// Shape interface

public class Logger {

// private static variable of the same class that is the only instance of the class

private static Logger logger;

// private constructor to restrict instantiation of the class from other classes

private Logger() {}

// static method to provide the global point of access

public static Logger getInstance() {

if (logger == null) {

synchronized (Logger.class) {

if (logger == null) {

logger = new Logger();

}

}

}

return logger;

}

// method to log messages

public void log(String message) {

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

}

}

// Usage

public class Main {

public static void main(String[] args) {

Logger logger1 = Logger.getInstance();

Logger logger2 = Logger.getInstance();

logger1.log("This is a log message.");

System.out.println("Are logger1 and logger2 the same instance? " + (logger1 == logger2));

}

}

```

interface Shape {

void draw();

}

// Concrete classes implementing the same interface

class Circle implements Shape {

@Override

public void draw() {

System.out.println("Drawing a Circle");

}

}

class Square implements Shape {

@Override

public void draw() {

System.out.println("Drawing a Square");

}

}

// Factory class with a method to get objects of the concrete classes

class ShapeFactory {

// use getShape method to get object of type Shape

public Shape getShape(String shapeType) {

if (shapeType == null) {

return null;

}

if (shapeType.equalsIgnoreCase("CIRCLE")) {

return new Circle();

} else if (shapeType.equalsIgnoreCase("SQUARE")) {

return new Square();

}

return null;

}

}

// Usage

public class Main {

public static void main(String[] args) {

ShapeFactory shapeFactory = new ShapeFactory();

// Get an object of Circle and call its draw method

Shape shape1 = shapeFactory.getShape("CIRCLE");

shape1.draw();

// Get an object of Square and call its draw method

Shape shape2 = shapeFactory.getShape("SQUARE");

shape2.draw();

}

}