

Airline Ticket Reservation System

Assignment 2

CPCS203 Programming-II - Term – Spring 2021

Assigned Date: 28/02/2021

Delivery Date: 20/03/2021 @ 11:00 PM

**Instructions**

* This program must ONLY be submitted on the Blackboard.
* This project worth 10% of the overall module marks (100%).
* NO assignment will be accepted after 11:59 pm for any reason.
* Students can submit their assignment between 11 and 11:59 PM but in this case it will be consider as late submission.
* For discussion schedule, check the captain name, date and time on the BlackBoard.
* Further information is provided in the course syllabus.
* **Objectives**
* Performing procedure on Objects and classes.
* Learn how to use and implement Class and Object concepts.
* Learn to use File I/O (Reading/Writing from/to files).
* **How to submit your assignment?**
* Submit your assignment on the Blackboard ONLY.
* Make sure to add your names / IDs / Section / Your name / Assignment number at the beginning of your program
* **Files provided with assignment**
* Input file samples:
  + **inputDB.txt**: which contains all Airlines, Airport, and Flight details that needs to be registered into the system.
  + **inputBookings.txt**: contains all the commands to generate air tickets. These commands are read from the file and processed by the system.
* Output files:
  + **printLog.txt**: This output file displays all the registered record for the airlines, airports and flights (The information in this file is read from **inputDB.txt)**.
  + **printBookings.txt**: This output file prints all the booked tickets with full details.

**Note: Please check the format of each of these files and make sure you follow this format in your assignment solution.**

**System Description**

This project is related to air industry by developing a computerized booking mechanism for reservation. This will help in reducing manual and paper work and therefore expected to enhance the overall booking experience both for the customers and travel agents. The airline reservation system is a booking solution that helps in integrating information from various airlines, airports and the flights. The system provides inventory and fare rates and is aimed to use by travel agents to generate the air tickets.

The system you are required to develop is called **Airline Ticket Reservation System** and is expected to assist the travel agents while making booking on behalf of customer. At the initial stage, the system will register all the available airlines, airports and flight details from **inputDB.txt.**  Information read from **inputDB.txt** are written with all the details into a log file, called **printLog.txt**.

After populating the data, the system will be ready to use and make bookings according to available flights. Since the data is already stored in the system, the travel agent will type minimum data while making a booking. For making a booking, the flight number, airline details, passenger details, and itinerary details are read from input file **inputBookings.txt**. After processing all bookings, the details is written to the output file **printBookings.txt**. Booking will include all the required details and fare will be calculated. The total fare will be automatically calculated as regular or business class. Different fare applicable for adult, child and infants. Similarly the system will add additional amount in the passenger exceeds the allowed baggage which again varies according to regular and business class.

For a more detailed description of the system and commands, please follow the next three steps which will explain how to develop the system.

**Step 1: Add Airline, Airport and Flight Records**

All airline, airport and flight details will be added into the system before making any booking. The first line read from ***inputDB.txt*** contains three integers, which determine the number of registered airlines, airports, and flights. For example, the total number of registered airlines is (6), the total number of airports is (9), and the total number of flights is (9). In the following, we describe the format of each command.

**1.1 Command: ADD\_Airline**

This command is used to add all information of the airline. This includes airline’s two letter code (such as SV, EK, BA etc.), airline name (such as Saudia, KLM, GulfAir etc.), country code of origin (such as SA, UE, BH etc), and website URL. Check the following example and table.

|  |
| --- |
| Command Example |
| ADD\_Airline SV Saudia SA saudia.com |

|  |  |  |
| --- | --- | --- |
| Field name | Type | Example |
| Two-letter airline code | String | SV |
| Airline name | String | Saudia |
| Country of origin code | String | SA |
| Website URL | String | Saudia.com |

**1.2 Command: ADD\_Airport**

This command will add airport details into the system.

|  |
| --- |
| Command Example |
| ADD\_Airport JED Jeddah KAIA |

|  |  |  |
| --- | --- | --- |
| Field name | Type | Example |
| Three-letter city code | String | JED |
| City name | String | Jeddah |
| Short airport name | String | KAIA |

**1.3 Command: ADD\_Flight**

This command will add flight details into the system.

|  |
| --- |
| Command Example |
| ADD\_Flight 432 JED DXB 1300 |

|  |  |  |
| --- | --- | --- |
| Field name | Type | Example |
| Flight number | String | 432 |
| Departure’s city code | String | JED |
| Arrival’s city code | String | DXB |
| Regular air fare | double | 1300 |

**1.4 Command: Quit**

The command quit will exit the process of entering the registration information.

**Step 2: Air Ticket Booking**

The booking details are provided in ***inputBookings.txt***. The first line of this file is an integer that determines the number of total bookings to be processed. For example, in the provided file ***inputBookings.txt***, 13 bookings need to be processed. In the following, a more detailed description of booking process has been explained.

