Section A: Theory

1. Explain the difference between if, elif, and else statements in Python.

in python we having if ,eilf and else statements for performing conditional operations
if : it is conditional statement used to define conditions and the block code will be executed once the condition is true

eilf: in your code if you having multiple conditions then you can define those conditions in **elif** , it executes when previous condition is not true.

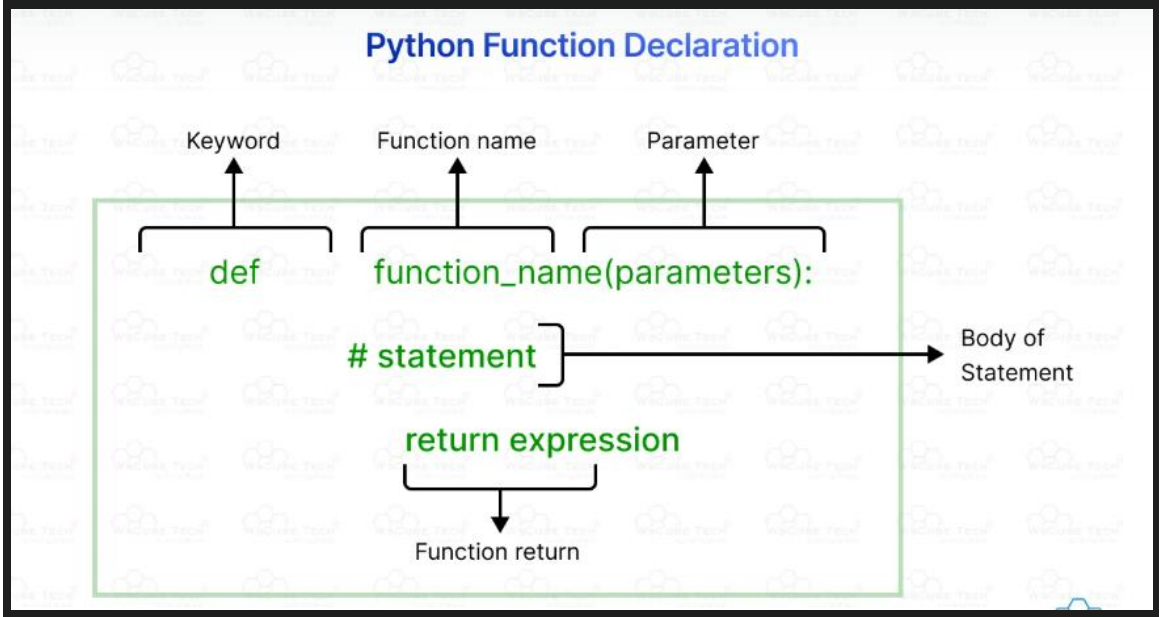
else: it will excuted when no any conditions are true , you can't use else without if

2. What is a nested loop? Provide an example of when you might use one.

When your data having more than one columns in that scenario you can use nested loops specially when your Working with multi-dimensional data you use nested loops
Real-world example:
Clock: The second hand repeats every 60 seconds and then the minute hand moves to the next minute.
Earth: Rotates itself as it also moves around the sun.

3. Define the term "function" in Python and explain its advantages.

Function - is Block of Code that designed to perform some task ,function can be reusable



Advantages:

- **Code Reusability :** we can call as many times as we want
- **Modularity :** we can separate into different modules that can be used by multiple developers
- **Code Readability: when tasks are divided into sections that will help for better code readability**
- **Maintainability: we can fix issues or update section wives no need to lookup whole code**

4. What is the purpose of the return statement in a function?

The Python return statement is used to exit from function and return to the function caller. It optionally returns the specified value or expression from the function to the caller. If there is no expression, it returns None.

Syntax

return [expressions]

```
def add_numbers(a, b):
    result = a + b
    return result
sum_result = add_numbers(10, 5)
print("The sum is:", sum_result)
output:15

def add()
    pass
print(add())
output: None
```

5. Explain the use of lambda functions in Python. How are they different from regular functions?

Anonymous function in Python is defined without a name. We define function in Python using the def keyword, but anonymous functions are defined using the lambda keyword. Therefore, they are also known as lambda functions. These functions can have zero or more arguments but only one return value.

