



Middle East Technical University Northern Cyprus Campus

CNG 443: Intr. to Object-Oriented Programming Languages and Systems Assignment 1: BASIC (*Bed And breakfaSt In Cyprus*)

Date handed-out: 23 October 2023, Monday

Date submission due: 6 November 2023, Monday 23:55 (Cyprus time)

Learning Outcomes

On successful completion of this assignment, a student will:

- Have written a class suitable for instantiation.
- Have written Javadoc comments for it.
- Have written code which creates and manipulates instances of this class.
- Have begun to appreciate the usefulness of reusing code, even within one class.
- Have developed a class based on use of an array as a means of storing a collection of objects.
- Have designed and written reusable methods for adding, searching for and deleting objects to/in/from the collection.
- Have defined and implemented appropriate test program to check the operation of the collection class.
- Have used a UML class diagram to implement an application.

Requirements

This assignment is about creating a small Java application for a Bed&Breakfast booking and management system which is called BASIC. This application will maintain users and also properties, and their hosts, and will manage bookings done on properties by users. The figure below shows a summary class diagram for this application.

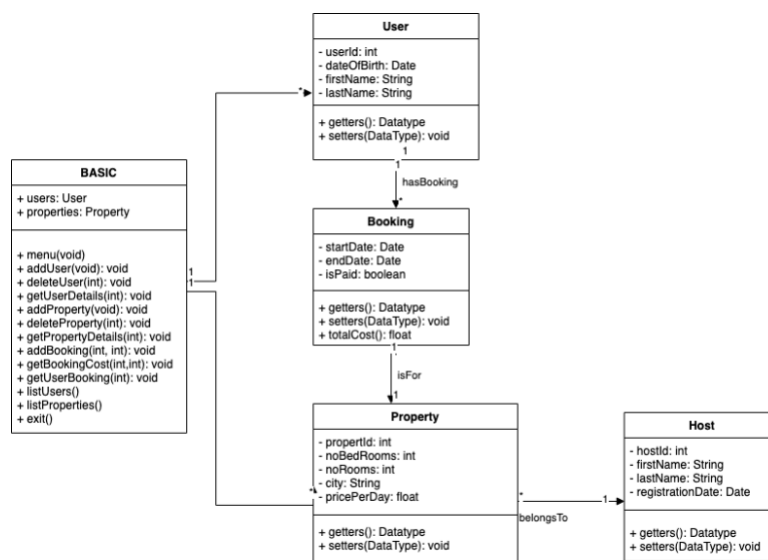


Figure 1 BASIC -- Class Diagram

The overall requirements are based on this class diagram, which is also summarised below:



Middle East Technical University Northern Cyprus Campus

- The main application called BASIC will be used to maintain information about users and the properties maintained by this application. BASIC will also have the main method and will provide the overall interaction with the application. The UML diagram given above includes arrows which shows in which way the classes interact. For example, the BASIC application which will have the main will maintain users and properties but not bookings. For instance, Booking is associated to a user and a property. Therefore, BASIC class should include the static main method where an instance of this class is constructed and the menu of commands is displayed to the user. Since we have not yet covered Graphical User Interfaces (GUI) in this course, you need to implement it as a command-line application. The required methods are as follows:
 - ***void menu()***: This method will display the interaction menu to the user.
 - ***void addUser()***: This method will add new user to the list of users maintained. Each user needs to have unique ID number.
 - ***void deleteUser(int userId)***: This method will read an ID number of a user, and delete the corresponding user object. If the user ID number does not exist, the program should provide an appropriate error message.
 - ***void getUserDetails(int userId)***: Given a user ID number, this method will display the user details. Please note that you should show the user ID, date of birth, first name and last name. If the ID number does not exist, the program should provide an appropriate error message.
 - ***void addProperty()***: This method will add a new property.
 - ***void deleteProperty(int propertyId)***: Given a property ID, this method will delete the given property. If the propertyID does not exist, the program should provide an appropriate error message.
 - ***void getPropertyDetails(int propertyID)***: Given a propertyID, this method will display the property details. It should show the property ID, no of bedrooms, no of room, city and the property price per day. If the propertyID does not exist, the method should provide an appropriate error message.
 - ***void addBooking(int userId, int propertyId)***: This method will record a booking for a given user and property. If the property with the given ID or the user with the given ID does not exist then the relevant error messages should be given. Please note that when a booking is added then all the relevant information should be asked to be recorded including the start date and endDate.
 - ***Void getUserBooking(int userId)***: This method will display the bookings done by a user. It will mainly display the booking start and end dates, if there are no bookings, then it should give the relevant feedback to the user.
 - ***void getBookingCost(int userId, int propertyId)***: This method will call to totalCost method on the Booking object. The method should first identify the booking object for a given user for a given property ID. If there are multiple bookings on a property by the same user, then you will need to display the total cost of each of those bookings. totalCost method on Booking will calculate the total number of days of the booking and will multiple it with the pricePerDay of the property. For



