King Abdulaziz University
Faculty of Computing and Information Technology Rabigh
Spring2016

**Course Code: COCS203** **Course Name: Programming II**

---

# Assignment # 2

**Delivery Date and time:Monday11/4/2016 at 2:00PM**

**Submission from 2:00pm to 11:59pm will be considered as late submission (-1 mark)**
**No Submission accepted after Monday 11:59pm**

## Objectives

- Declaring classes
- Declaring data fields, constructors, setters and getters of the class
- Declaring methods to deal with data fields of the class
- Creating objects from classes
- Passing Object
- Array of Objects
- Aggregation relationship between Classes

## Delivery

- Submit your assignment to the lab instructor on Blackboard.
- Make sure to add your name / ID / Section / course name / Assignment number, as comment at the beginning of each Java file.
- The source file(s) of your program should be zipped up. You must name the zip file using the following naming convention:

**`SectionNumberStudentIDProjectNumber.zip`**

`Example: AAR134343P2.zip`

- For program discussion, you will receive an email mentioning the Venue discussion, date time and teacher name.
- You will be given 10 minutes to demonstrate your project. Questions asked will be related to the working of your program.
- Student who doesn't appear in the discussion will receive zero.

## Description

KAUFrS is a Booking system which can be used by any company to reserve seats for his customers. Sales staff of a company will use KAUFRS system for:

(1) Add Flight Details in the System,
(2) Add Passenger Details in the System,
(3) Make a new booking for a customer based on the request.

To use KAUFBS System, Sale staff has to set initial parameters one time, like Total (number) Flight system has to manage, Number of Passengers system has to create records and total booking system has to manage. [ see sample output file]

## The Initial Procedure of the Program

Initially your program will display the following:

System is going to set initial parameters, please provide the following details ...
Enter Total Flight available
4 suppose user enter **4**[ means system will accept only FOUR Flight details]
Now, Enter Total Passengers, System is going accommodate
10 suppose user enter **10**[ means system will accept only TEN Passenger details]
Lastly, Enter number of reservations System has to create
5 suppose user enter **5**[ means system will create only FIVE Reservation ]

## The Menu Procedure of the Program

```
##########################################
***** Welcome to KAU Flight Reservation System *****
1. Add Flight Details in the System
2. Add Passenger Details in the System
3. Make a new Booking
4. Search and Print a Booking
5. List Flight Status
6. Exit from the System
##########################################
Enter Your Choice
```

### (1) Add Flight details in the System

Your program should have an option (1), User will enter the Flight details by entering choice 1 from the listed menu, Based on number of Flight allowed in the system. [ see sample output file] .

### (2) Add Passenger Details in the System

Your program should have an option (2),User will enter Passenger details by entering option 2 from the listed menu. Based on number of Passengers allowed in the system [ see sample output file].

## (3) Make a new Booking

Your program should have an option (3), User will create a new booking for a customer by entering option 3 from the listed menu. Based on number of booking allowed in the system [ see sample output file].

When you make booking You have to list out all the flight available in the system along with their indexes , user will enter the flight index whom he want to create a booking. If the user enter an index which is not shown in your list you will display error message and allow user to enter the correct index of the flight from the listed index. [ see sample output file] Also, in booking process you have to enter passenger details, for this your program will ask user to enter number of seat [passenger] required in this booking, based on the number you will list all the passenger indexes along with their detail in the system, user has to enter the passenger index who want to travel in current booking.[ see sample output file].

If the user enter an index which is not shown in your list you will display error message and allow user to enter the correct index of the Passenger from the listed index.

Check has to be made that booking can only be done on allowed number of passengers and allowed seats in the flight. [ see sample output file].You will generate a unique PNR for every booking, PNR must start from JED100 onwards i.e. First booking JED100 , next booking JED101, Next JED102 .. so on ...[remember PNR is not fixed value]

## (4) Search and Print a Booking

Your program should have an option (4), User will Search and Print Booking for a customer by entering option 4 from the listed menu. [ see sample output file] , this option shows complete details of the booking, which includes flight details, All passenger details along total Booking price. [ see sample output file]

## (5) List Flight Status

Your program should have an option (5), User will print / display complete flight status using this option .You have to list out all the flight details for example available seats, city from and to etc. [ see sample output file]

**(6)** Exit from the KAU Booking System: this option will be used to end your program. , [see sample output file].

## The Implementation

1. You will need Flight.java class which should contain the details of the flight. **Data members** [**bold words**] required in the Flight class are as follows.

| **flightCode** | SV768 | **cityFrom** | Jeddah | **cityTo** | Dammam |
|---|---|---|---|---|---|
| **totalSeats** | 30 | **remmaningSeats** | 30 | **price** | 450.0 |

2. You will need another class Passenger.java to store passenger details. **Data members [bold words]** required in the Passenger class are as follows.

| **name** | Ahmed Zahrani | **age** | 24 | **gender** | M |
|---|---|---|---|---|---|
| **phone** | 05354703383 | | | | |

3. Also, You will need another class Reservation.java to store Item Reservation details. **Data members [bold words]** required in the **Reservation class** are as follows.

| **pNRNumber** | JED100 | **reservationDate** | 2015-02-26 |
|---|---|---|---|
| **Flight** | Object to access flight details | **Passenger []** | Array of Objects as there might be more than one passenger travelling in same booking. |

and you will also need AAR134343P2KAUFlightReservation.java class to test the Booking System. This is your main class in the application. Note [AAR134343P2 **will be different from student to student as it is sectionStudentIDP2**]

See the UML diagram to know complete data members and methods required in each class.

