function with parentheses.

For an example:

sum= lambda x, y, z : x + y + z

print(sum(10,15,20))

It shows the result: 45

5. How does the concept of "scope" apply to functions in Python? Explain the difference between local scope and global scope.

**Ans: -** Variables defined within a function have a restricted scope such that they do not exist outside of that function. Most other contexts for defining variables in python produce variables with file scope [i.e., they can be accessed anywhere in the file’s code, subsequent to their definition].

Global variables are useful for values that are relatively constant, or that many functions in the script must access, such as a session id.

However, a local variable has a limited scope, it exists only within the block that it is declared in. once that block ends, the variable is destroyed and its values lost.

6. How can you use the "return" statement in a Python function to return multiple values?

**Ans: -** In Python, you can return multiple values by simply separating them with commas in the return statement.

7. What is the difference between the "pass by value" and "pass by reference" concepts when it comes to function arguments in Python?

**Ans: -** In Python, the main difference between pass by value and pass by references is that, in a pass by value, the parameter value copies to another variable while, in a pass by reference, the actual parameter passes to the function.

In pass by value, the value of a function parameter is copied to another location of the memory. When accessing or modifying the variable within the function, it accesses only the copy. Thus, there is no effect on the original value.

In pass by reference, the memory address is passed to that function. In other words, the function gets access to the actual variable.

8. Create a function that can intake integer or decimal value and do following operations:

a. Logarithmic function (log x)

Ans: -

import math

# returning the log of 2,3

print ("The value of log 2 with base 3 is : ", end="")

print (math.log(2,3))

Result: The value of log 2 with base 3 is : 0.6309297535714574

b. Exponential function (exp(x))

Ans: -

import math

# returning the exp of 4

print ("The e\*\*4 value is : ", end="")

print (math.exp(4))

Result: The e\*\*4 value is : 54.598150033144236

c. Power function with base 2 (2x)

Ans: -

import math

# returning the value of 3\*\*2

print ("The value of 3 to the power 2 is : ", end="")

print (math.pow(3,2))

Result: The value of 3 to the power 2 is : 9.0

d. Square root

Ans: -

import math

# returning the square root of 25

print ("The value of square root of 25 : ", end="")

print (math.sqrt(25))

Result: The value of square root of 25 : 5.0

9. Create a function that takes a full name as an argument and returns first name and last name.

**Ans: -**

def fullname(firstname, lastname):

    fullname= fisrtname+ " " + lastname

    print(fullname)

firstname=input("enter the first name:")

lastname=input("enter the last name:")

print(fullname)

Result: -

“enter the first name: **Akshay**

“enter the last name: **Kadam**

**Akshay Kadam**