CODE:

**Exercise 1: Implementing the Singleton Pattern**

Logger.java

public class Logger {

    private static Logger singleInstance;

    private Logger() {

        System.out.println("Logger initialized.");

    }

    public static Logger getInstance() {

        if (singleInstance == null) {

            singleInstance = new Logger();

        }

        return singleInstance;

    }

    public void log(String message) {

        System.out.println("Log message: " + message);

    }

}

Main.java  
import SingletonPatternExample.Logger;

public class Main {

    public static void main(String[] args) {

        Logger logger1 = Logger.getInstance();

        logger1.log("Kinshuk Ganguly's first log message");

        Logger logger2 = Logger.getInstance();

        logger2.log("Kinshuk Ganguly's first log message");

        if (logger1 == logger2) {

            System.out.println("Logger instances found to be same");

        } else {

            System.out.println("Logger instances found to be different");

        }

    }

}

OUTPUT



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

public interface Document {

void open();

}

public abstract class DocumentFactory {

    public abstract Document createDocument();

}

public class ExcelDocument implements Document {

    @Override

    public void open() {

        System.out.println("Opening an Excel document.");

    }

}

public class ExcelFactory extends DocumentFactory {

    @Override

    public Document createDocument() {

        return new ExcelDocument();

    }

}

public class PdfDocument implements Document {

    @Override

    public void open() {

        System.out.println("Opening a PDF document.");

    }

}

public class PdfFactory extends DocumentFactory {

    @Override

    public Document createDocument() {

        return new PdfDocument();

    }

}

public class WordDocument implements Document {

    @Override

    public void open() {

        System.out.println("Opening a Word document.");

    }

}

public class WordFactory extends DocumentFactory {

    @Override

    public Document createDocument() {

        return new WordDocument();

    }

}

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

