ITECH1000/5000 Programming 1 Semester 1 2017

Assignment 2

Overview

This is an individual assignment that requires you to design, develop and test a small Java program using object-oriented approaches.

Timelines and Expectations

Percentage Value of Task: 20%

Due: 16:00 Friday Week 11 (refer to the course description for your campus for the actual date)

Minimum time expectation: 20 hours

Learning Outcomes Assessed

The following course learning outcomes are assessed by completing this assessment:

- Identify and use the correct syntax of a common programming language
- Recall and use typical programming constructs to design and implement simple software solutions
- Reproduce and adapt commonly used basic algorithms
- Utilise pseudocode and/or algorithms as a major program design technique
- Write and implement a solution algorithm using basic programming constructs
- Demonstrate debugging and testing skills whilst writing code
- Develop self-reliance and judgement in adapting algorithms to diverse contexts
- Design and write program solutions to identified problems using accepted design constructs

Assessment Details

Your task is to design, develop and test a small application to assist a company in providing hotel recommendations for tourists and business travelers.

Stage 1: Design

This stage requires you to prepare documentation that describes the function of the program and how it is to be tested. There is no coding or code testing involved in this stage. A document template has been provided for your use.

Requirements:

1) Read through *Stage 2: Program Development* to obtain details of the requirements of this program.
2) Write an algorithm that describes how the program will operate.
    a. All program requirements must be included, even if you do not end up including all these requirements in your program code.

b. The algorithm must be structured logically so that the program would function correctly.

3) Prepare and document test cases that can be used to check that the program works correctly, once it has been coded. You do NOT need to actually run the test cases in this stage; this will occur in *Stage 3: Testing*.

   a. All program requirements must be included, even if you do not end up including all these requirements in your program code.

   b. Make sure the test cases include checking of data entered by the user to make sure that only valid data is accepted. If the user enters invalid data, the user should be informed of this and given another chance to enter the data. *NB: As we have not covered exception handling, you may assume that the user will always enter the expected data type.*

   c. Test cases should be documented using a template like the one below. You may include extra information if you wish. At this stage, the Actual Result column will be left blank.

Test Case	**Expected Result**	**Actual Result**
The user opts to display all hotel details.	A report is displayed that, for each hotel, lists that hotel's details (name, address, star rating), followed by the details of each of its room types (name, max occupancy, regular price and also the sale price if it differs from the regular price).	

Stage 2: Program Development

Using the Design Documentation to assist you, develop a Java program that uses object-oriented coding approaches to store the details of several hotels, and uses text-driven menus to allow a user to perform various queries based on that hotel data.

Overview of the Program:

This section provides an overview of how the program works from the user's perspective. You may change the appearance of the program provided you implement the same functionality.

1. When the program starts, it provides a short welcome message including your name and student number, and then automatically initialises the program. This initialisation includes creating the hotels, creating the room-types and registering each room-type with its associated hotel (the details of hotels and room-types are listed in the tables on the next page). When initialisation is complete, a menu appears providing the user with options to display all hotel details, find the cheapest room, set the sale price of a room, perform an advanced query based on multiple criteria, or exit the system.

```
Welcome to the Hotel Recommendation System
Developed by Peter Vamplew, student ID 19051251 for ITECH1000
Semester 1 2017

MAIN MENU
Please select an option from the menu:
1. Display all hotels
2. Find cheapest room
3. Set a sale price
4. Find rooms matching criteria
5. Exit System
```

Table 1: Details of hotels

Name	Address	Star rating	# of room types
El Grando	Lakeside Drive, San Diego	3	3
Ivory Tower	Pinehill Road, Boston	4	3
Elite	Federation Drive, Washington DC	5	4
Fleabox Motel	Wrong Side of the Tracks Road, Cleveland	1	1

Table 2: Details of room-types

Hotel	Room-type name	Regular price	Sale price	Max occupancy	Number of rooms	Number of vacancies
El Grando	Executive studio	210	210	2	20	17
El Grando	Standard studio	180	180	2	30	10
El Grando	Family room	220	200	4	10	2
Ivory Tower	Executive studio	235	235	2	5	3
Ivory Tower	Standard studio	205	205	2	12	0
Ivory Tower	Super-sized family room	350	350	6	4	1
Elite	Ultra-luxury suite	890	890	2	5	3
Elite	Elitist Studio	1400	1400	1	1	1
Elite	Penthouse Apartment	1200	1200	4	3	2
Elite	Gold-plated Luxury	1799	1799	2	10	8
Fleabox Motel	Sardine Room	80	55	8	120	25