Mostly lambda are usfull when your are working with HOF's

Syntax: `lambda arguments: expression`

print((lambda x:x**2)(5)) →25

list(map(lambda x:x**5,[1,2,3]))→[1,4,25]

6. What is a Higher-Order Function (HOF)? Provide two examples.

Function if it contains other functions as a parameter or returns a function as an output That functions called as HOF's

Passing Function as Argument to another Function

```
def greeting(name):
   print("Hello",name)

def fun(greet):
   greet('lavanya')
fun(greeting)
```

output:Hello lavanya

Returning Function from another function

```
def add():
   def each_add_10(y):
      print(10+y)
   return each_add_10
fun=add()
fun(20)
```

7. Describe the difference between while loops and for loops.

for loops are used when the number of iterations is known beforehand, such as when iterating over elements in a list or any iterable object.

```
numbers = [1, 2, 3, 4, 5]

print("Iterating over the list using a for loop:")
for num in numbers:
    print(num)
```

While loops are used when the number of iterations is not known beforehand, and the loop continues as long as a certain condition is true. The loop variable is typically initialized before the loop, and the condition is checked before each iteration.

```
n='y'
while(n!='n'):

    n=input("enter y if want to continue n if not: ")
    print('welcome')
print("Bye")
```

output:

enter y if want to continue n if not: y

welcome

enter y if want to continue n if not: y

welcome enter y if want to continue n if not: n welcome Bye

8. Explain the purpose of the break and continue statements in loops.

Break: The break statement in Python is used to terminate the loop

Continue: continue statement is opposite to that of the break statement, instead of terminating the loop, it forces to execute the next iteration of the loop.
Current iteration is skipped and next itration is starts

9. What are default arguments in functions? Provide an example.

Python : allows function arguments to have default values. If the function is called without the argument, the argument gets its default value.
Rules:

1. In the case of passing the keyword arguments, the order of arguments is not important.
2. There should be only one value for one parameter.
3. The passed keyword name should match with the actual keyword name.
4. In the case of calling a function containing non-keyword arguments, the order is important.

```
def student(firstname, lastname ='Mark', standard ='Fifth'):
    print(firstname, lastname, 'studies in', standard, 'Standard')

# 1 positional argument
student('John')

# 3 positional arguments
student('John', 'Gates', 'Seventh')

# 2 positional arguments
student('John', 'Gates')
student('John', 'Seventh')
```

Output: John Seventh Studies in Fifth Standard

```
def student(firstname, lastname ='Mark', standard ='Fifth'):
    print(firstname, lastname, 'studies in', standard, 'Standard')

# 1 keyword argument
student(firstname ='John')

# 2 keyword arguments
student(firstname ='John', standard ='Seventh')

# 2 keyword arguments
student(lastname ='Gates', firstname ='John')
```

10. How can a function return multiple values in Python?

1)Using list :
def add(x,y)
Return [x+y,y-x]
Print(add(1,2))→[3,2]

2)Using Comma Separated value/tuple

```
def add(x,y):
 return x,y
print(add(1,2))→ (1,2)
def add(x,y):
  return (x,y)
print(add(1,2))→ (1,2)
```

