**WEEK-1**

**Design Principles And Patterns**

**Exercise 1: Implementing the Singleton Pattern:**

**Main.java**

public class Main {

static class StaticBlockSingleton {

private static final StaticBlockSingleton INSTANCE;

static {

try {

INSTANCE = new StaticBlockSingleton();

System.out.println("Singleton instance created using static block.");

} catch (Exception ex) {

throw new RuntimeException("Singleton initialization failed", ex);

}

}

private StaticBlockSingleton() {

}

public static StaticBlockSingleton getInstance() {

return INSTANCE;

}

public void displayMessage() {

System.out.println("Hello from StaticBlockSingleton!");

}

}

public static void main(String[] args) {

StaticBlockSingleton s1 = StaticBlockSingleton.getInstance();

StaticBlockSingleton s2 = StaticBlockSingleton.getInstance();

s1.displayMessage();

if (s1 == s2) {

System.out.println("Both references point to the same Singleton instance.");

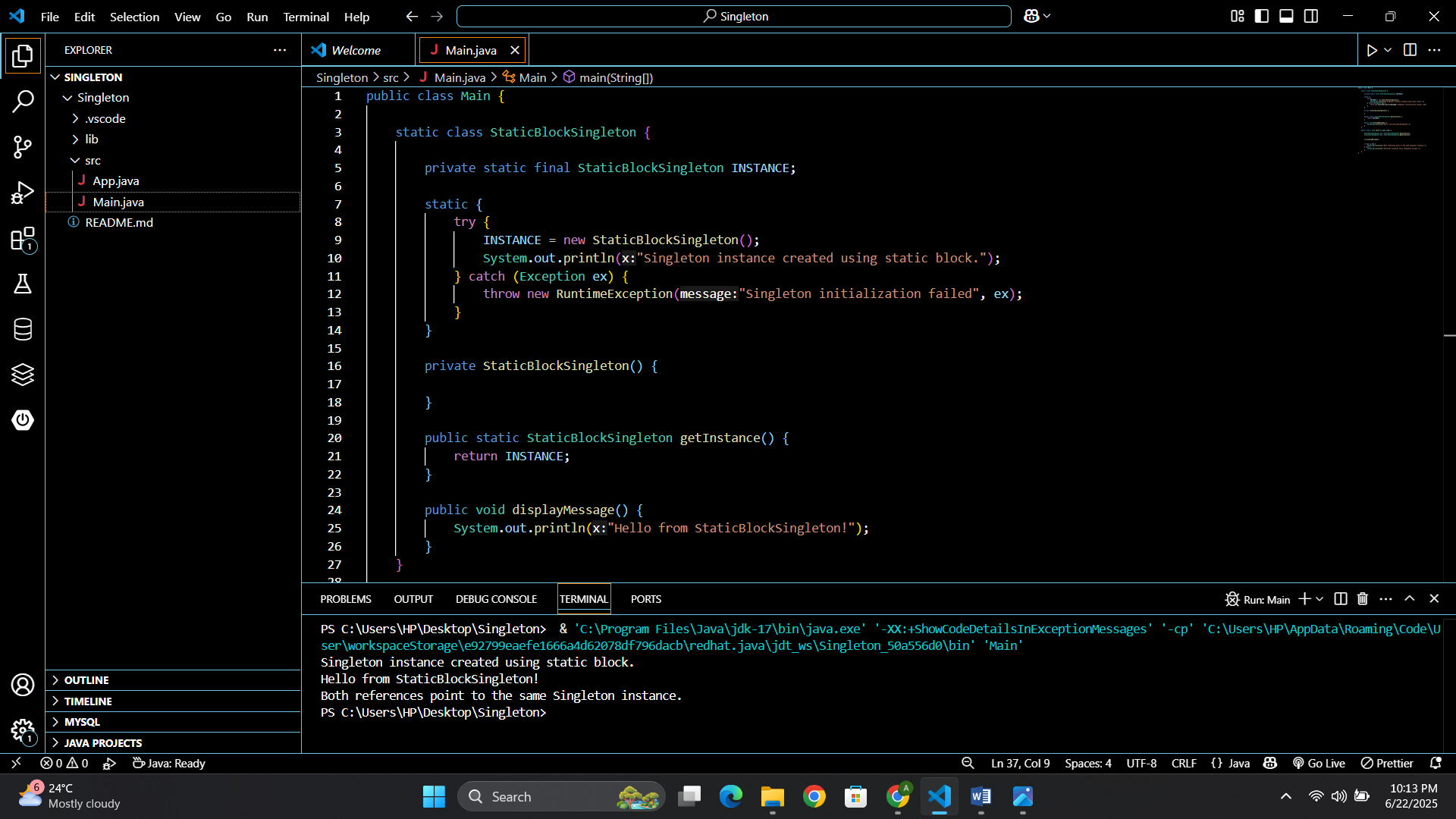
} else {

System.out.println("Different instances exist (Singleton failed).");

}

}

}



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

**Main.java**

public class Main {

interface Notification {

void notifyUser();

}

static class EmailNotification implements Notification {

public void notifyUser() {

System.out.println("Sending an Email Notification");

}

}

static class SMSNotification implements Notification {

public void notifyUser() {

System.out.println("Sending an SMS Notification");

}

}

static class PushNotification implements Notification {

public void notifyUser() {

System.out.println("Sending a Push Notification");

}

}

static class NotificationFactory {

public Notification createNotification(String type) {

if (type == null || type.isEmpty()) return null;

switch (type.toLowerCase()) {

case "email":

return new EmailNotification();

case "sms":

return new SMSNotification();

case "push":

return new PushNotification();

default:

return null;

}

}

}

public static void main(String[] args) {

NotificationFactory factory = new NotificationFactory();

Notification n1 = factory.createNotification("email");

Notification n2 = factory.createNotification("sms");

Notification n3 = factory.createNotification("push");

n1.notifyUser();

n2.notifyUser();

n3.notifyUser();

}

}

