Assignment 2

Name: Sami Imran

Roll Number: BASIM-S24-033

Sir Name: Rasikh Ali

Subject: Object Oriented Programming

Lab 2 – Task

Task 1; Class Concepts:

1. Class vs. Object:

a. Explain the difference between a class and an object in Python.

b. Provide an example

Class:

- A class is a blueprint for creating objects. It defines the structure and behavior that the objects created from the class will have.
- A class itself does not store any data but specifies what data and behaviors its objects will have.

Object:

- An object is an instance of a class. It is created based on the class definition and it holds actual data and methods
- An object is like a specific entity of the class that all the properties define in the class

Example:

```
class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age

    def __str__(self):
        return f"(Name: {self.name}, Age: {self.age})"
person1 = Person("Ali", 30)

print(person1)



class Car:
    def __init__(self, make, model, year):
        self.make = make
        self.model = model
        self.year = year
    def display_info(self):
        print(f"Car: {self.year} {self.make} {self.model}")
mycar = Car("BMW", "m4", 2021)
mycar.display_info()
```

2. Constructor Method (__init__) vs __str__() Function:

a. Explain the difference between them in Python.

b. Provide an example.

__init__:

- The __init__ method is a special method in Python that is automatically called when an object of a class is created.
- It is used to initialize the object's attributes with the values provided when the object is instantiated.

__str__:

- The __str__ method is a special method that is called when an object needs to be represented as a string such as when printing the object.
- It is meant to return a human readable or informal string representation of the object making it easy to understand.

Example:

```
class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age

    def __str__(self):
        return f"Person(Name: {self.name}, Age: {self.age})"
# Creating an object of the Person class
person1 = Person("Ali", 30)
print(person1)
```