**Exercise 1: Implementing the Singleton Pattern**

**Question:** You need to ensure that a logging utility class in your application has only one instance throughout the application lifecycle to ensure consistent logging.

**Code:**

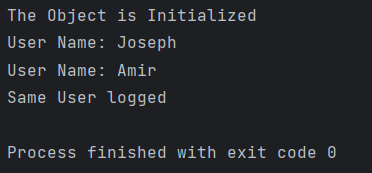
**Logger.java**

public class Logger {  
 private static Logger *object*;  
  
 private Logger(){  
 System.*out*.println("The Object is Initialized");  
 }  
  
 public static Logger getObject(){  
 if(*object* == null) *object* = new Logger();  
 return *object*;  
 }  
  
 public void display(String user){  
 System.*out*.println("User Name: " + user);  
 }  
}

**Main.java:**

public class Main {  
 public static void main(String[] args) {  
 Logger user\_1 = Logger.*getObject*();  
 user\_1.display("Joseph");  
  
 Logger user\_2 = Logger.*getObject*();  
 user\_2.display("Amir");  
  
 if(user\_1 == user\_2){  
 System.*out*.println("Same User logged");  
 }else{  
 System.*out*.println("Different User logged");  
 }  
 }  
}

**Output:**



**Exercise 2: Implementing the Factory Method Pattern**

**Question:** You are developing a document management system that needs to create different types of documents (e.g., Word, PDF, Excel). Use the Factory Method Pattern to achieve this.

**Code:**

**Document.java:**

public interface Document {  
 void open();  
}

**WordDocument.java:**

public class WordDocument implements Document{  
 @Override  
 public void open(){  
 System.*out*.println("Creating Word Document");  
 }  
}

**PdfDocument.java:**

public class PdfDocument implements Document{  
 @Override  
 public void open(){  
System.*out*.println("Creating PDF Document");

}  
}

**ExcelDocument.java:**

public class ExcelDocument implements Document{  
 @Override  
 public void open(){  
 System.*out*.println("Creating Excel Document");  
 }  
}

**DocumentFactory.java:**

public abstract class DocumentFactory {  
 public abstract Document createDocument();  
}

**WordFactory.java:**

public class WordFactory extends DocumentFactory{  
 @Override  
 public Document createDocument(){  
 return new WordDocument();  
 }  
}

**PdfFactory.java:**

public class PdfFactory extends DocumentFactory{  
 @Override  
 public Document createDocument(){  
 return new PdfDocument();  
 }  
}

**Main.java:**

public class Main {  
 public static void main(String[] args) {  
 DocumentFactory wordFactory = new WordFactory();  
 Document wordDoc = wordFactory.createDocument();  
 wordDoc.open();  
  
 DocumentFactory pdfFactory = new PdfFactory();  
 Document pdfDoc = pdfFactory.createDocument();  
 pdfDoc.open();  
  
 DocumentFactory excelFactory = new ExcelFactory();  
 Document excelDoc = excelFactory.createDocument();  
 excelDoc.open();  
 }  
}

**Output:**

