Digital-Nurture-4.0-JAVA FSE

WEEK 1 ASSIGNMENTS

1) File: Design Patterns and Principles

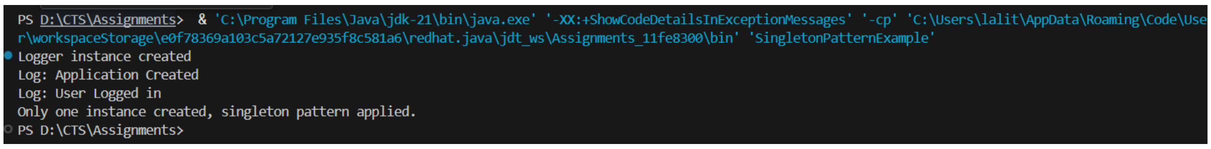
Exercise: Exercise 1: Implementing the Singleton Pattern

Code for the exercise:

```
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 m){
      System.out.println("Log: "+m);
   }
}
public class SingletonPatternExample {
   public static void main(String[] args) {
      logger logger1= logger.getInstance();
      logger1.log("Application Created");

      logger logger2= logger.getInstance();
      logger2.log("User Logged in");

      if(logger1==logger2){
         System.out.println("Only one instance created, singleton pattern applied.");
```

```
        }
        else{
            System.out.println("More than one instance found, singleton pattern not applied");
        }


    }
}
```

Output screenshot:



```
PS D:\CTS\Assignments>  & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\lalit\AppData\Roaming\Code\Use
r\workspaceStorage\e0f78369a103c5a72127e935f8c581a6\redhat.java\jdt_ws\Assignments_11fe8300\bin' 'SingletonPatternExample'
Logger instance created
Log: Application Created
Log: User Logged in
Only one instance created, singleton pattern applied.
PS D:\CTS\Assignments>
```

2) File: Design Patterns and Principles
Exercise: **Exercise 2: Implementing the Factory Method Pattern**

Code for the exercise:

- Document.java

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

- DocumentFactory.java

```
package FactoryMethodPatternExample;

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

- ExcelDocument.java

```
package FactoryMethodPatternExample;

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

- ExcelFactory.java

```
package FactoryMethodPatternExample;
```

```
public class ExcelFactory extends DocumentFactory {
   @Override
   public Document createDocument() {
      return new ExcelDocument();
   }
}
```

- PdfDocument.java

```
package FactoryMethodPatternExample;
public class PdfDocument implements Document {
   @Override
   public void open() {
      System.out.println("Opening PDF document...");
   }
}
```

- PdfFactory.java

```
package FactoryMethodPatternExample;

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

- WordDocument.java

```
package FactoryMethodPatternExample;

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

- WordFactory.java

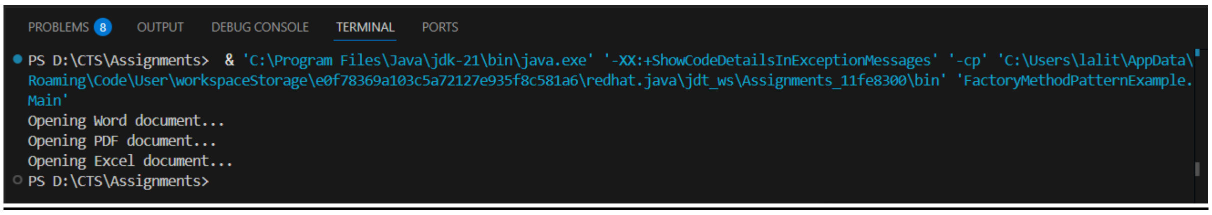
```
package FactoryMethodPatternExample;
public class WordFactory extends DocumentFactory {
   @Override
   public Document createDocument() {
      return new WordDocument();
   }
}
```

- Main.java

```
   package FactoryMethodPatternExample;
```

```java
public class Main {
    public static void main(String[] args) {
        // Word
        DocumentFactory wordFactory = new WordFactory();
        Document wordDoc = wordFactory.createDocument();
        wordDoc.open();

        // PDF
        DocumentFactory pdfFactory = new PdfFactory();
        Document pdfDoc = pdfFactory.createDocument();
        pdfDoc.open();

        // Excel
        DocumentFactory excelFactory = new ExcelFactory();
        Document excelDoc = excelFactory.createDocument();
        excelDoc.open();
    }
}
```

<u>Output Screenshots:</u>



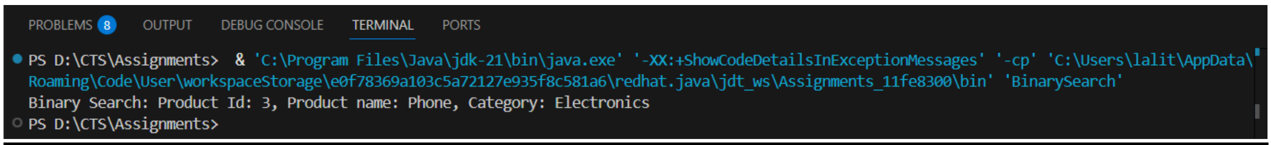
```
PS D:\CTS\Assignments>  & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\lalit\AppData\Roaming\Code\User\workspaceStorage\e0f78369a103c5a72127e935f8c581a6\redhat.java\jdt_ws\Assignments_11fe8300\bin' 'FactoryMethodPatternExample.Main'
Opening Word document...
Opening PDF document...
Opening Excel document...
PS D:\CTS\Assignments>
```

3) File: Algorithms Data Structures
Exercise: Exercise 2: E-commerce Platform Search Function

<u>Code for the exercise:</u>

```java
import java.util.*;
class product{
    int productId;
    String name;
    String cat;
    public product(int p, String n, String c){
        this.productId=p;
        this.name=n;
        this.cat=c;
    }

}
public class BinarySearch{
    public static product binarySearch(product[] pro, String ta){
        Arrays.sort(pro, Comparator.comparing(p -> p.name.toLowerCase()));
        int left=0;
        int right= pro.length-1;
        while(left<=right){
          int mid= (left+right)/2;
          int com= pro[mid].name.compareToIgnoreCase(ta);

          if(com==0){
            return pro[mid];
          }
          else if(com<0){
            left= mid+1;
          }
          else{
            right=mid-1;
          }
        }
        return null;
    }
    public static void main(String[] args) {
        product[] catalog = {
        new product(1, "Laptop", "Electronics"),
        new product(2, "Shirt", "Clothing"),
        new product(3, "Phone", "Electronics"),
        new product(4, "Book", "Education"),
        new product(5, "Watch", "Accessories")
    };
    product cat= binarySearch(catalog, "phone");
    if(cat!=null){
        System.out.println("Binary Search: "+"Product Id: "+cat.productId+", Product name:
           "+cat.name+", Category: "+cat.cat);
    }
    else{
        System.out.println("Binary Search: Not found");
    }
    }
}
```

Output Screenshots:



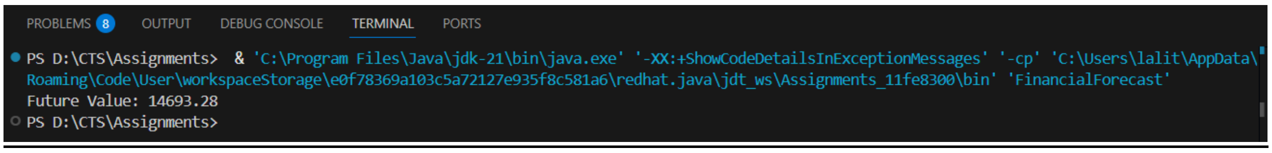
```
PS D:\CTS\Assignments>  & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\lalit\AppData\Roaming\Code\User\workspaceStorage\e0f78369a103c5a72127e935f8c581a6\redhat.java\jdt_ws\Assignments_11fe8300\bin' 'BinarySearch'
Binary Search: Product Id: 3, Product name: Phone, Category: Electronics
PS D:\CTS\Assignments>
```

4) File: Algorithms Data Structures
Exercise: Exercise 7: Financial Forecasting

<u>Code for the exercise:</u>

```
public class FinancialForecast {
    static double futVal(double CV, double GR, double y){
        if(y==0){
            return CV;
        }
        return futVal(CV, GR, y-1)*(1+GR);
    }
    public static void main(String[] args) {
        double c= 10000;
        double g=0.08;
        double y=5;
        double F= futVal(c, g, y);
        System.out.printf("Future Value: %.2f", F);
    }
}
```

<u>Output Screenshots:</u>



```
PS D:\CTS\Assignments>  & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\lalit\AppData\Roaming\Code\User\workspaceStorage\e0f78369a103c5a72127e935f8c581a6\redhat.java\jdt_ws\Assignments_11fe8300\bin' 'FinancialForecast'
Future Value: 14693.28
PS D:\CTS\Assignments>
```