ROMIZ
ID: IT22045

Date
S S M T W T F

Abstraction, Interface, Inheritance

```
import java.io.PrintStream;
interface Engine {
        void startEngine();
        void stopEngine();
    }   PrintStream o = System.out;

abstract class Vehicle {
        String brand;
        int year;
    public Vehicle(String brand, int year){
        this.brand = brand;
        this.year = year;
    }
    abstract void drive();
    public void displayInfo(){
        sys Engine.o.println("Band: "+ Brand
                +", Year: "+year);
    }
}
class Car extends Vehicle implement Engine{
    public Car(String brand, int year){
        super(brand, year);
    }
```

Date :
S S M T W T F

```
@Override
public void stopEngine(){
    o.println("Car Engine Stop.");

@Override
void drive(){
    o.println("Driving a car.");

}}

class Bike extends Vehicle implement Engine{
    public Bike(String Brand, int year){
        super(brand, year);

@Override
public void startEngine(){
    o.println("Bike engine Start.");
}

@Override
public void stopEngine(){
    o.println("Bike Enginestop.");
}

public class TestVehicle{
    public static void main(string[] args){
    Vehicle car = new Car("Toyota", 2022);
    car.displayInfo();
    ((Engine) car).stopEngine();
    car.drive();
    ((Engine) car).stopEngine();
```

Date
S S M T W T F

```
Engine.O.println();
Vehicle bike = new Bike("Yamaha", 2023);
bike.displayInfo();
((Engine) bike).startEngine();
bike.drive();
((Engine) bike).stopEngine();
}
}
```