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

**ANSWER-**

*Functions reduces the need for duplicate code. This makes programs shorter, easier to read, easier to update.*

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

**ANSWER-**

*The code in function executes when it called not when the function is specified.*

***Example-***

*def greet():*

*print('Hello World!')*

*# call the function*

*greet()*

1. **What statement creates a function?**

**ANSWER-**

*In python, If you define a function with* ***def*** *keyword, then write the function identifier(name) followed by parentheses and a colon.*

***Syntax****- def function\_name (argument):*

*Function\_body*

*Return(optional)*

***Example****- def scam():*

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

**ANSWER-**

*A function is piece of code which enhanced the reusability and modularity of your program means that function need not be written again.*

*A function call means calling that function.*

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

**ANSWER-**

*We can declare variables in three different scope Global scope, Local scope and Non local scope.*

1. *Global scope- When we declared a variable outside the function theses variable is called a global scope. This means global variable can be accessed inside or outside of the function. It also be accessible from any scope of the program.*

***Example****-*

*# gloabl variable*

*TYPEmemeassage= 'adhjadak'*

*def greet():*

*# local variable*

*message = 'Hello'*

*print('Local', message)*

*print(TYPEmemeassage)*

*greet()*

*print(TYPEmemeassage)*

*Here, message is a local variable because it declare within the function and TYPEmemeassage is a global variable it declare outside of the function it can call within a function also and outside of the function also.*

*2) Local scope- When we declare a variable inside the function these variables have local scope (within the function). We cannot use them outside the function.*

***Example****-*

*Def add\_number():*

*Sum= 5+9*

*Here, sum is local variable so it can only be accessed within function.*

*3)Nonlocal Scope- means these variables are neither local scope nor global scope.*

*Nonlocal varibles are used in nested function whose local scope is not defined.*

*We use* ***nonlocal*** *keyword to create nonlocal variable.*

***Example-***

*# outside function*

*def outer():*

*message = 'local'*

*# nested function*

*def inner():*

*# declare nonlocal variable*

*nonlocal message*

*message = 'nonlocal'*

*print("inner:", message)*

*inner()*

*print("outer:", message)*

*outer()*

*Here, there is nested inner() function. We have used the nonlocal keyword to create a nonlocal variable.*

*The inner() function is defined in the scope of another function outer() function.*

***NOTE-*** *If we change the value of a nonlocal variable the changes appear in the local variable.*

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

**ANSWER-**

*A function’s local variable exists only while the function is executing. When the function begins, its local variables and its parameter variables are created in memory, and when the function ends, the local variables and parameter variables are destroyed.*

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

**ANSWER-**

*A python function always have a return value.*

*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 variables types:*

*Handle, Integer, Object or String.*

*The type of value your function returns depends largely on the task it performs.*

***Example****-*

*def add(x, y):*

*result = x + y*

*return result*

*output = add(5, 4)*

*print(f'Output of add(5, 4) function is {output}')*

*Here, return will return local variable value.*

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

**ANSWER-**

*Every function in python returns something.*

*If the function doesn’t have any return statement, then it returns* ***None.***

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

**ANSWER-**

*Normally, when you create a variable inside the function, that variable is local and can only be used inside that function.*

*To create a global variable inside the function we have to use* ***Global*** *keyword.*

*We can only access the global variable but cannot modify it from inside the function.*

***Example-***

*# global variable*

*c = 1*

*def add():*

*# use of global keyword*

*global c*

*# increment c by 2*

*c = c + 2*

*print(c)*

*add()*

*Here, we have defined c as global keyword inside the add() function*

1. **What is the data type of None?**

**ANSWER-**

*None is used to define a null value or null object in python.*

*It is not the same as empty string, a false or zero.*

*It is a datatype of the class NoneType object.*

*None is also used as the return type for function that don’t return a value.*

1. **What does the sentence import areallyourpetsnamederic do?**

**ANSWER-**

*That import statement will Import module name areallyourpetsnamederic.*

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

**ANSWER-**

*This function can be called with spam.bacon().*

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

**ANSWER-**

*Place the line code that might cause an error in a try clause.*

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

**ANSWER-**

*Try clause lets you test a block of code of errors.*

*Except clause let you handle the errors.*

***Example****-*

*try:*

*1 / 0*

*except ZeroDivisionError:*

*print('Divided by zero')*

*print('Should reach here')*

*Here, the above statement will print*

*'Divided by zero'*

*'Should reach here' As output*

*But if divide 1/ anything instead of 0 it will only print ‘should reach here’ as output.*