Section B: Correct the Error

```
# 1.
# if x=10:
#       print("x is 10")
# ANS:
# x=10
# if x==10:
#     print(x)
# output:
# 10


# 2.
#    for i in range(10)
#       print(i)
# ANS:
# for i in range(10):
#     print(i)
#
# 3.
#    def greet(name)
#       return "Hello" + name
# def greet(name):
#     return "Hello"+ name
# print(greet("lavanya"))
# output:Hellolavanya


# 4.
# numbers = [1, 2, 3, 4, 5]
# print(numbers[5])
 # output:
# print(numbers[4])
# list index out of range
```

```
# 5.
# lambda x, y: x + y
# print(lambda(5, 10))

# ANS:
# output:
# invalid Syntax
# print((lambda x, y: x + y)(5,10))
# 15


# 6.
#    def add(a, b):
#        print(a + b
# ANS:
# sysntax Error
# def add(a,b):
#     print(a+b)
# add(10,20)
# output:
# 30


# 7.
# while True:
#        print("Looping...")
# continue
# ANS: syntax Error continue not properly in loop


# 8.
# hof = map(lambda x: x * 2, [1, 2, 3, 4])
# print(hof[0])

# ANS:
# map oject not subscriptable
# hof = list(map(lambda x: x * 2, [1, 2, 3, 4]))
# print(hof[0])

# 9.
# def multiply(x, y = 2):
#        return x * y
# print(multiply())
```

```
# ANS:
# TypeError: missing 1 argument
# def multiply(x, y = 2):
#       return x * y
# print(multiply(10))


# 10.
# result = lambda x: x  2
# print(result 5)
# ANS
# result = lambda x: x * 2
# print(result(5))
```

Section C: Write a Program

```
#1 Write a program to find the largest of three numbers using conditional statements.
# n1=int(input("Enter no1: "))
# n2=int(input("Enter no2: "))
# n3=int(input("Enter no3: "))
# if n1>n2 and n1>n3:
#     print(n1,"is greater")
# elif n2>n3 and n2>n1:
#     print(n2,'greater')
# else:
#     print(n3,'greater')
# output:
# Enter no1: 10
# Enter no2: 20
# Enter no3: 30
# 30 greater


#2 Write a program to print all prime numbers between 1 and 50 using a loop.
# list=[]
# for i in range(2,51):
#     f=1
#     for j in range(2,i):
#         if i%j==0:
#             f=0
#     if f==1:
```

```
#         list.append(i)
# print(list)
# output:
# [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47]


#3 Create a function that takes a list of numbers and returns the sum of its elements.
# l=[10,20,30,40,1,2,3,4]
# sum=0
# for i in l:
#     sum+=i
# print("Sum=",sum)
# output: 110


#4 Write a lambda function to calculate the square of a given number.
# x=int(input("Enter no "))
# print((lambda x: x**2)(x))
# output:
# 100


#5 Implement a program to check whether a given number is a palindrome using a loop.
# n=int(input("Enter no "))
# r=0
# n1=n
# while(n!=0):
#     d=n%10
#     r=r*10+d
#     n=n//10
# print(r)
# if n1==r:
#     print("Palindrom")
# else:
#     print("Not palindrom")
# output:
# Enter no 1221
# 1221
# Palindrom
```

```
#6 Write a function that takes two arguments and returns the greater of the two.
# n1=int(input("Enter no1 "))
# n2=int(input('Enter no2 '))
# if n1>n2:
#     print(n1,'Greater')
# else:
#     print(n2,'greater')
# output:
# Enter no1 12
# Enter no2 23
# 23 greater


#7 Create a Higher-Order Function that applies a given function to each element in a list.
# list=[1,2,3,4,5]
# def func(list):
#     return sum(list)
# def HOF(fun,list):
#     return fun(list)
# print(HOF(func,list))
# output:
# 15


#8 Write a program to calculate the factorial of a number using recursion.
# def fact(n):
#     if n==1 or n==0:
#         return 1
#     else:
#         return n * fact(n-1)
# print(fact(5))
# output:
# 120


# #9 Create a function that filters out odd numbers from a list using filter and a lambda function.
# list=[1,20,3,40,5,7]
# list=filter(lambda x:x%2!=0,list)
# print(tuple(list))
# output:
# (1, 3, 5, 7)
```

#10 Write a program to count the number of vowels in a given string using loops and conditionals.

```
# v='aioueAIOUE'
# s=input("Enter string ")
# c=0
# for i in s:
#    if i in v:
#       c+=1
# print(c)
# output:
# Enter string lavanya
# 3
```