**2.1 Command: NewBooking**

This command is used to create a new booking in the system. The command contains the necessary information of airline, airport and flight. For each booking, it reads the flight number, airline’s two-letter code, airport’s three-letter code, travel date, passenger name, passenger’s date of birth, business class status, and the baggage weight in kilograms the passenger wishes to carry.

|  |
| --- |
| Command Example |
| NewBooking 432 SV JED 2021 3 15 Ali 1972 9 23 true 55 |

|  |  |  |
| --- | --- | --- |
| Field name | Type | Example |
| Flight number | String | 432 |
| Two-letter airline code | String | SV |
| Three- letter city code | String | JED |
| Date of travel | Date | 2021 3 15 |
| Name of passenger | String | Ali |
| Date of birth of passenger | Date | 1972 9 23 |
| Business class | Boolean | true |
| Baggage weight | int | 55 |

Consider the following notes when issuing tickets:

|  |
| --- |
| Important Notes |
| * The system will read the ***Flight number*** as a string. You need to search for the flight object associated with the given flight number. |
| * The system will read the ***two-letter airline code*** as a string. You need to search for the airline object associated with the given code. |
| * The system will read the ***three-letter city code*** as a string. You need to search for the airport object associated with the given code. |
| * For each ticket, the system should generate a unique 13-digits time stamped ***booking number*** (Hint: Use System.currentTimeMillis() to generate it). Since it is a time stamped number, it will be different each time you run the program.   This ticket number will be printed in the ***printBookings.txt*** (check step 3). |
| * **You must calculate total air ticket fare amount** * **Consider all of the following factors while calculating fare.**   **Age**   |  |  |  |  | | --- | --- | --- | --- | | **Classification** | **Criteria** | **Fare** | **Example** | | Adult | More than 12 years | Regular fare | SAR 1000 | | Child | Less than or equal to 12 years | 70% of Regular fare | SAR 700 | | Infant | Less than or equal to 2 years | 10% of Regular fare | SAR 100 |   Hint: Calculate age by subtracting passenger’s date of birth from current date.  **Baggage Allowance**   |  |  |  | | --- | --- | --- | | **Cabin Class** | **Allowed Baggage (Free)** | **Excess Baggage Charges (for every extra KG)** | | Regular (Economy) | 30 KG | SAR 30/KG | | Business | 50 KG | SAR 30/KG |   **Cabin Class**   |  |  |  | | --- | --- | --- | | **Cabin Class** | **Fare** | **Example** | | Regular (Economy) | Regular | SAR 1000 | | Business | 1.4 times of regular | SAR 1400 | |

**1.2 Command: Quit**

The command quit will exit the process of entering the ticket information.

**Step 3: Print all the information**

* 1. **Print log of all added airline, airport, and flight records**

As mentioned earlier, all the added airline, airport, and flight records are read from **inputDB.txt** and are written to the file **printLog.txt.**

* 1. **Print all booking tickets with fare calculation.**

The system should print booking tickets. Moreover, the system will calculate the total fare amount for each ticket. For example, in step 2, once the system process the following command from the ***inputBookings.txt,*** the following information presented in the table below is written to the file ***printBookings.txt:***

|  |
| --- |
| Command Example |
| NewBooking 432 SV JED 2021 3 15 Ali 1972 9 23 true 55 |

|  |
| --- |
| Ticket details |
| Booking No. 1614039506786    Passenger Details Passenger Name: Ali Date of Birth: 1972-23-9    Flight Details Flight Code: SV-432 2021-15-3  Departure: JED Destination: DXB    Airport Details tAirport Name: KAIA (JED), Jeddah    Airline Details Name: Saudia based in saudia.com    Fare Details Regular Fare: SAR 1300.0  Total Fare: SAR 1970.0 |

Fare Calculation in above sample *(Business class fare applied)*

Fare = 1300 x 1.4 = 1820

Excess Baggage = 55 – 50 = 5 x 30 = 150

Total Fare = 1820 + 150 = SAR 1970

**1.2 UML Class Diagram**

In addition to the main class, you should create four classes as shown in the following UML diagram. Note that you should write appropriate constructor, setter, and getter methods for all classes. (You don’t need to follow the same given arguments). Be aware of the visibility (public-private) for each attribute/method.

Diagram

Description automatically generated

**Important Notes:**

* Use of class & object, arrays of Object, and passing object to method
* Use of Files, Reading/Writing from/on files
* Your program output must be exactly same as given sample output files.
* Your display should be in a readable form.
* Organize your code in separated methods.
* Document your code with comments.
* Use meaningful variables.
* Use dash lines between each method.
* **Delayed submission will not be accepted and there will not be any extension of**   
  **the project.**

**Deliverables:**

* You should submit one zip file containing all java codes: BA1887415P2\_EasyRent.java where BA is your section, 1887415 your ID and P2 is program 2.
* **NOTE: your name, ID, and section number should be included as comments in all files!**

**Input and Output Format**   
Your program must generate output in a similar format to the sample run provided. **Sample input:** See sample input file.  
**Sample output:** See sample output files.

**Good Luck!**