DAY-3:

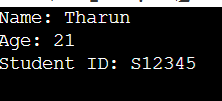
OOPS:  
PROGRAM:

class Person:  
    def \_\_init\_\_(self, name, age):  
        self.name = name  
        self.age = age

    def display\_info(self):  
        print(f"Name: {self.name}")  
        print(f"Age: {self.age}")  
class Student(Person):  
    def \_\_init\_\_(self, name, age, student\_id):  
        # Call the constructor of the parent class  
        super().\_\_init\_\_(name, age)  
        self.student\_id = student\_id

    def display\_student\_info(self):  
        # Call parent method  
        self.display\_info()  
        print(f"Student ID: {self.student\_id}")

OUTPUT:



PROGRAM:

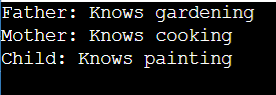
class Father:  
    def skill(self):  
        print("Father: Knows gardening")

class Mother:  
    def talent(self):  
        print("Mother: Knows cooking")

class Child(Father, Mother):  # Multiple inheritance  
    def own\_skill(self):  
        print("Child: Knows painting")

c = Child()  
c.skill()  
c.talent()  
c.own\_skill()

OUTPUT:



PROGRAM:  
class Animal:  
    def speak(self):  
        print("Animal speaks")

class Dog(Animal):  # Single inheritance  
    def bark(self):  
        print("Dog barks")

dog = Dog()  
dog.speak()  
dog.bark()

OUTPUT:



PROGRAM:

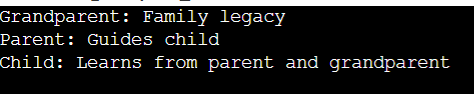
class Grandparent:  
    def legacy(self):  
        print("Grandparent: Family legacy")

class Parent(Grandparent):  
    def guide(self):  
        print("Parent: Guides child")

class Child(Parent):  # Multilevel inheritance  
    def learn(self):  
        print("Child: Learns from parent and grandparent")

c = Child()  
c.legacy()  
c.guide()  
c.learn()

OUTPUT:



PROGRAM:

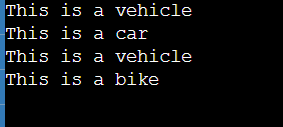
class Vehicle:  
    def category(self):  
        print("This is a vehicle")

class Car(Vehicle):  
    def type(self):  
        print("This is a car")

class Bike(Vehicle):  
    def type(self):  
        print("This is a bike")

car = Car()  
bike = Bike()

car.category()  
car.type()  
bike.category()  
bike.type()

OUTPUT:  


PROGRAM:

class A:  
    def show(self):  
        print("Class A")

class B(A):  
    def show(self):  
        print("Class B")

class C(A):  
    def show(self):  
        print("Class C")

class D(B, C):  # Hybrid Inheritance  
    pass

d = D()  
d.show()  # Output: Class B (due to MRO)

OUTPUT:

