Exercise 1: Implementing the Singleton Pattern

Code:

```
public class Main {


    static class Logger {
        private static Logger instance;


         private Logger() {
       System.out.println("Logger initialized.");
    }


          public static Logger getInstance() {
       if (instance == null) {
          instance = new Logger();
       }
       return instance;
    }


        public void log(String message) {
       System.out.println("[LOG]: " + message);
    }
}



  public static void main(String[] args) {
         Logger logger1 = Logger.getInstance();
    Logger logger2 = Logger.getInstance();


        logger1.log("This is the first log message.");
    logger2.log("This is the second log message.");
```

```
        if (logger1 == logger2) {
            System.out.println("Both logger1 and logger2 are the same instance.");
        } else {
            System.out.println("Different Logger instances detected!");
        }
    }
}
```

Result:

Exercise 1: Inventory Management System

Code:

```
import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;
```

```
public class InventorySystem {

    static class Product {
     int productId;
     String productName;
     int quantity;
     double price;

     Product(int productId, String productName, int quantity, double price) {
        this.productId = productId;
        this.productName = productName;
        this.quantity = quantity;
        this.price = price;
     }

     @Override
     public String toString() {
        return "ProductID: " + productId + ", Name: " + productName +
              ", Quantity: " + quantity + ", Price: $" + price;
     }
  }

     static class InventoryManager {
     private Map<Integer, Product> inventory;

     public InventoryManager() {
        inventory = new HashMap<>();
     }


     public void addProduct(Product product) {
```

```
    if (inventory.containsKey(product.productId)) {
        System.out.println("Product ID already exists. Use update instead.");
    } else {
        inventory.put(product.productId, product);
        System.out.println("Product added successfully.");
    }
}

      public void updateProduct(Product product) {
    if (inventory.containsKey(product.productId)) {
        inventory.put(product.productId, product);
        System.out.println("Product updated successfully.");
    } else {
        System.out.println("Product not found. Use add to create it.");
    }
}



public void deleteProduct(int productId) {
    if (inventory.remove(productId) != null) {
        System.out.println("Product deleted successfully.");
    } else {
        System.out.println("Product not found.");
    }
}



public void displayInventory() {
    if (inventory.isEmpty()) {
        System.out.println("Inventory is empty.");
    } else {
```

```
        for (Product product : inventory.values()) {
            System.out.println(product);
        }
    }
  }
}

  public static void main(String[] args) {
  InventoryManager manager = new InventoryManager();
  Scanner scanner = new Scanner(System.in);
  int choice;

  do {
    System.out.println("\nInventory Management System");
    System.out.println("1. Add Product");
    System.out.println("2. Update Product");
    System.out.println("3. Delete Product");
    System.out.println("4. Display Inventory");
    System.out.println("5. Exit");
    System.out.print("Enter your choice: ");
    choice = scanner.nextInt();

    switch (choice) {
      case 1 -> {
        System.out.print("Enter Product ID: ");
        int id = scanner.nextInt();
        scanner.nextLine(); // consume newline
        System.out.print("Enter Product Name: ");
        String name = scanner.nextLine();
        System.out.print("Enter Quantity: ");
        int qty = scanner.nextInt();
```

```
                System.out.print("Enter Price: ");
                double price = scanner.nextDouble();
                manager.addProduct(new Product(id, name, qty, price));
            }
            case 2 -> {
                System.out.print("Enter Product ID to update: ");
                int id = scanner.nextInt();
                scanner.nextLine(); // consume newline
                System.out.print("Enter New Product Name: ");
                String name = scanner.nextLine();
                System.out.print("Enter New Quantity: ");
                int qty = scanner.nextInt();
                System.out.print("Enter New Price: ");
                double price = scanner.nextDouble();
                manager.updateProduct(new Product(id, name, qty, price));
            }
            case 3 -> {
                System.out.print("Enter Product ID to delete: ");
                int id = scanner.nextInt();
                manager.deleteProduct(id);
            }
            case 4 -> manager.displayInventory();
            case 5 -> System.out.println("Exiting...");
            default -> System.out.println("Invalid choice. Try again.");
          }
        } while (choice != 5);

        scanner.close();
    }
}
```

Result:

```
PS C:\Users\harsh> ^C
PS C:\Users\harsh>
PS C:\Users\harsh>  & 'C:\Program Files\Eclipse Adoptium\jdk-21.0.5.11-hotspot\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\harsh\AppData\Local\Temp\vscodesws_c8662\jdt_ws\jdt.ls-java-project\bin' 'InventorySystem'

Inventory Management System
1. Add Product
2. Update Product
3. Delete Product
4. Display Inventory
5. Exit
Enter your choice: 1
Enter Product ID: 2
Enter Product Name: apple
Enter Quantity: 5
Enter Price: 1.5
Product added successfully.
```