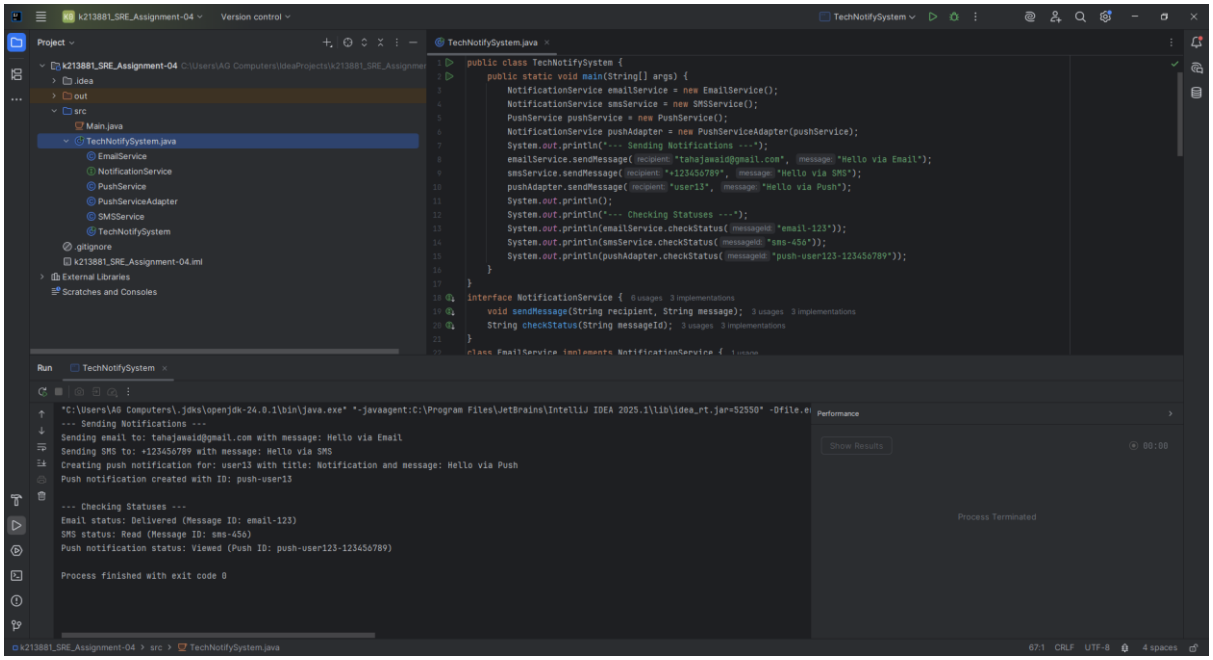
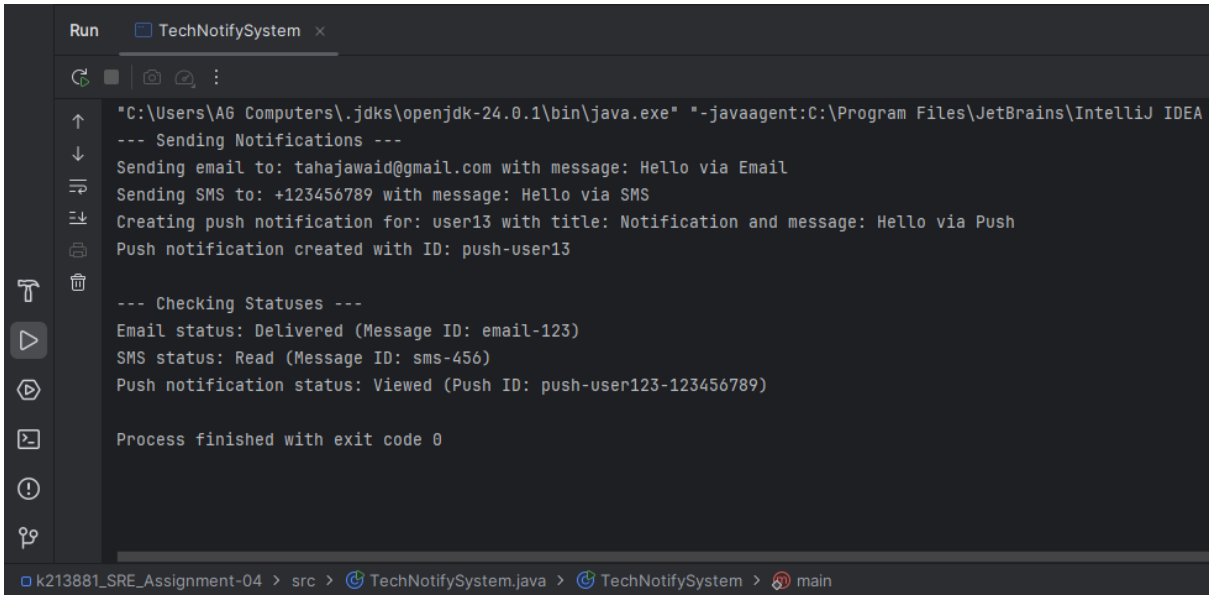
Question 01:

```
public class TechNotifySystem {
        public static void main(String[] args) {
           NotificationService emailService = new EmailService();
           NotificationService smsService = new SMSService();
           PushService pushService = new PushService();
           NotificationService pushAdapter = new PushServiceAdapter(pushService);
           System.out.println("--- Sending Notifications ---");
           emailService.sendMessage("tahajawaid@gmail.com", "Hello via Email");
           smsService.sendMessage("+123456789", "Hello via SMS");
           pushAdapter.sendMessage("user13", "Hello via Push");
           System.out.println();
           System.out.println("--- Checking Statuses ---");
           System.out.println(emailService.checkStatus("email-123"));
           System.out.println(smsService.checkStatus("sms-456"));
           System.out.println(pushAdapter.checkStatus("push-user123-123456789"));
        }
       }
interface NotificationService {
 void sendMessage(String recipient, String message);
 String checkStatus(String messageId);
}
class EmailService implements NotificationService {
 @Override
 public void sendMessage(String recipient, String message) {