# The UML Diagram

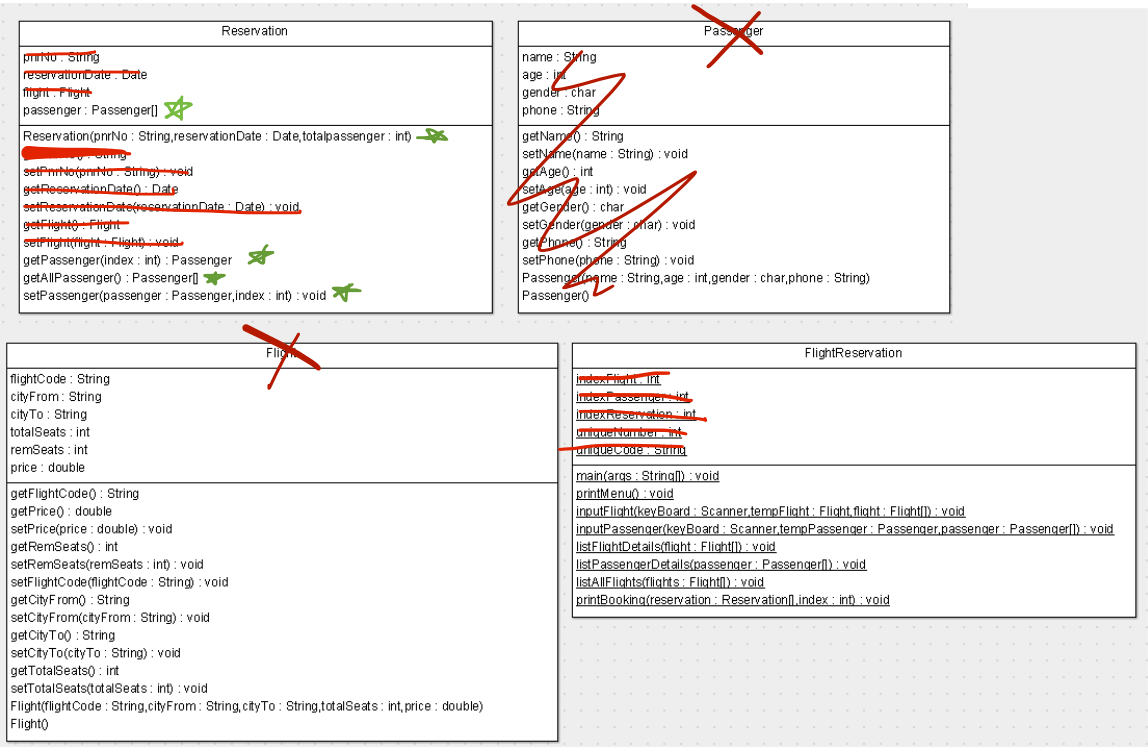


Figure 1: UML Diagram KAU Booking System

**Zoom to see the diagram**

# Methods and their description:

a) **public static void mainMenu():**

This method will be used generate system menu [see sample output file].

b) **public static void inputFlight(Scanner keyBoard, Flight tempFlight, Flight[] flight):**

This method will be used to enter Flight detail, since user can enter flight detail at any time while using user system, you have to maintain an index so that you know how many flight record user has already created, System only allows limited flight entry that you can check by the index[see sample output file].

c) **public static void inputPassenger(Scanner keyBoard, Passenger tempPassenger, Passenger[] passenger):**

This method will be used to enter Passenger detail, since user can enter passenger detail at any time while using user system, you have to maintain an index so that you know how many passenger record user has already created, System only allows limited passenger entry that you can check by the index[see sample output file].

d) **public static void listFlightDetails(Flight[] flight)**

supporting method to list Flight index along with Flight details [see sample output file].

e) **public static void listPassengerDetails (Passenger[] passenger)**

supporting method to list Passenger index along with Passenger details[see sample output file].

f) **public static void listAllFlights(Flight[] flights)**

List all flight details along with the current status of the seats. [see sample output file].

g) **public static void printBooking(Reservation[] reservation, int index)**

This functionality will print the complete reservation details

Other functionality of the classes is shown in UML diagram, see figure 1, mainly there are mutators and accessors in every class see figure 1.

[Hint] To allow user to only enter the index from the listed index, you use do while loop.

## Important Notes

- Your output program should be in a similar format to the sample run provided.
- Make your code readable.
- Repeat the program until the user select exit.
- Document your code with comments.
- Use meaningful variables.
- Use dash lines between each method.