

|  |  |
| --- | --- |
| Name | Ali Mustafa Shah |
| Roll Number | SU92-BSDSM-S24-005 |
| Section | 2A |
| Semester | 2nd |
| Subject | Opp lab |
| Task | 2 |
| Submitted by | Syed Ali Mustafa Shah |
| Submitted to | Sir Rasikh |

**Task 2**

**Class Concepts:**

1. Class vs. Object:

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

b. Provide an example.

2. Constructor Method (\_\_init\_\_) vs \_\_str\_\_() Function:

a. Explain the difference between them in Python.

b. Provide an example

**Solution**

**Class vs. Object:**

**a. Difference:**

* **Class:** A class is a blueprint or template for creating objects. It defines the attributes (data members) and methods (functions) that objects of that class will have. It's like a cookie cutter that defines the shape of cookies.
* **Object:** An object is an instance of a class. It's a concrete realization of the class, containing specific values for the attributes defined in the class. It's like an actual cookie created using the cookie cutter.

**Example**

class Vehicle:

    def \_\_init\_\_(self, make):

*self*.make = make

class Car(Vehicle):

    def \_\_init\_\_(self, model, make, no\_of\_doors, engine):

*self*.model = model

*self*.make = make

*self*.no\_of\_doors = no\_of\_doors

*self*.engine = engine

        super().\_\_init\_\_(make)

    def printdetails(self):

        print(f"Make: {*self*.make}\nModel:{*self*.model}\nNo of doors: {*self*.no\_of\_doors}\nEngien: {*self*.engien}")

myobj = Car("HONDA", "Civic", 4, "V8")

myobj.printdetails()

**2. Constructor Method (init) vs str() Function:**

**a. Difference:**

* **init()**: This is a special method called automatically when an object of a class is created. It's used to initialize the attributes of the object with initial values. It's like giving the object a starting point.
* **str()**: This is a special method that returns a string representation of an object. When you try to print an object or convert it to a string, Python calls this method. It's useful for debugging or displaying information about the object in a human-readable format.

**Example**

class gender:

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

*self*.name = name

*self*.age = age

    def \_\_str\_\_(self):

        return f"Name: {*self*.name}, Age: {*self*.age}"

gender1 = gender("Alice", 30)

print(gender1)