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

**Ans**:

Use of Function is advantageous because it helps to reuse the code without writing the code again and again. Duplication of code is removed; the written program is simpler without repeated code.

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

**Ans:**

The function runs when it is called.

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

**Ans:**

The def keyword is used to define a function.

**For example:** def function\_name():

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

**Ans:**

A function is a set of code that is defined to do a particular task. For example len(), print(), append(), etc

Whereas a function call is, when function is called and is required to be used in a code.

For example:

x=7

print(x) # this is an example of function call where print() function is called to display value of variable x.

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

**Ans**: there are no subtypes for global and local scopes, but there are four types of scope in python namely; local, non- local, global and built-in scope.

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

**Ans:** The value stored in a variable in local scope exists till the execution of function and after the function call returns, the value stored in a variable in local scope ceases to exists.

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

**Ans:**

Return is a keyword available for functions only and cannot be used outside a function. The return keyword is used to end the execution of function and return the value to the function caller.

In a function, return statement returns the value of variable with its data type or expression value it is returning. If return statement is without an expression or a variable then function call returns None type.

|  |  |
| --- | --- |
| **For example:**  def test():  a=1  b=2  return a\*b  the return of above function call will return a value with int data type. | Whereas  def test():  a=1  b=2  return  the return of above function call will give a **None data type**. |

The return cannot be used inside an expression but it can return the value of expression.

For example:

def test():

a=1

b=2

return a\*b

here return keyword is used to return the value of expression a\*b.

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

**Ans: return value of a call to that function that does not have a return statement is None type**

**for example**

In a function return statement returns the data type of variable or expression value it is returning. If return statement is without an expression or a variable then function call returns None type.

|  |  |
| --- | --- |
| **For example:**  def test():  a=1  b=2  return a\*b  the return of above function call will return a value with int data type. | Whereas  def test():  a=1  b=2  return  the return of above function call will give a **None data type**. |

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

**Ans:** A function variable refer to the global variable by using a keyword **global**

|  |  |
| --- | --- |
| **For example:**  def test1():  x= "shubham"  print(x)  test1()  the output: ‘shubham’  print(x)  the output: **it gives error that variable x is not defined** | **Example of global keyword:**  def test1():  **global x**  x= "shubham"  print(x)  test1()  the output: ‘shubham’  print(x)  the output: ‘shubham’ |

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

**Ans:** The data type of **None** is **NoneType.**

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

**Ans:** It tries to import a module named areallyourpetsnamederic, but since there is no such standard module present, hence it gives error.

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

**Ans:** bacon() feature will be called a function in spam module after importing the module.

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

**Ans:** To save a programme from crashing if it encounters an error, we can use error handling techniques like exception handling using **try- except** and **try-except-finally.** Also **assertion** can be used but it is more of a fail first approach.

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

**Ans:**

**try – except** clause is used for exception handling. If we know a block of code may fail then we can use try – except clause to prevent crashing of a program.

**The purpose of try clause is to let user test a code for error and the purpose of except clause is to let the user handle that error.**

|  |  |
| --- | --- |
| For example:  try:  a= input()  b= 10\* a  except:  print("error has occurred") | Here in this example in try block, **a** variable will take input in str data type by default.  The expression b will give error and to handle this error except block will print ‘error has occurred’ instead of crashing the peogram. |