2. When the user selects the Display All Hotels option, the program should output a report listing all of the hotels, and each of the room types existing at the hotel. The hotel name and address should be displayed inside a header (marked with == or some other symbol), along with the hotel's star rating, displayed via the correct number of asterisks (*). For each hotel, the details of each room type should be listed (name, maximum occupancy and regular price. The sale price should only be displayed if it differs from the regular price. The example below shows the report layout for just one of the hotels – your program should list all of the hotels. After the report is printed, the program should return to the main menu.

```
============================================================
El Grando, Lakeside Drive, San Diego  ***
============================================================

Room type: Executive studio
Maximum occupancy: 2
Regular price $210

Room type: Standard studio
Maximum occupancy: 2
Regular price $180

Room type: Family room
Maximum occupancy: 4
Regular price $220 Sale price $200
```

3. When the user chooses the Find Cheapest Room command, the program should search through all of the room-types at all of the hotels to find the cheapest option, and report the details of this back to the user. Note that your program must work correctly if the initialisation data is changed – that is, it is not sufficient to simply print out the details of the Fleabox Motel's Sardine Room without actually checking that it is the cheapest option. The program should use the sale price of each room-type, and does **not** need to check if the hotel actually has a vacancy for that type of room. After the report is printed, the program should return to the main menu.

```
MAIN MENU
Please select an option from the menu:
1. Display all hotels
2. Find cheapest room
3. Set a sale price
4. Find rooms matching criteria
5. Exit System
2

The cheapest rate available in any hotel is at Fleabox Motel, Wrong Side of
the Tracks Road, Cleveland  *
Room type: Sardine Room
Maximum occupancy: 8
Regular price $80 Sale price $55
```

4. When the user selects the option to Set a Sale Price, the program should prompt the user to enter the name of a hotel as a String. It should check that this actually matches one of the hotels in the system, and if not request a new name, until the user enters a valid name. At this point the program will display a numbered list of the names of the room-types existing at the specified hotel (this should be numbered starting at 1, not 0). The user should be prompted to select one of these room-types by entering a valid number. The program should then display the details of the selected room, and prompt the user to enter a sale price. The value entered must be between 50% and 100% of the regular price. Once a valid sale price has been entered, the object for that room-type should be updated with the new sale price. The main menu is then redisplayed.

```
MAIN MENU
Please select an option from the menu:
1. Display all hotels
2. Find cheapest room
3. Set a sale price
4. Find rooms matching criteria
5. Exit System
3

Setting sale price

Enter hotel name: Hyatt
No hotel matches that name. Please try again.
Enter hotel name: Elite

1: Ultra-luxury suite
2: Elitist Studio
3: Penthouse Apartment
4: Gold-plated Luxury
Enter room type number: 0
Value must be between 1 and 4. Please try again:
3
Regular price = $1200  Current sale price = $1200
Enter sale price: $400
Value must be between 600 and 1200. Please try again:
1300
Value must be between 600 and 1200. Please try again:
1199
Sale price updated.
```

5. When the user selects the option to Find Rooms Matching Criteria, the program should get them to enter valid values for each of the criteria (the minimum occupancy required, the minimum star-quality of the hotel, and the maximum price which the customer is willing to pay). The program should then search for and display the details of any rooms which satisfy these criteria and for which at least one vacancy exists. These should be grouped under a heading for each hotel. If no suitable room exists at a particular hotel, then no heading should be displayed for that hotel. After the report is shown, the program should return to the main menu.

```
Please enter the criteria which you require.
Minimum occupancy required: 2
Minimum star rating required: 3
Maximum daily price you are willing to pay: 1500
============================================================
El Grando, Lakeside Drive, San Diego  ***
============================================================

Room type: Executive studio
Maximum occupancy: 2
Regular price $210

Room type: Standard studio
Maximum occupancy: 2
Regular price $180

Room type: Family room
Maximum occupancy: 4
Regular price $220 Sale price $200

============================================================
Ivory Tower, Pinehill Road, Boston  ****
============================================================

Room type: Executive studio
Maximum occupancy: 2
Regular price $235

Room type: Super-sized family room
Maximum occupancy: 6
Regular price $350

============================================================
Elite, Federation Drive, Washington DC  *****
============================================================

Room type: Ultra-luxury suite
Maximum occupancy: 2
Regular price $890

Room type: Penthouse Apartment
Maximum occupancy: 4
Regular price $1200 Sale price $1199
```

6. When the user selects the Exit System option, the program should print out an appropriate message and exit.

```
5

Exiting system...
```

Technical Information:

This section provides technical implementation details required by the programmer to create the program.

