**Exercise 1: Implementing the Singleton Pattern**

**Code:**

**Logger.java**

public class Logger {

    private static Logger instance;

    private Logger(){

        System.out.println("Logger initialization complete");

    }

    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**

public class Main {

    public static void main(String[] args){

        Logger logger1 = Logger.getInstance();

        logger1.log("First Log message created");

        Logger logger2 = Logger.getInstance();

        logger2.log("Second Log message created");

        if(logger1 == logger2){

            System.out.println("Singleton is confirmed as both logger1 and logger2 are same");

        }else{

            System.out.println("Different instances, Singleton pattern fialed.");

        }

    }

}

**Output:**

A screenshot of a computer program

AI-generated content may be incorrect.

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

**Code:**

**Document.java**

public interface Document {

    void open();

}

**DocumentFactory.java**

public abstract class DocumentFactory {

    public abstract Document createDocument();

}

**ExcelDocument.java**

public class ExcelDocument implements Document {

    @Override

    public void open() {

        System.out.println("Opening Excel Document.");

    }

}

**ExcelDocumentFactory.java**

public class ExcelDocumentFactory extends DocumentFactory {

    @Override

    public Document createDocument() {

        return new ExcelDocument();

    }

}

**Main.java**

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();

    }

}

**PdfDocument.java**

public class PdfDocument implements Document {

    @Override

    public void open() {

        System.out.println("Opening PDF Document.");

    }

}

**PdfDocumentFactory.java**  
class PdfDocumentFactory extends DocumentFactory {

    @Override

    public Document createDocument() {

        return new PdfDocument();

    }

}

WordDocument.java

public class WordDocument implements Document{

    @Override

    public void open(){

        System.out.println("Opening Word Document");

    }

}

WordDocumentFactory.java

public class WordDocumentFactory extends DocumentFactory {

    @Override

    public Document createDocument() {

        return new WordDocument();

    }

}

**Output:**

A screen shot of a computer program

AI-generated content may be incorrect.