Project-1

Step 1: Identify Products and Attributes

Identify 6 products and their attributes (name, price, units in stock, item number).

Example products:

1. Name: Office Chair, Price: 120.99, Units in Stock: 15, Item Number: 101

2. Name: Monitor, Price: 299.99, Units in Stock: 10, Item Number: 102

3. Name: Keyboard, Price: 49.99, Units in Stock: 50, Item Number: 103

4. Name: Mouse, Price: 25.99, Units in Stock: 75, Item Number: 104

5. Name: USB Drive, Price: 15.99, Units in Stock: 100, Item Number: 105

6. Name: Printer, Price: 199.99, Units in Stock: 8, Item Number: 106

Step 2: Identify Data Types

Add a column for data types.

Attribute	Sample Data	Data Type
Name of the product	Office Chair	String
Price	120.99	double
Number of units in stock	15	int
Item number	101	int

Step 3: Create Project

Create a project named inventory.

Step 4: Create Product Class

Create an object class called Product.

Step 5: Add Private Instance Fields

Add private instance fields for item number, name, units in stock, and price.

java

```
public class Product {
    // Instance field declarations
    private int itemNumber;
    private String name;
    private int unitsInStock;
    private double price;

    // Default constructor
    public Product() {
        // Initializes fields to default values
        this.itemNumber = 0;
        this.name = "";
        this.unitsInStock = 0;
        this.price = 0.0;
    }

    // Parameterized constructor
    public Product(int number, String name, int qty, double price) {
        this.itemNumber = number;
        this.name = name;
        this.unitsInStock = qty;
        this.price = price;
    }

    // Getter and Setter methods
    public int getItemNumber() {
        return itemNumber;
    }
```

```
public void setItemNumber(int itemNumber) {
   this.itemNumber = itemNumber;
}

public String getName() {
   return name;
}

public void setName(String name) {
   this.name = name;
}

public int getUnitsInStock() {
   return unitsInStock;
}

public void setUnitsInStock(int unitsInStock) {
   this.unitsInStock = unitsInStock;
}

public double getPrice() {
   return price;
}

public void setPrice(double price) {
   this.price = price;
}

// Override toString() method
@Override
```

```
    public String toString() {
        return "Item Number: " + itemNumber +
               "\nName: " + name +
               "\nQuantity in stock: " + unitsInStock +
               "\nPrice: " + price;
    }
}
```

Step 10: Create ProductTester Class

Create a Java main class called ProductTester.

Step 11: Create and Initialize Products

Initialize six Product objects.

java

```
public class ProductTester {
    public static void main(String[] args) {
        // Create products using default constructor
        Product defaultProduct1 = new Product();
        Product defaultProduct2 = new Product();

        // Create products using parameterized constructor
        Product product1 = new Product(101, "Office Chair", 15, 120.99);
        Product product2 = new Product(102, "Monitor", 10, 299.99);
        Product product3 = new Product(103, "Keyboard", 50, 49.99);
        Product product4 = new Product(104, "Mouse", 75, 25.99);

        // Display details of each product
        System.out.println(defaultProduct1.toString());
        System.out.println(defaultProduct2.toString());
        System.out.println(product1.toString());
        System.out.println(product2.toString());
        System.out.println(product3.toString());
        System.out.println(product4.toString());
    }
}
```

Step 12: Save Your Project

Ensure your project is saved in your IDE or text editor.