Development of this program requires two classes: Hotel and RoomType, as well as a Driver class to control the flow of the program.

Driver Class: Name this file using ID<Your Student ID>. This will contain a main() method to manage the flow of the program, and other methods as necessary to ensure the code is modularized and functions correctly.

Hotel Class: This stores all of the information relevant for a single hotel. Use the class diagram below as the basis for designing your class.

Hotel
- name: String - address: String - starRating: int - roomTypes: RoomType[]
~ Hotel(String _name, String _address, int _starRating, int _numRoomTypes) + getName():String + getAddress():String + getStarRating():int + getNumRoomTypes():int + getRoomType(int): RoomType + setRoomType(RoomType): Boolean // store in next empty element of roomTypes array + toString():String // just include hotel details, not the room types

RoomType Class: This represents a specific type of room at a particular Hotel. References to objects of RoomType will be stored in the roomTypes array within each Hotel object.

RoomType
- name: String - regularPrice: int - salePrice: int - maximumOccupancy: int - numberOfRooms: int - numberOfVacancies: int
~ RoomType(String _name, int _regularPrice, int _salePrice, int _maximumOccupancy, int _numberOfRooms, int _numberOfVacancies) + getName(): String + getRegularPrice(): int + getSalePrice(): int + setSalePrice(int newPrice): void + int getMaximumOccupancy(): int + getNumberOfRooms(): int + getNumberofVacancies():int + toString(): String

Initialising the Program:

Initialising the program requires the following steps:

1. Create an array that can store 4 Hotel objects
2. Create a new Hotel for each of the hotels listed in Table 1 on page 3, defining their attribute values. Place each of these hotels into an element of the array from the previous step.
3. Create new RoomType objects for each entry in Table 2 on page 4. Each RoomType object should be added to the Hotel object to which it belongs.

Stage 3: Testing

Using a copy of the test cases developed in *Stage 1: Design*, test the program you have developed in *Stage 2: Program Development*. Document your results, including both failed and successful tests.

Note: Please do not leave out any failed tests. If your testing highlights that your program has not worked correctly, then the failed tests help to demonstrate that you have been testing your program properly.

To show that you have tested your program, include small (but readable) screen captures in your Actual Results as well as any explanatory comments. Microsoft Windows includes a Snipping Tool that is useful for taking captures of the relevant parts of the screen.

Submission

Your program code, design and testing documentation should be zipped into a single file and loaded into the Assignment Box provided in Moodle by the due date and time.

Marking Criteria/Rubric

Task	Available Marks	Student Mark
Stage 1: Design Documentation		
Development of an algorithm describing how the program should function • All requirements from the Assessment Details section included • Logical structure	1	
Documented test cases to validate program • All requirements from the Assessment Details section included • Data is validated to ensure user entries are appropriate and incorrect user entries are handled smoothly	1	
Stage 2: Program Development		
A Java program addressing the requirements outlined in the Assignment Details section, <u>including appropriate use of classes, instances, loops, conditional statements, constants and variables:</u>		
Initialisation of the program: • Creation and storage of Hotels and RoomTypes	2	
Running Program • Welcome message and appropriate menu display until user chooses to exit or return to previous menu	1	
• Displaying details of all hotels	2	
• Finding and displaying cheapest room	2	
• Setting sale price for a nominated hotel and room-type	3	
• Finding a room matching all specified criteria	3	
Coding Standards • Use of coding conventions throughout entire program, including readable and clear layout, following naming conventions and including meaningful and appropriate comments.	1	
• Code modularized, correctly using method calls and passing data between methods	1	
• Object-oriented approaches have been implemented appropriately	1	
Stage 3: Testing		
Documented test results clearly showing all the testing that has been conducted and the results of this testing.	2	
Total	20	

Feedback

Assignments will be marked within 2 weeks of submission. Marks will be loaded in fdlGrades, and a completed marking sheet will be available via Moodle.

Plagiarism:

Plagiarism is the presentation of the expressed thought or work of another person as though it is one's own without properly acknowledging that person. You must not allow other students to copy your work and must take care to safeguard against this happening. More information about the plagiarism policy and procedure for the university can be found at http://federation.edu.au/students/learning-and-study/online-help-with/plagiarism.