    System.out.println("Sending email to: " + recipient + " with message: " + message);
 }
 @Override
 public String checkStatus(String messageId) {
```

```
    return "Email status: Delivered (Message ID: " + messageId + ")";
  }
}
class SMSService implements NotificationService {
  @Override
  public void sendMessage(String recipient, String message) {
    System.out.println("Sending SMS to: " + recipient + " with message: " + message);
  }
  @Override
  public String checkStatus(String messageId) {
    return "SMS status: Read (Message ID: " + messageId + ")";
  }
}
class PushService {
  public String createPushNotification(String recipientId, String title, String message) {
    System.out.println("Creating push notification for: " + recipientId +
                " with title: " + title + " and message: " + message);
    return "push-" + recipientId;
  }
  public String checkPushNotificationStatus(String pushId) {
    return "Push notification status: Viewed (Push ID: " + pushId + ")";
  }
}
class PushServiceAdapter implements NotificationService {
  private final PushService pushService;
  public PushServiceAdapter(PushService pushService) {
    this.pushService = pushService;
  }
  @Override
```

```
    public void sendMessage(String recipient, String message) {
        String pushId = pushService.createPushNotification(recipient, "Notification", message);
        System.out.println("Push notification created with ID: " + pushId);
    }
    @Override
    public String checkStatus(String messageId) {
        return pushService.checkPushNotificationStatus(messageId);
    }
}
```





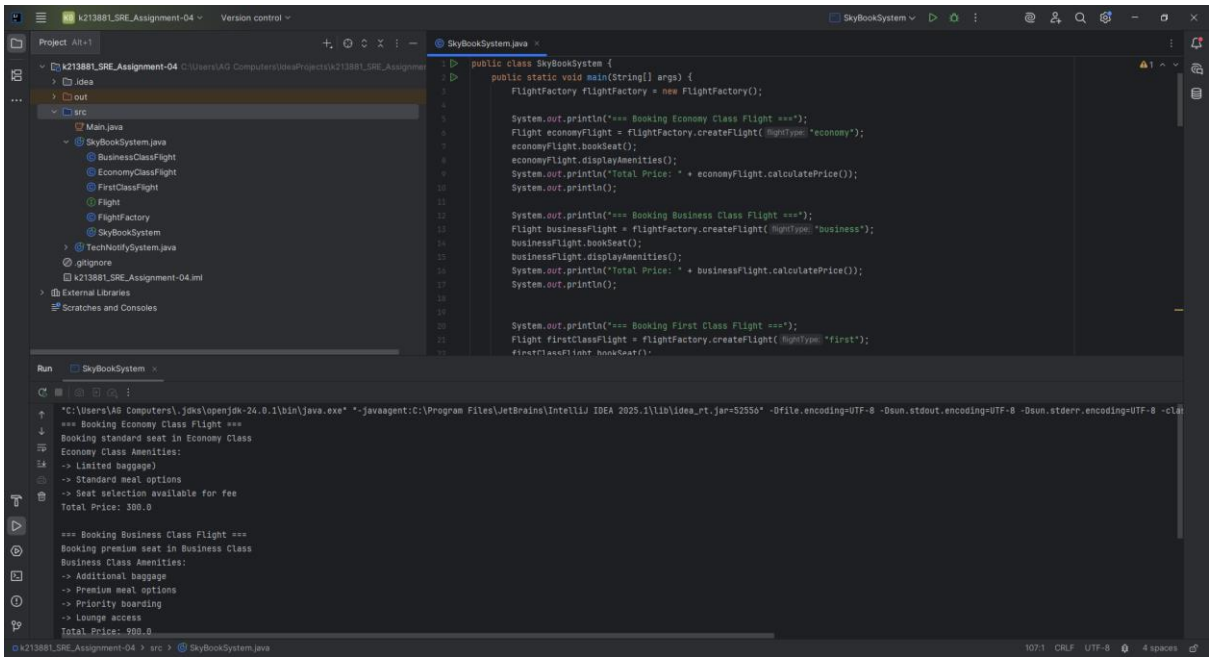
Question 02:

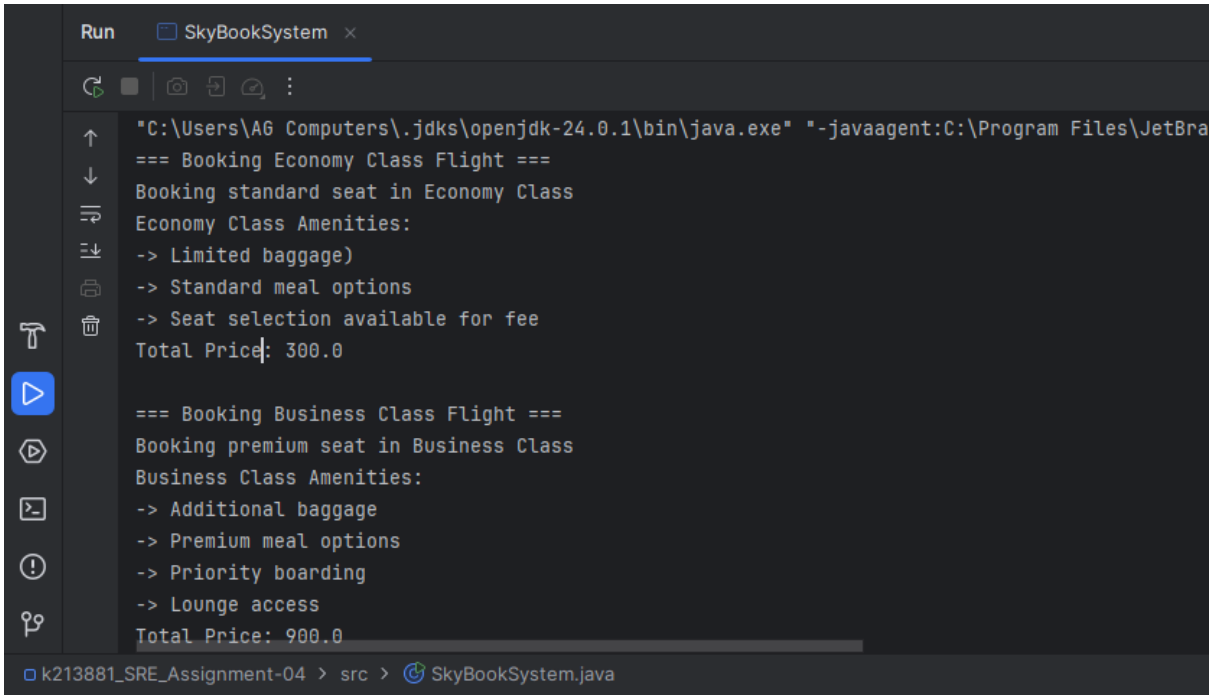
```
public class SkyBookSystem {
        public static void main(String[] args) {
          FlightFactory flightFactory = new FlightFactory();

          System.out.println("=== Booking Economy Class Flight ===");
          Flight economyFlight = flightFactory.createFlight("economy");
          economyFlight.bookSeat();
          economyFlight.displayAmenities();
          System.out.println("Total Price: " + economyFlight.calculatePrice());
          System.out.println();

          System.out.println("=== Booking Business Class Flight ===");
          Flight businessFlight = flightFactory.createFlight("business");
          businessFlight.bookSeat();
          businessFlight.displayAmenities();
          System.out.println("Total Price: " + businessFlight.calculatePrice());
          System.out.println();

          System.out.println("=== Booking First Class Flight ===");
