Task 1:

1. Describe the difference between mutable and immutable types in Python. Provide examples.

Ans:-

- There are two different types of objects in python namely mutable and immutable. Based on the type the method of handling is decided.
- First type is mutable objects. This are objects that are allowed to change the values of it i.e. update on content is allowed for example list is mutable which means we can add, update, delete elements of list.
- Mutable objects are easy to handle when there is change in its size because of updation or deletion in object. Following data types falls under this type list, dictionary, set etc

  E.g lst = ['hello',1,23,4]

  lst[1] = 'world' #will not give error and change the second element to 'world'

- Second type is immutable objects. This are the types of objects that do not allow change in its value and size. Once created we cannot update the object example of such type is tuple. If we try to add element in tuple after declaration we will get error. Immutable objects are easy to create and expensive to change. Following data types falls under this type tuple, string.

  E.g. tpl = (1,"Kalp",334)

  tpl[0] = 10 # this will give error

- The use of the mutable type is recommanded when there is possibility in change in size.

2. Write a Python program to check if a number is a palindrome.

Ans:-

```python
def reverseNum1(num):
    return num[::-1]

def reverseNum2(num):
    return ''.join(reversed(num))

def reverseNUm3(num):
    num = int(num)
    temp = 0

    while num > 0:
        temp = temp * 10 + num % 10
        num = num // 10

    return str(num)


num = input("Enter the number to be checked:")

if num == reverseNum2(num):
    print("Given number is palindrome!")
else:
    print("Given number is not palindrome!")
```

3. What are Python decorators? Provide an example of how to use one.

Ans:-

- Decorators in python is a strong tool that can help us to modify the working of the function without modifying the function's code.
- They are higher order functions that takes other functions as arguments. They are applied to functions or methods using the @decorator_name syntax, which is syntactic methode for wrapping a function with a decorator.
- Inside decorator there is often another function called wrapper is defined which wrappes the behaviour of original function. Then decorator will return the wrapper function which is executed in place of original function.
- E.g.

```
def my_decorator(func):
def wrapper():
print("Before the function is called.")
func()
print("After the function is called.")
return wrapper

@my_decorator
def say_hello():
print("Hello, world!")

say_hello()
```

4. Explain the purpose of __init__ in Python classes.

Ans:-

- The __init__ method is an special methode that is called when object of any class is created. It works as an constructor in python whose work is to initialize the object of the class.
- It is a special methode so it can have an return statement but must return None else will give error because it is not used to do any ordinary work but only initialization of object. It can also take arguments if defined and if there are arguments when defined and we fail to provide those arguments it will give TypeError.
- We cannot call this methode manually it is called automatically when object is created.

5. What is the difference between a list and a tuple? When would you use each?

Ans:-

- List is an ordered collection of elements in python that is mutable. It is mutable which means it can be updated after the declaration. Iteration over list is time consuming. They consumer more memory and are suitable for operation such as insertion and deletion. Lists are generally used when the amount of elements to store are not known and also where frequent updation of data in list is imminent. List is declared using '[]' square brackets.

- Tuple is an ordered collection of elements in python that is immutable. It is immutable which means is cannot be updated after declaration. Iteration over tuple is fast as compared with lists. They consume less memory and suitable where once written it needed to be read frequently. Tuples are used when data needed to be stored inside is already known and updation on it is not necessary. Tuples are declared using '()' round brackets.