**EXERCISE 2: IMPLEMENTING THE FACTORY METHOD PATTERN**

interface Notification {

    void notifyUser();

}

class SMSNotification implements Notification {

    public void notifyUser() {

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

    }

}

class EmailNotification implements Notification {

    public void notifyUser() {

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

    }

}

class PushNotification implements Notification {

    public void notifyUser() {

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

    }

}

class NotificationFactory {

    public Notification createNotification(String type) {

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

            return null;

        switch (type.toLowerCase()) {

            case "sms":

                return new SMSNotification();

            case "email":

                return new EmailNotification();

            case "push":

                return new PushNotification();

            default:

                throw new IllegalArgumentException("Unknown notification type: " + type);

        }

    }

}

public class Factory {

    public static void main(String[] args) {

        NotificationFactory factory = new NotificationFactory();

        Notification notification1 = factory.createNotification("email");

        notification1.notifyUser();

        Notification notification2 = factory.createNotification("sms");

        notification2.notifyUser();

        Notification notification3 = factory.createNotification("push");

        notification3.notifyUser();

    }

}

**OUTPUT:**