          Flight firstClassFlight = flightFactory.createFlight("first");
          firstClassFlight.bookSeat();
          firstClassFlight.displayAmenities();
          System.out.println("Total Price: " + firstClassFlight.calculatePrice());

        }
        }
interface Flight {
 void bookSeat();
```

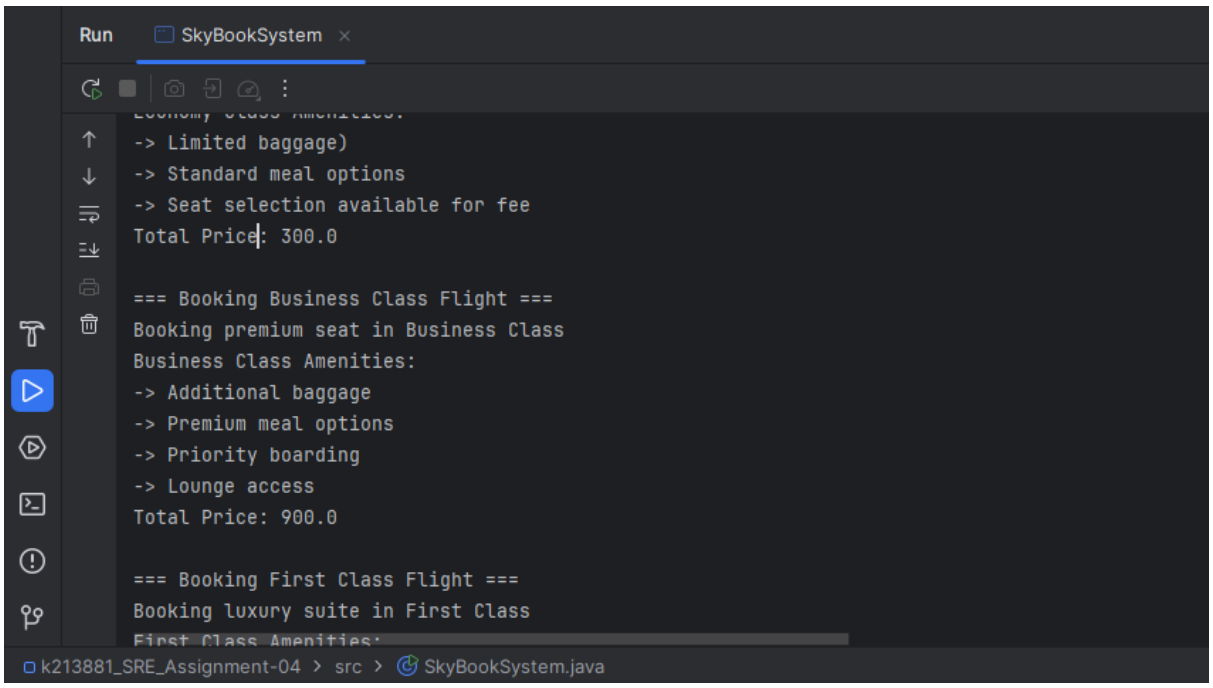
```
 double calculatePrice();
 void displayAmenities();
}
class EconomyClassFlight implements Flight {
 @Override
 public void bookSeat() {
    System.out.println("Booking standard seat in Economy Class");
 }
 @Override
 public double calculatePrice() {
    return 300.00;
 }

 @Override
 public void displayAmenities() {
    System.out.println("Economy Class Amenities:");
    System.out.println("-> Limited baggage)");
    System.out.println("-> Standard meal options");
    System.out.println("-> Seat selection available for fee");
 }
}
class BusinessClassFlight implements Flight {
 @Override
 public void bookSeat() {
    System.out.println("Booking premium seat in Business Class");
 }
 @Override
 public double calculatePrice() {
    return 900.00;
 }
 @Override
```

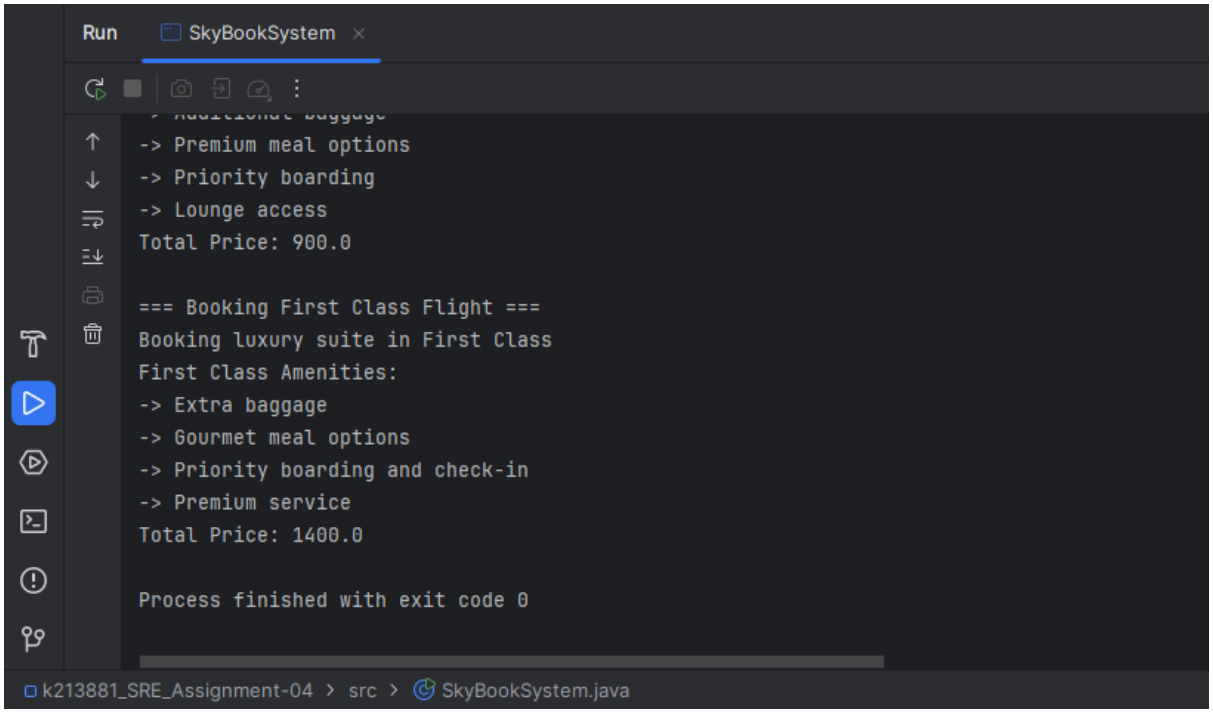
```
 public void displayAmenities() {
    System.out.println("Business Class Amenities:");
    System.out.println("-> Additional baggage");
