class InvalidProductException extends Exception {

public InvalidProductException(String message) {

super(message);

}

}

// Product Class

class Product {

private int productId;

private String productName;

private double price;

private int quantity;

// Constructor with all fields

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

if (price < 0 || quantity < 0) {

throw new InvalidProductException("Price or quantity cannot be negative.");

}

this.productId = productId;

this.productName = productName;

this.price = price;

this.quantity = quantity;

}

public Product(int productId, String productName) {

this.productId = productId;

this.productName = productName;

this.price = 0.0;

this.quantity = 0;

}

// Method Overloading: Add Product with full details

public void addProduct(int productId, String productName, double price, int quantity) throws InvalidProductException {

if (price < 0 || quantity < 0) {

throw new InvalidProductException("Price or quantity cannot be negative.");

}

this.productId = productId;

this.productName = productName;

this.price = price;

this.quantity = quantity;

}

// Overloaded: Add product with name and price only

public void addProduct(String productName, double price) throws InvalidProductException {

if (price < 0) {

throw new InvalidProductException("Price cannot be negative.");

}

this.productName = productName;

this.price = price;

}

// Overloaded: Add product with name, price, and quantity

public void addProduct(String productName, double price, int quantity) throws InvalidProductException {

if (price < 0 || quantity < 0) {

throw new InvalidProductException("Price or quantity cannot be negative.");

}

this.productName = productName;

this.price = price;

this.quantity = quantity;

}

// Display method

public void displayProduct() {

System.out.println("Product ID : " + productId);

System.out.println("Product Name : " + productName);

System.out.println("Price : " + price);

System.out.println("Quantity : " + quantity);

System.out.println("----------------------------");

}

}

// Main Class

public class ProductManager {

public static void main(String[] args) {

try {

// Product with all fields (constructor)

Product p1 = new Product(101, "Laptop", 55000.00, 10);

// Product with only ID and name (constructor)

Product p2 = new Product(102, "Mouse");

// Use overloaded methods

Product p3 = new Product(0, "Temp"); // Temp constructor

p3.addProduct(103, "Keyboard", 1500.00, 20);

Product p4 = new Product(0, "Temp");

p4.addProduct("Monitor", 8000.00);

Product p5 = new Product(0, "Temp");

p5.addProduct("Webcam", 2500.00, 15);

// Display all products

System.out.println("=== Product List ===");

p1.displayProduct();

p2.displayProduct();

p3.displayProduct();

p4.displayProduct();

p5.displayProduct();

// Attempt to add invalid product (negative price)

Product p6 = new Product(104, "InvalidProduct");

p6.addProduct("Pen Drive", -500.00, 10); // This will throw exception

} catch (InvalidProductException e) {

System.out.println("Exception: " + e.getMessage());

}

}

}

OUTPUT

=== Product List ===

Product ID : 101

Product Name : Laptop

Price : 55000.0

Quantity : 10

----------------------------

Product ID : 102

Product Name : Mouse

Price : 0.0

Quantity : 0

----------------------------

Product ID : 103

Product Name : Keyboard

Price : 1500.0

Quantity : 20

----------------------------

Product ID : 0

Product Name : Monitor

Price : 8000.0

Quantity : 0

----------------------------

Product ID : 0

Product Name : Webcam

Price : 2500.0

Quantity : 15

----------------------------

Exception: Price or quantity cannot be negative.