

Experiment – 2

BUILDER DESIGN PATTERN

SHRUTI MISHRA

21BCP110

CODE:

Instrument.java

```
public class Instrument {
    private String Type;
    private String Feature;
    private int Warranty;
    private int Quantity;

    public Instrument(String Type , String Feature , int Warranty , int
Quantity){

        super();
        this.Type = Type;
        this.Feature = Feature;
        this.Warranty = Warranty;
        this.Quantity = Quantity;
    }

    @Override
    public String toString() {
        return "Instrument [Type="+ Type + ", Feature="+ Feature + ",
Warranty="+ Warranty + ", Quantity="+ Quantity + "]";
    }
}
```

InstrumentBuilder.java

```
public class InstrumentBuilder{
    private String Type;
    private String Feature;
    private int Warranty;
    private int Quantity;

    public InstrumentBuilder setType(String Type){
        this.Type = Type;
        return this;
    }
}
```

```

    public InstrumentBuilder setFeature(String Feature){
        this.Feature = Feature;
        return this;
    }

    public InstrumentBuilder setWarranty(int Warranty){
        this.Warranty = Warranty;
        return this;
    }

    public InstrumentBuilder setQuantity(int Quantity){
        this.Quantity = Quantity;
        return this;
    }

    public Instrument getInstrument(){
        return new Instrument(Type , Feature ,Warranty, Quantity);
    }
}

```

MusicInstrument.java

```

//Main File

public class MusicInstrument {
    public static void main(String[] args) {
        Instrument h = new
InstrumentBuilder().setWarranty(2).setFeature("Clicks-
Buttons").setType("Keyboard").getInstrument();
        System.out.println(h);
    }
}

```

OUTPUT:

Instrument [Type=Keyboard, Feature=Clicks-Buttons, Warranty=2, Quantity=0]