KidsorHome

AirConditioner:

```
public class AirConditioner extends ElectronicProducts {
 private String airConditionerType;
 private double capacity;
 public AirConditioner(String productId, String productName, String batchId, String
dispatchDate, int warrantyYears, String airConditionerType, double capacity) {
 super(productId, productName, batchId, dispatchDate, warrantyYears);
 this.airConditionerType = airConditionerType;
 this.capacity = capacity;
 }    public String getAirConditionerType() {
 return airConditionerType;
 } public void setAirConditionerType(String airConditionerType) {
 this.airConditionerType = airConditionerType;
 }   public double getCapacity() {
 return capacity;
 } public void setCapacity(double capacity) {
 this.capacity = capacity;
 }
 public double calculateProductPrice(){
 double price = 0;
 if(airConditionerType.equalsIgnoreCase("Residential")){
 if (capacity == 2.5){
 price = 32000;
 } else if(capacity == 4){
 price = 40000;
 } else if(capacity == 5.5){
 price = 47000;
 }  }
 else if(airConditionerType.equalsIgnoreCase("Commercial"))
```

```
{   if (capacity == 2.5){
 price = 40000;
 } else if(capacity == 4){
 price = 55000;
 } else if(capacity == 5.5){
 price = 67000;
 }   }
 else if(airConditionerType.equalsIgnoreCase("Industrial")){
 if (capacity == 2.5){
 price = 47000;
 }  else if(capacity == 4){
 price = 60000;
 }  else if(capacity == 5.5){
 price = 70000;
 } }
 return price;
 } }
```

ElectronicProducts:

```
public class ElectronicProducts {
 protected String productId;
 protected String productName;
 protected String batchId;
 protected String dispatchDate;
 protected int warrantyYears;
 public ElectronicProducts(String productId, String productName, String batchId,
String dispatchDate, int warrantyYears) {
 this.productId = productId;
 this.productName = productName;
 this.batchId = batchId;
 this.dispatchDate = dispatchDate;
```

```
this.warrantyYears = warrantyYears;
}
public String getProductId() {
return productId;
} public void setProductId(String productId) {
this.productId = productId;
} public String getProductName() {
return productName;
} public void setProductName(String productName) {
this.productName = productName;
} public String getBatchId() {
return batchId;
} public void setBatchId(String batchId) {
this.batchId = batchId;
} public String getDispatchDate() {
return dispatchDate;
} public void setDispatchDate(String dispatchDate) {
this.dispatchDate = dispatchDate;
} public int getWarrantyYears() {
return warrantyYears;
} public void setWarrantyYears(int warrantyYears) {
this.warrantyYears = warrantyYears;
} }
```

LEDTV:

```
public class LEDTV extends ElectronicProducts {
private int size;
private String quality;
public LEDTV(String productId, String productName, String batchId, String
dispatchDate, int warrantyYears, int size, String quality) {
super(productId, productName, batchId, dispatchDate, warrantyYears);
```

```
this.size = size;
this.quality = quality;
}  public int getSize() {
return size;
}  public void setSize(int size) {
this.size = size;
}  public String getQuality() {
return quality;
}  public void setQuality(String quality) {
this.quality = quality;
}  public double calculateProductPrice(){
double price = 0;
if(quality.equalsIgnoreCase("Low")){
price = size * 850;
}  else if(quality.equalsIgnoreCase("Medium")){
price = size * 1250;
}  else if(quality.equalsIgnoreCase("High")){
price = size * 1550;
}
return price;
} }
```

MicrowaveOven:

```
public class MicrowaveOven extends ElectronicProducts{
 private int quantity;
 private String quality;
 public MicrowaveOven(String productId, String productName, String batchId, String
dispatchDate, int warrantyYears, int quantity, String quality) {
 super(productId, productName, batchId, dispatchDate, warrantyYears);
 this.quantity = quantity;
 this.quality = quality;
```

```
} public int getQuantity() {
return quantity;
} public void setQuantity(int quantity) {
this.quantity = quantity;
} public String getQuality() {
return quality;
} public void setQuality(String quality) {
this.quality = quality;
} public double calculateProductPrice(){
double price = 0;
if(quality.equalsIgnoreCase("Low")){
price = quantity * 1250;
} else if(quality.equalsIgnoreCase("Medium")){
price = quantity * 1750;
} else if(quality.equalsIgnoreCase("High")){
price = quantity * 2000;
}
return price;
} }
```

UserInterface:

```
import java.util.Scanner;
public class UserInterface {
 public static void main(String[] args) {
 Scanner sc = new Scanner(System.in);
 System.out.println("Enter Product Id");
 String productId = sc.next();
 System.out.println("Enter Product Name");
 String productName = sc.next();
 System.out.println("Enter Batch Id");
 String batchId = sc.next();
```

```
System.out.println("Enter Dispatch Date");
String dispatchDate = sc.next();
System.out.println("Enter Warranty Years");
int warrantyYears = sc.nextInt();
double price;
String quality;
switch(productName){
case "AirConditioner":
System.out.println("Enter type of Air Conditioner");
String type = sc.next();
System.out.println("Enter quantity");
double capacity = sc.nextDouble();
AirConditioner ac = new AirConditioner(productId, productName, batchId,
dispatchDate, warrantyYears, type, capacity);
price = ac.calculateProductPrice();
System.out.printf("Price of the product is %.2f", price);
break;
case "LEDTV":
System.out.println("Enter size in inches");
int size = sc.nextInt();
System.out.println("Enter quality");
quality = sc.next();
LEDTV l = new LEDTV(productId, productName, batchId, dispatchDate,
warrantyYears, size, quality);
price = l.calculateProductPrice();
System.out.printf("Price of the product is %.2f", price);
break;
case "MicrowaveOven":
System.out.println("Enter quantity");
int quantity = sc.nextInt();
```

```
 System.out.println("Enter quality");
 quality = sc.next();
 MicrowaveOven m = new MicrowaveOven(productId, productName, batchId,
dispatchDate, warrantyYears, quantity, quality);
 price = m.calculateProductPrice();
 System.out.printf("Price of the product is %.2f", price);
 break;
 default:
 System.out.println("Provide a valid Product name");
 System.exit(0);
 }
 }
}
```