Week 1 Assignments - SOLID & Design Patterns

1. Singleton Pattern

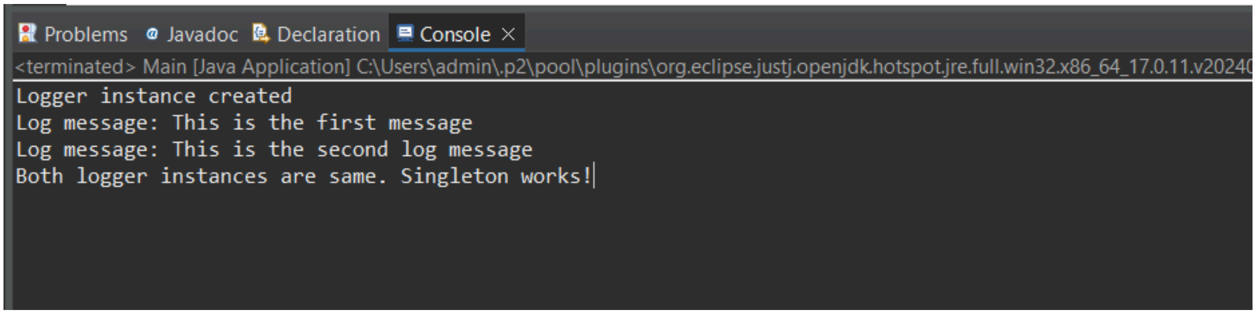
Logger.java

```
package SingletonPattern;
public class Logger {
   private static Logger instance;
   private Logger() {
       System.out.println("Logger instance created");
   }

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

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

}
```

Main.java

```
package SingletonPattern;
public class Main {

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

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

        if(logger1==logger2) {
            System.out.println("Both logger instances are same. Singleton works!");
        }
        else {
            System.out.println("Logger instances are different. Singleton failed!");
        }

    }

}
```

Output



<terminated> Main [Java Application] C:\Users\admin\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.11.v20240

```
Logger instance created
Log message: This is the first message
Log message: This is the second log message
Both logger instances are same. Singleton works!
```

2. Factory Patterns

Document.java

```
package factorymethodpattern;

public interface Document {
    void open();
}
```

WordDocument.java

```
package factorymethodpattern;

public class WordDocument implements Document{
    public void open() {
        System.out.println("Opening a Word Document");
    }

}
```

ExcelDocument.java

```
package factorymethodpattern;

public class ExcelDocument implements Document{
   public void open() {
       System.out.println("Opening a Excel Document");
   }
}
```

PdfDocument.java

```
package factorymethodpattern;

public class PdfDocument implements Document{
   public void open() {
       System.out.println("Opening a PDF Document");
   }
}
```

DocumentFactory.java

```
package factorymethodpattern;

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

PdfDocumentFactory.java

```
package factorymethodpattern;

public class PdfDocumentFactory extends DocumentFactory{
  public Document createDocument() {
      return new PdfDocument();
  }
}
```

ExcelDocumentFcatory.java

```
package factorymethodpattern;

public class ExcelDocumentFactory extends DocumentFactory{
      public Document createDocument() {
          return new ExcelDocument();
      }
    }
```

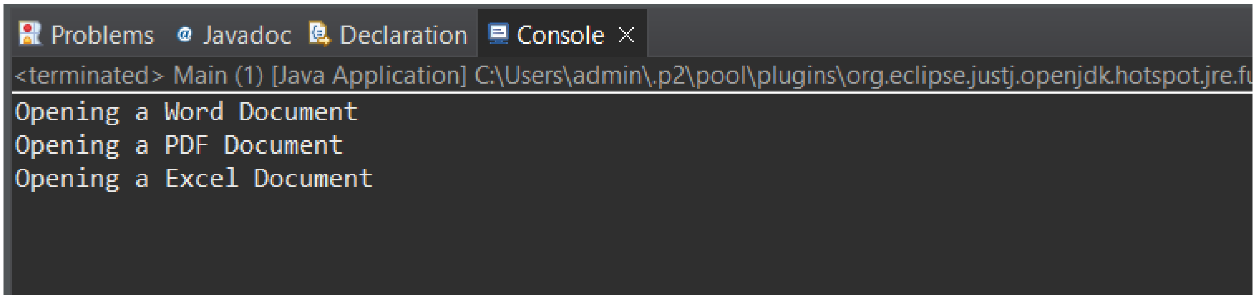
WordDocumentFactory.java

```
package factorymethodpattern;

public class WordDocumentFactory extends DocumentFactory{
      public Document createDocument(){
          return new WordDocument();
      }
}
```

Main.java

```
package factorymethodpattern;

public class Main {
    public static void main(String[] args) {
   DocumentFactory wordFactory = new WordDocumentFactory();
   Document wordDoc=wordFactory.createDocument();
   wordDoc.open();

   DocumentFactory pdfFactory = new PdfDocumentFactory();
   Document pdfDoc = pdfFactory.createDocument();
   pdfDoc.open();

   DocumentFactory excelFactory = new ExcelDocumentFactory();
   Document excelDoc = excelFactory.createDocument();
   excelDoc.open();
    }
}
```

Output



```
<terminated> Main (1) [Java Application] C:\Users\admin\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.fu
Opening a Word Document
Opening a PDF Document
Opening a Excel Document
```