    System.out.println("-> Premium meal options");
    System.out.println("-> Priority boarding");
    System.out.println("-> Lounge access");
 }
}
class FirstClassFlight implements Flight {
 @Override
 public void bookSeat() {
    System.out.println("Booking luxury suite in First Class");
 }

 @Override
 public double calculatePrice() {
    return 1400.00;
 }

 @Override
 public void displayAmenities() {
    System.out.println("First Class Amenities:");
    System.out.println("-> Extra baggage");
    System.out.println("-> Gourmet meal options");
    System.out.println("-> Priority boarding and check-in");
    System.out.println("-> Premium service");
 }
}
class FlightFactory {
 public Flight createFlight(String flightType) {
    if (flightType == null || flightType.isEmpty()) {
```

```
        return null;
    }

    switch (flightType.toLowerCase()) {
        case "economy":
            return new EconomyClassFlight();
        case "business":
            return new BusinessClassFlight();
        case "first":
            return new FirstClassFlight();
        default:
            throw new IllegalArgumentException("Unknown flight type: " + flightType);
    }
}
}
```




Run
SkyBookSystem
"C:\Users\AG Computers\.jdks\openjdk-24.0.1\bin\java.exe" "-javaagent:C:\Program Files\JetBra
=== Booking Economy Class Flight ===
Booking standard seat in Economy Class
Economy Class Amenities:
-> Limited baggage)
-> Standard meal options
-> Seat selection available for fee
Total Price: 300.0
=== Booking Business Class Flight ===
Booking premium seat in Business Class
Business Class Amenities:
-> Additional baggage
-> Premium meal options
-> Priority boarding
-> Lounge access
Total Price: 900.0
k213881_SRE_Assignment-04 > src > SkyBookSystem.java


Run
SkyBookSystem
-> Limited baggage)
-> Standard meal options
-> Seat selection available for fee
Total Price: 300.0
=== Booking Business Class Flight ===
Booking premium seat in Business Class
Business Class Amenities:
-> Additional baggage
-> Premium meal options
-> Priority boarding
-> Lounge access
Total Price: 900.0
=== Booking First Class Flight ===
Booking luxury suite in First Class
k213881_SRE_Assignment-04 > src > SkyBookSystem.java


Run
SkyBookSystem
-> Premium meal options
-> Priority boarding
-> Lounge access
Total Price: 900.0
=== Booking First Class Flight ===
Booking luxury suite in First Class
First Class Amenities:
-> Extra baggage
-> Gourmet meal options
-> Priority boarding and check-in
-> Premium service
Total Price: 1400.0
Process finished with exit code 0
k213881_SRE_Assignment-04 > src > SkyBookSystem.java

The End!!!