**MINAHIL QADIR (SU92BSAIM\_S24\_047)**

TASK NO 2 **OOP**

**Difference between Class and Object:**

**Class:**

class is a blueprint or template for creating object. It tells which properties and action the object will have.

**EXAMPLE:**

class Student:

def \_\_init\_\_(self,name):

self.name=name

def attendance(self):

print(f"the student with Name{self.name}is present")

in this code ‘Student’ is the class

Object:

An object is a instance of a class .it contain the data of its own. It represent the behavior describe by the class.

**EXAMPLE:**

mystd=Student('minahil')

mystd.attendance()

in this code ‘mystd’ is the object of ‘Student’ class

**EXAMPLE:**

Let’s suppose our university building is a class because it represent the object inside it so the objects are our Classrooms and departments and all the things that are present inside the building simply the object shows the properties inside the class and our class will tell us how our object will react…

**Constructor Method (\_\_init\_\_) vs \_\_str\_\_() Function:**

class Person:

def \_\_init\_\_(self, name, age):

self.name = name

self.age = age

person = Person("Minahil", 18)

print(person.name)

**Minahil**

\_\_init\_\_ initializes name and age attributes when a Person object is created.

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})"

person = Person("Minahil", 18)

print(person)

**Person(name=Minahil, age=18)**

\_\_str\_\_ provides a readable string representation of the Person object

**EXAMPLE :**

When you get a new book, you need to put its title and author on the cover. The \_\_init\_\_ method is like the process of putting this information on the book.

**EXAMPLE :**

When you want to tell someone about the book, you might say, "It's a book titled 'Harry Potter' by J.K. " The \_\_str\_\_ method is like that description.

THANKYOU😊