1. Why are functions advantageous to have in your programs?

Solution –

Functions reduce our effort on coding repetitive tasks. There are in-built functions which we can directly use based on our use. And there are user defined functions where we can create our own functions to perform our defined tasks multiple times over the program.

1. When does the code in a function run: when it's specified or when it's called?

Solution –

The code in a function runs when it’s called.

1. What statement creates a function?

Solution –

‘def’ keyword then the function name followed by parenthesis.

def calculator():

print(“Hello”)

1. What is the difference between a function and a function call?

Solution –

A function is piece of code created to do some specific tasks and can be used multiple times to avoid writing the same code again and again.

A function call means calling/invoking the function to do the defined tasks.

**Function Definition:**

def minimum\_balance\_check():

min\_balance = 5000

balance\_inp = float(input(“Enter your account balance”)

if balance\_inp <min\_balance:

print(“Your account balance is less than the required minimum”)

else:

print(“Your account balance is”, balance\_inp)

**Function Call:**

minimum\_balance\_check():

1. How many global scopes are there in a Python program? How many local scopes?

Solution –

There is only one global Python scope per program execution.

Each function in Python introduces a new local scope. Hence, number of local scopes depend on the structure of the program, how many functions are nested functions are defined.

1. What happens to variables in a local scope when the function call returns?

Solution –

Variables in a local scope are limited within that function it is used. If we try to access that variable outside of the function it will throw an error.

Example –

def sum(a,b):

result = a+b

return result

print (result)

The local variable “result” cannot be accessed outside of the function. For accessing the value stored in result, we need to call the function and the return statement will give us the value stored in “result”.

1. What is the concept of a return value? Is it possible to have a return value in an expression?

Solution –

When we try to compute something inside a function and want to use that result in our program we can use return statement inside a function, when we call that function, we can fetch the result and store/use it. For example –

def area\_circle(r):

result = 3.14\*r\*r

return result

print(area\_circle(4))

Python always returns a value if return statement is not explicitly provided, it implicitly returns a default value.

Yes, in Python we can use function call as a part of an expression and the returned value can be used directly in that expression.

For example –

def area\_circle(r):

result = 3.14\*r\*r

return result

height = 5

area\_cylinder = area\_circle(2)\*height

1. If a function does not have a return statement, what is the return value of a call to that function?

Solution –

None (Null)

1. How do you make a function variable refer to the global variable?

Solution –

By using a keyword “global”

1. What is the data type of None?

Solution –

None is a datatype of its own nontype

1. What does the sentence import areallyourpetsnamederic do?

Solution –

If there is a module present with that name, it will import that module else it will throw an error.

1. If you had a bacon() feature in a spam module, what would you call it after importing spam?

Solution –

spam.bacon()

1. What can you do to save a programme from crashing if it encounters an error?

Solution –

Error Handling can be used

1. What is the purpose of the try clause? What is the purpose of the except clause?

Solution –

The try clause tests the code for any error, the except clause lets us handle the error.