**WEEK – 1 : DESIGN PRINCIPLES AND PATTERN**

**EXERCISE 1: INVENTORY MANAGEMENT SYSTEM**

Inventory.java:

package com.example.inventory;

public class Main {

public static void main(String[] args) {

Inventory inventory = new Inventory();

inventory.addProduct(new Product(101, "Mouse", 50, 399.0));

inventory.addProduct(new Product(102, "Keyboard", 30, 699.0));

inventory.addProduct(new Product(103, "Monitor", 20, 8999.0));

System.***out***.println("Initial Inventory:");

inventory.displayInventory();

inventory.updateProduct(102, 25, 649.0);

inventory.deleteProduct(101);

System.***out***.println("\nUpdated Inventory:");

inventory.displayInventory();

}

}

Product.java:

package com.builder.example;

package com.example.inventory;

public class Product {

private int productId;

private String productName;

private int quantity;

private double price;

public Product(int productId, String productName, int quantity, double price) {

this.productId = productId;

this.productName = productName;

this.quantity = quantity;

this.price = price;

}

// Getters and setters

public int getProductId() { return productId; }

public String getProductName() { return productName; }

public int getQuantity() { return quantity; }

public double getPrice() { return price; }

public void setQuantity(int quantity) { this.quantity = quantity; }

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

*@Override*

public String toString() {

return productId + " - " + productName + " | Qty: " + quantity + " | Price: ₹" + price;

}

}

Main.java:

package com.example.inventory;

public class Main {

public static void main(String[] args) {

Inventory inventory = new Inventory();

inventory.addProduct(new Product(101, "Mouse", 50, 399.0));

inventory.addProduct(new Product(102, "Keyboard", 30, 699.0));

inventory.addProduct(new Product(103, "Monitor", 20, 8999.0));

System.***out***.println("Initial Inventory:");

inventory.displayInventory();

inventory.updateProduct(102, 25, 649.0);

inventory.deleteProduct(101);

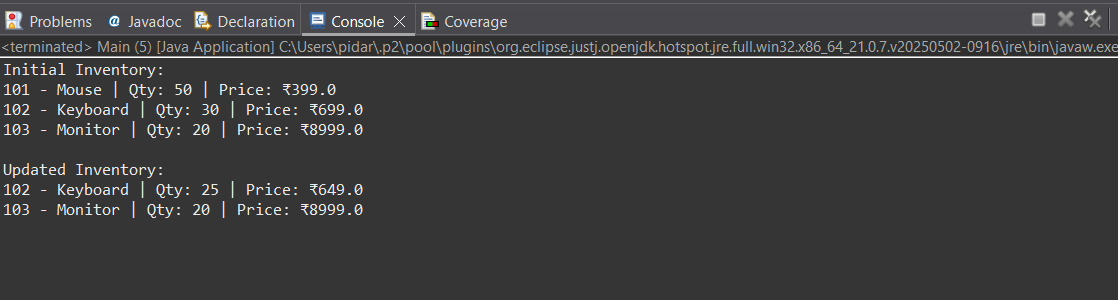
System.***out***.println("\nUpdated Inventory:");

inventory.displayInventory();

}

}

**OUTPUT:**



**EXERCISE 2: E-COMMERCE PLATFORM SEARCH FUNCTION**

Product.java:

package com.example.search;

public class Product {

private int productId;

private String productName;

private String category;

public Product(int id, String name, String category) {

this.productId = id;

this.productName = name;

this.category = category;

}

public int getProductId() { return productId; }

public String getProductName() { return productName; }

public String getCategory() { return category; }

*@Override*

public String toString() {

return productId + ": " + productName + " (" + category + ")";

}

}

Search.java:

package com.example.search;

public class Search {

// Linear Search

public static Product linearSearch(Product[] products, int targetId) {

for (Product p : products) {

if (p.getProductId() == targetId)

return p;

}

return null;

}

// Binary Search (array must be sorted by productId)

public static Product binarySearch(Product[] products, int targetId) {

int low = 0, high = products.length - 1;

while (low <= high) {

int mid = (low + high) / 2;

int midId = products[mid].getProductId();

if (midId == targetId)

return products[mid];

else if (midId < targetId)

low = mid + 1;

else

high = mid - 1;

}

return null;

}

}

Main.java:

package com.example.search;

import java.util.Arrays;

public class Main {

public static void main(String[] args) {

Product[] products = {

new Product(101, "Shoes", "Fashion"),

new Product(102, "Phone", "Electronics"),

new Product(103, "Book", "Education"),

new Product(104, "Watch", "Accessories"),

};

// Linear Search

Product result1 = Search.*linearSearch*(products, 103);

System.***out***.println("Linear Search Result: " + result1);

// Sort by productId before binary search

Arrays.*sort*(products, (a, b) -> Integer.*compare*(a.getProductId(), b.getProductId()));

Product result2 = Search.*binarySearch*(products, 103);

System.***out***.println("Binary Search Result: " + result2);

}

}

**OUTPUT:**

