IT22005 09/05/25

```
interface Drivable{
    void drive(); // just declared not implemented.
}
abstract class vehicle{
    String brand;
    public vehicle(String brand){
        this.brand = brand;
    }
    abstract void fueltype(); //abstract method
    public void showBrand(){
        System.out.println("Vehicle Brand: " + brand);
    }
}
```

• interface has only abstract method.

• But abstract class can have both abstract and concrete method.

```
class Car extends vehicle implements Drivable{
    public car(String brand){
        super(brand); // call constructor of base class.
    }
    @override public void drive(){
        System.out.println("Car is driving");
    }
    @Override void fueltype(){
        System.out.println("Car uses petrol");
```

```
  }

 }

class Bike extends Vehicle implements Drivable{
    public Bike (String brand){
        super (brand);
    }
 @Override public void drive(){
       System.out.println(" Bike is riding. ");
   }
}

public class AbstractionExample{
public static void main (String [] args){
  Car car1 = new Car (" Toyota ");
  car1 . showBrand ();
  car1 . drive ();
  car1 . fuelType();

 Bike b1 = new Bike("Yamaha");
  b1 . showBrand();
  b1 . drive ();
//b1 . fuel
}
}
```