Day – 4 Morning Assessment

Functions:

1. Function is a block of code which is used to reduce the reuse of same code for different parameters.
2. Function in python is defined using def keyword and the function name. i.e.,

def function\_name(parameters):

#block of code

1. By using the function call after declaring the function.
2. ‘return’ statement returns the result of the function to the function call.
3. Function to add two numbers:

def add(a,b):

return a+b

print(add(x,y))

1. Parameters are the values which are received by the function through function call and use them in the function. Whereas, arguments are the values which are given to the function call.
2. The function gets defined but never gets executed unless it is called.
3. Function to check if the number is even or odd:

def even\_odd(num):

if num%2 == 0: return “Even”

else: return “Odd”

print(even\_odd(n))

1. Default argument is a value which is taken default when there is no argument is passed.
2. Keyword arguments are the arguments which are in a key, value pair.
3. def greet(name = “user”):

print(f”Hello, {name}”)

1. Function to return the square of the number:

def square(num):

return num\*num

print(square(n))

1. Yes, function can return multiple values but there should be enough variables provided to catch the return values.

Eg: def operation(a,b):

return a+b,a-b,a\*b,a/b

Add,sub,mul,div = operation(a,b)

Print(add,sub,mul,div)

1. Print() prints the output, whereas return returns the output to the functioncall.
2. Global Variable is a variable which can be accessed anywhere throughout the program.
3. Local variable is a variable which can only be accessed in the respective function.
4. Function to print name and age:

def details(name, age):

print(name,age)

name,age = input(“Enter your name: ”),int(input(“Enter your age: “))

1. Pass is a keyword which is used to pass through the function when no block of code is there to declare.
2. Multiply = lambda x,y:x\*y

Print(x,y)

1. Def fact(n):

If n==0 or n==1: print(“1”)

Else: print(n\*(n-1))

Fact(n)

While Loop:

1. While loop is a looping function. It iterates until the given condition is satisfied or True.
2. num=1

while num<=5:

Print(num)

num += 1

1. Infinite while loop is a condition where while loop statement always be true and the loop runs infinite times.
2. We can stop infinite loop through break statement.
3. num = 1

while num<=10:

If num%2 == 0: print(num)

num += 1

1. In for loop we can estimate the number of iterations but in while loop we cannot estimate the number of iterations. In for loop we have to define the start and stop numbers where in while loop the function iterates until the condition is true.
2. Break statement is used to break the loop.
3. Continue statement is used to skip the remaining part of the function and goes to the next iteration.
4. Program to sum the numbers:

num = 1

sum = 0

while num<=100:

sum = sum+num

num += 1

1. While true:

Text = input(“Enter a string: “)

If text == “exit”:break

1. In while loop there is no separate place where we can initialize the value. So, if we want to initialize a variable we can do it before while loop. If we initialize variable after while loop then every time the loop runs, the variable get initialized.
2. Program for table of 5:

num = 5

n = 1

while n<=10:

print(f”{num} x {n} = {num\*n}”)

n= n+1

1. If the condition of the while loop is always true then it leads to infinite looping.
2. num = 10

while num>0:

print(num)

num = num-1

1. num = 1

while num <=20:

if num%2!=0: print(num)

num = num+1

1. Yes, we can use else with a while loop.

Eg: num = 1

while num <=10:

if num%2 == 0: print(“Even”)

else: print(“Odd”)

num = num+1

1. Num = 123

rev = 0

While num > 0:

rem = num%10

rev = (rev\*10)+rem

num = num//10

print(rev)

1. In while loop there is no specific place for updating the variable as in for loop so, we need to update the variable inside the while loop, if not the while loop condition will run based on the existing condition and this leads to infinite loop.
2. num = 4

fact = 1

while num>0:

fact\*= num

num = num-1

print(fact)

1. num = 626

x=num

rev =0

while num>0:

rem = num%10

rev = rev\*10+rem

num = num//10

if x == rev: print(“palindrome”)

else:(”not palindrome”)