**LAB ASSIGNMENT 12**

**DECORATOR DESIGN PATTERN**

CODE:

* Instrument.java

package decoratorDP;

public interface Instrument {

*void* play();

}

* Guitar.java

package decoratorDP;

public class Guitar implements *Instrument* {

   @*Override*

   public *void* play() {

      System.out.println("Instrument is Guitar");

   }

}

* Drum.java

package decoratorDP;

public class Drum implements *Instrument* {

   @*Override*

   public *void* play() {

      System.out.println("Instrument is Drum");

   }

}

* InstrumentDecorator.java

package decoratorDP;

public abstract class InstrumentDecorator implements *Instrument* {

   protected *Instrument* decoratedInstrument;

   public InstrumentDecorator(*Instrument* *decoratedShape*){

      this.decoratedInstrument = decoratedShape;

   }

   public *void* play(){

      decoratedInstrument.play();

   }

}

* EnhancedInstrumentDecorator.java //main file

package decoratorDP;

public class EnhancedInstrumentDecorator extends *InstrumentDecorator* {

   public EnhancedInstrumentDecorator(*Instrument* *decoratedInstrument*) {

      super(decoratedInstrument);

   }

   @*Override*

   public *void* play() {

      decoratedInstrument.play();

      setElectricBoard(decoratedInstrument);

   }

   private *void* setElectricBoard(*Instrument* *decoratedInstrument*){

      System.out.println("Enhance Feature: Electric");

   }

}

* DecoratorPatternDemo.java

package decoratorDP;

public class DecoratorPatternDemo {

   public static *void* main(*String*[] *args*) {

*Instrument* guitar = new Guitar();

*Instrument* electricGuitar = new EnhancedInstrumentDecorator(new Guitar());

*Instrument* electricDrum = new EnhancedInstrumentDecorator(new Drum());

      System.out.println("Acoustic Guitar : Basic Version");

      guitar.play();

      System.out.println("\nElectric Guitar : Enhanced Version");

      electricGuitar.play();

      System.out.println("\nElectric Drum : Enhanced Version");

      electricDrum.play();

   }

}

OUTPUT:

