HANDS-ON-EXERCISE

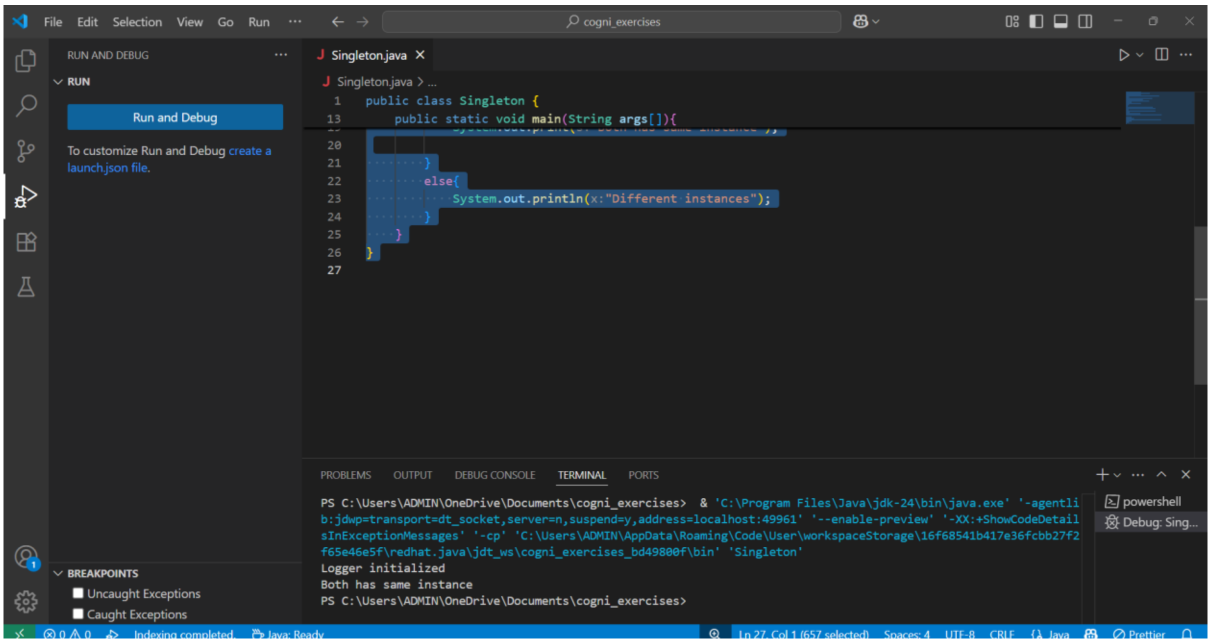
MODULE I- Designs patterns and principles

EX.NO.01 SINGLETON_PATTERN

CODE:

```
public class Singleton {
   private static Singleton instance;
   private Singleton(){
        System.out.println("Logger initialized");
   }
   public static Singleton getInstance(){
      if(instance==null){
         instance=new Singleton();
      }
      return instance;
   }

   public static void main(String args[]){
      Singleton log1=Singleton.getInstance();

      Singleton log2=Singleton.getInstance();

      if(log1==log2){
         System.out.print("Both has same instance");

      }
      else{
         System.out.println("Different instances");
      }}}
```

OUTPUT:



```java
public class Singleton {
    public static void main(String args[]){

        }
        else{
            System.out.println(x:"Different instances");
        }
    }
}
```

```
PS C:\Users\ADMIN\OneDrive\Documents\cogni_exercises>  & 'C:\Program Files\Java\jdk-24\bin\java.exe' '-agentli
b:jdwp=transport=dt_socket,server=n,suspend=y,address=localhost:49961' '--enable-preview' '-XX:+ShowCodeDetail
sInExceptionMessages' '-cp' 'C:\Users\ADMIN\AppData\Roaming\Code\User\workspaceStorage\16f68541b417e36fcbb27f2
f65e46e5f\redhat.java\jdt_ws\cogni_exercises_bd49800f\bin' 'Singleton'
Logger initialized
Both has same instance
PS C:\Users\ADMIN\OneDrive\Documents\cogni_exercises>
```

EX.NO.02 FACTORY_METHOD_PATTERN

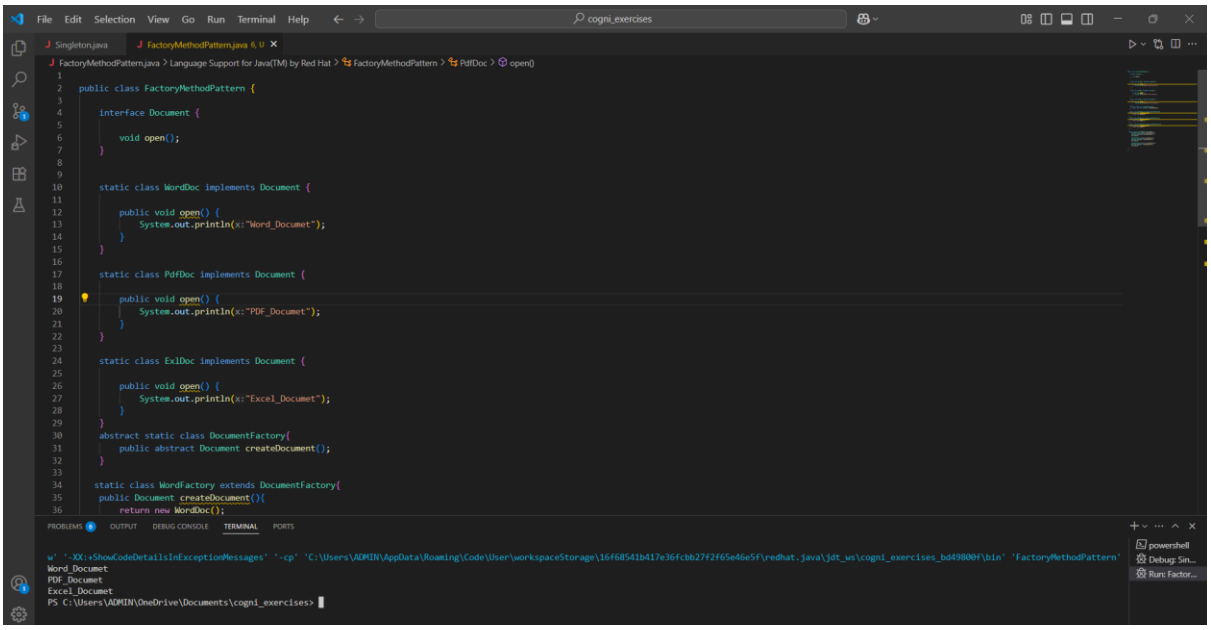
CODE:

```
public class FactoryMethodPattern {

  interface Document {

    void open();
  }


  static class WordDoc implements Document {

    public void open() {
      System.out.println("Word_Documet");
    }
  }

  static class PdfDoc implements Document {

    public void open() {
      System.out.println("PDF_Documet");
    }
  }

  static class ExlDoc implements Document {

    public void open() {
      System.out.println("Excel_Documet");
```

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

   static class WordFactory extends DocumentFactory{
    public Document createDocument(){
        return new WordDoc();
    }
  }
   static class PdfFactory extends DocumentFactory{
    public Document createDocument(){
        return new PdfDoc();
    }
  }
   static class ExcelFactory extends DocumentFactory{
    public Document createDocument(){
        return new ExlDoc();
    }
  }

  public static void main(String[] args) {
      DocumentFactory word=new WordFactory();
      Document doc1=word.createDocument();
      doc1.open();

      DocumentFactory pdf=new PdfFactory();
      Document doc2=pdf.createDocument();
```

```
        doc2.open();


        DocumentFactory exl=new ExcelFactory();
        Document doc3=exl.createDocument();
        doc3.open();


    }


}
```

OUTPUT:



```java
public class FactoryMethodPattern {

    interface Document {

        void open();
    }


    static class WordDoc implements Document {

        public void open() {
            System.out.println(x:"Word_Documet");
        }
    }

    static class PdfDoc implements Document {

        public void open() {
            System.out.println(x:"PDF_Documet");
        }
    }

    static class ExlDoc implements Document {

        public void open() {
            System.out.println(x:"Excel_Documet");
        }
    }
    abstract static class DocumentFactory{
        public abstract Document createDocument();
    }

   static class WordFactory extends DocumentFactory{
    public Document createDocument(){
        return new WordDoc();
```

```
w' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\ADMIN\AppData\Roaming\Code\User\workspaceStorage\16f68541b417e36fcbb27f2f65e46e5f\redhat.java\jdt_ws\cogni_exercises_bd49800f\bin' 'FactoryMethodPattern'
Word_Documet
PDF_Documet
Excel_Documet
PS C:\Users\ADMIN\OneDrive\Documents\cogni_exercises> 
```