# JAVA FUNDAMENTALS SECTION 4: Creating an Inventory Project

**Topics:**

* **Data types**
* **Creating classes/objects**
* **Instance variables/fields**
* **Constructors**
* **Methods(getters/accessors, setters/mutators)**
* **Overloading**
* **Main/tester classes**
* **toString()**

|  |  |
| --- | --- |
| **Attribute** | **Sample Data** |
| Names of the product | Chairs, Ice creams, Paper Plates, Refreshments, Return Gifts |
| Price | 2890.76, 9006.54, 1500.00, 4567.86, 10000.00 |
| Number of units in the stock | 150, 500, 200, 600, 300 |
| Item number | 19156, 19157, 19158, 19159, 19160 |

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Sample Data** | **Data Type** |
| Name of the product | Chairs, Ice creams, Paper Plates, Refreshments, Return Gifts | String |
| Price | 2890.76, 9006.54, 1500.00, 4567.86, 10000.00 | double |
| Number of units in stock | 150, 500, 200, 600, 300 | int |
| Item number | 19156, 19157, 19158, 19159, 19160 | int |

MERCHANDISE INVENTORY SYSTEM

PROBLEM STATEMENT:

Design a Merchandise Inventory System that allows users to create and maintain merchandise items. Each merchandise item will have a unique catalog ID, description, quantity of stock, and list price. The system shall display information for each of the merchandise items.

REQUIREMENTS:

1. A class Merchandise with the following attributes:

• catalogID—A unique integer-based identifier for each merchandise

• description—A string-based, meaningful description of the merchandise item

• stockLevel—An integer representing the current stock level of the merchandise item

• listPrice—A double representing the list price for the merchandise item.

2. The system shall have a constructor which will initialize these attributes with default values.

3. Getter and setter methods shall be available within the system for each attribute.

4. The system shall support a toString method returning a string representation of the Merchandise Item, including its catalog ID, description, stock level, and list price.

5. The system shall have a main method where multiple merchandise items would be created and their details will be displayed.

CODE:

public class Merchandise {

private int catalogId;

private String description;

private int stockLevel;

private double listPrice;

public Merchandise() {

this.catalogId = 0;

this.description = "";

this.stockLevel = 0;

this.listPrice = 0.0;

}

public Merchandise(int catalogId, String description, int stockLevel, double listPrice) {

this.catalogId = catalogId;

this.description = description;

this.stockLevel = stockLevel;

this.listPrice = listPrice;

}

public int getCatalogId() {

return catalogId;

}

public void setCatalogId(int catalogId) {

this.catalogId = catalogId;

}

public String getDescription() {

return description;

}

public void setDescription(String description) {

this.description = description;

}

public int getStockLevel() {

return stockLevel;

}

public void setStockLevel(int stockLevel) {

this.stockLevel = stockLevel;

}

public double getListPrice() {

return listPrice;

}

public void setListPrice(double listPrice) {

this.listPrice = listPrice;

}

public String toString() {

return "Catalog ID: " + catalogId + "\n" +

"Description: " + description + "\n" +

"Stock Level: " + stockLevel + "\n" +

"List Price: " + String.format("%.2f", listPrice)+" rs.";

}

public static void main(String[] args) {

Merchandise item1 = new Merchandise(19156, "CHAIRS", 150, 2890.76);

Merchandise item2 = new Merchandise(19157, "ICECREAMS", 500, 9006.54);

Merchandise item3 = new Merchandise(19158, "PAPER PLATES", 200, 1500.00);

Merchandise item4 = new Merchandise(19159, "REFRESHMENTS", 600, 4567.86);

Merchandise item5 = new Merchandise(19160, "RETURN GIFTS", 300, 10000.00);

System.out.println(item1.toString());

System.out.println(item2.toString());

System.out.println(item3.toString());

System.out.println(item4.toString());

System.out.println(item5.toString());

}

}

OUTPUT:

