Assignment_3

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

Ans.

Python is an interpreted language. The interpreter goes line by line and catches any syntax error(s). The code in a function is also checked but it is not run. Code in the function only runs when the function is invoked, that is, when the function is called.

2. What statement causes a function to be created?

Ans.

In Python, a function is created by using the 'def' keyword. For example

```
def func():
  pass
```

3. What is the distinction between a function and a call to a function?

Ans.

A function can be said as a procedure to achieve a particular result, whereas a function call is using this function to achieve that task. For example consider the following function to print numbers 0 to 9

```
def func():    # this is function definition
  for number in range(10):
    print(number, end = " ")

func()  # this is function call
```

4. In a Python application, how many global scopes are there? How many local scopes are there?

Ans.

There is only one global Python scope per program execution. This scope remains in existence until the program terminates and all its names are forgotten.

There is only one local scope per Python program.

5. When a function call returns, what happens to variables in the local scope?

Ans.

In Python, the variables inside a function are local to it and are only in the memory when the function is called, that is, when the code inside the function is running. After the function call is over and the control returns to the calling statement, the local variables are gone forever. There memory inside the stack is freed.

Consider the following example for clarity:

```
# function definition
def func(n):
  for number in range(n):
    print(number, end = " ")

num = int(input("Enter number: "))
func(num)  # function call
print(n)   # this line throws error
```

The line print(n) throws error because the variable n is local to func() and it is erased from the memory as soon as control reaches to next line after function call.

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

Ans.

In Python, a return statement is a special statement that can be used inside a function to send the function’s result back to the caller. A return statement consists of the return keyword followed by an optional return value.

Yes it is possible to have return statement in an expression. For a simple example consider the code snippet below :

```
def increment(x):
  return x + 1    # this returns an expression, i.e., x + 1
```

7. What is the return value of a call to a function that does not have a return statement?

Ans.

If a function does not have a return statement, in Python, the return value of that function is 'None'.

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

Ans.

We can make a function variable to refer the global variable by using the 'global' keyword in Python. For example

```
var = 5

def func1():
# var = 10
# if we don't use global then var refers to the local variable of this function
  global var = 10

def func2():
  print(var) # this will print 10
```

9. What data form does None belong to?

Ans.

The None belongs to its own datatype which is None type. Only None is None type.

10. What does the sentence import areallyourpetsnamederic do?

Ans.

This statement will try to import the module 'areallyourpetsnamederic' in our program, if the module really exists. Otherwise it will throw error.

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

Ans.

After importing the spam module, the function can be called as spam.bacon()

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

Ans.

We can put the particular block of code that might cause an error in a 'try' clause.

The try and except block in Python is used to catch and handle exceptions. Python executes code under the try statement as a part of the program. The code that follows the except statement is the program’s response to any exceptions in the preceding try clause.

For example

```
def division(n):
  return n/0

num = int(input("Enter the number: "))
try:
  result = division(n)
except:
  print("Division by zero not allowed.")
```

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

Ans.

The try and except block in Python is used to catch and handle exceptions. Python executes code under the try statement as a part of the program. The code that follows the except statement is the program’s response to any exceptions in the preceding try clause.