Middle East Technical University Northern Cyprus Campus

example, if the booking is for 1/06/2023 to 11/06/2023 then this booking is for 10 days. If the per day cost of the property is 50 then it will show 500.

- ***void listUsers()***: This method will list all the users. All user details should be displayed.
- ***void listProperties()***: This method will list all the properties. All property details should be displayed.
- ***void exit()***: This method should terminate the program.
- The given class diagram has all the fields and methods needed, so please follow the diagram. If you need extra fields, you can but please make sure that you update your class diagram.
- Since you did not learn how to make your class persistent or use a database, you will lose data every time you run your application. Therefore, you need to create some objects before you start your application. Your application needs to start with 3 user objects, 3 property objects, with each user having one booking and each property having one host. To create this data, you need to create a class which is called *PopulateData* that can be used to populate your application with these initial data. Please note that this is not given in the UML diagram but it is an extra class that needs to be created.
- Once you complete your implementation, fully update the UML class diagram, in case any changes are needed and submit it as well. Original UML diagram was created with Draw.io. You can use that or any other tool to create your updated UML diagram (e.g. Draw.io (www.draw.io), LucidChart (www.lucidchart.com/), Visio, etc.). This assignment also has an attachment that is the Visio version of this diagram so that you can import it to a tool and edit it.

Environment: As a development environment, you can use any IDE you like but you are strongly recommended to use **IntelliJ** (<https://www.jetbrains.com/idea/>).

Submission: Please organise your submission as a *single ZIP file* that includes the following:

- **[Jar file]**: A JAR file that can be executed on a command line. Make sure that BASIC is the main class.
- **[doc folder]**: This should include the full Javadocs generated.
- **[source folder]**: This should include your full source code. Please note that if you do not include *.java files, we cannot grade your work and you will automatically receive zero.
- **[diagram folder]**: This should include the updated UML.

If you are not following this submission structure, you will not receive marks from the Package item in the grading policy.



Middle East Technical University Northern Cyprus Campus

Extra Requirements:

Some additional requirements are listed below:

- We have not yet covered how to use a Database or make objects persistent in this course. Therefore, this assignment maintains objects such as property and users in ArrayLists.
- We have not yet covered Graphical User Interfaces (GUI) in this course. So please provide a command-line interaction (CLI).
- For each class, please decide what kind of constructors are required, the access types of methods and fields. If you use private fields, make sure that you provide accessor and mutators. For each class, you need to do constructor overloading and provide at least **two constructors**. Please note that the default constructor does not count so your code should include default plus two different versions.
- You should use the Date class provided in java.util in order to read the date from the user, read a string with the “dd/mm/yyyy” format, in which dd, mm and yyyy represent the day, month and the year, respectively. Study the “Parsing Strings into Dates” section provided in:
 - https://www.tutorialspoint.com/java/java_date_time.htm
- Pay attention to the overall design, layout and presentation of your code.
- You need to submit your Java code with proper Javadoc comments. For each class, you need to have used at least @author and @version, and for each method, you need at least @param and @return.

Assessment Criteria

This assignment will be marked as follows:

Aspect	Marks (Total 100)
All classes are implemented	25
Constructors are properly implemented for all classes. There should be at least two different constructors for each class (default does not count).	10
All methods are implemented	25
Command line interaction and menu	10
Initial data population is done (3 user objects, 3 property objects, with each user having one booking and each property having one host)	10
Package	10
<i>PopulateData</i> Class	5
UML Class Diagram	5

For each of the items above, we will also use the following grading criteria:

Half working	%20
Fully working	%20
Appropriate reuse of other code	%10
Good Javadoc comments	%10
Good quality code ¹	%40

¹ 15 principles will be considered during grading:

<https://www.informit.com/articles/article.aspx?p=2223710>