IT22049

```
interface Operatable{
    void operate(); }
abstract class Appliance{
    String model;
    public Appliance (String model){
        this. model = model; }
    abstract void powerSource ();
    public void showModel () {
        System.out.println ("Appliance Model:"+mod
    }
}
class WashingMachine extends Appliance impements
                                        Operatable {
    public WashineMachine (String model){
        super (model); }
    public void operate () {
        System.out.println ("washing machine is
                running a wash cycle.");}
    void PowerSource () {
        System.out.println ("washing machine uses
                electricity");}
}
```

```
class SolarFan extends Appliance implements
                              Operatable {
    public SolarFan(String model){
        super(model);}
    public void operate(){
        System.out.println("Solar fan is spinning
              with solar energy.");
     void powerSource(){
        system.out.println("Solar fan uses
              solar power.");}
}

public class Abstraction {
    public static void main(String[] args){
        Appliance wm = new Washingmachine("LG-
                                    Turbowash);
        wm.showModel();
        ((Operatable)wm).operate();
        wm.powerSource();
        System.out.println();
        Appliance sf = new SolarFan("SunWind-pro);
        sf.showModel();
        ((operatable)sf).operate():
        sf.powerSource();
    }
}
```