NGEE ANN POLYTECHNIC

SCHOOL OF INFOCOMM TECHNOLOGY

Diploma in Information Technology
Diploma in Cybersecurity & Digital Forensics
Diploma in Data Science

PROGRAMMING II

Year 2022/23 - Semester 2

ASSIGNMENT

Duration	:	2½ weeks (16 Jan 2023 to 01 Feb 2023)
Weightage	:	30% of total coursework
Individual/Team	:	Team of 2
Format	:	Programming - Class Implementation (10%) Basic Features (50%) Advanced Features (20%) Presentation (20%)

Cut-Off Date/Time: Wednesday, 01 February 2023, 8.00 am

Penalty for late submission:

- 10% of the marks will be deducted for each day (inclusive of Saturdays, Sundays and public holidays) after the deadline.
- No submission will be accepted after 08 Feb 2023, 8.00am.

There is a total of 10 pages (including this page) in this handout.

WARNING

If a student is found to have submitted work not done by him/her, he/she will not be awarded any marks for this assignment. Disciplinary action will also be taken.
Similar action will be taken for the student who allows other student(s) to copy his/her work.

Hotel Guest Management System

In this assignment, you are to apply Object Oriented Programming to develop a simple ***Hotel Guest Management System***. The assignment requirements described below are broken down into 2 stages of development, described in this document as '***Basic Features***' and '***Advanced Features***'. You are advised to do your programming progressively in these stages. Refer to the '***Grading Criteria***' to have an idea of how the different components are graded.

1. BACKGROUND

The ICT Hotel offers two types of rooms for stay, with different configuration. For basic Standard rooms, Wi-Fi and breakfast is not included in the daily rate and incurs an additional cost. For basic Deluxe room, Wi-Fi and breakfast is included, but guests can choose to add an additional bed should they require it. The room numbers, daily rate, add-on charges are shown in the Table 1 below (The data is also available in the data file ***Rooms.csv***).

Room Type	Bed Configuration	Room Numbers	No. of rooms	Daily rate (basic)	Add on (per day)
Standard	Single	101,102	2	90	• Wi-Fi $10 • Breakfast $20/room
	Twin	201,202,203	3	110	
	Triple	301,302	2	120	
Deluxe	Twin	204,205	2	140	• Additional bed $25
	Triple	303,304	2	210	

Table 1 – Information of the rooms

The hotel staff uses this system to perform all the administrative matters including register paying guest, check in and check out etc. If the room has an occupant, it will not be available for check-in. A room is released back into the pool of available rooms when check out.

For this system, the dates captured is merely a tool used to determine the duration of stay, and Check-Out manually happens when the hotel staff collects payment and select the **Check Out Guest** option, and not when the check-out date arrives.

The information of some registered paying guests is as shown in Table 2 below (More data is available in the data file ***Guests.csv***).

Name	Passport Number	Membership status	Membership points
Amelia	S1234567A	Gold	280
Bob	G1234567A	Ordinary	50
Cody	G2345678A	Silver	190

Table 2 – Registered paying guests

Some stay records are as shown in Table 3 below (More data is available in the data file ***Stays.csv***).

Name	Passport Number	Room Number	Extras	Check in	Check out
Amelia	S1234567A	101	• Wi-Fi • breakfast	15 Nov 2022	20 Nov 2022
Bob	G1234567A	302	• breakfast	25 Dec 2022	31 Dec 2022
Cody	G2345678A	202	• breakfast	26 Dec 2022	31 Dec 2022
Edda	S3456789A	303	• Extra bed	16 Jan 2023	26 Jan 2023
		204			

Table 3 – Stay records

A paying guest is auto-included into the membership programme the moment they register. A guest begins as an ordinary member and is only converted to a silver or gold membership once 100 or 200 points are earned respectively.

Points are earned from the final amount paid by guest for their stay at the hotel. The conversion rate for points to be earned is amount paid during check-out divided by 10. For example, if Amelia stays a total of 7 nights she pays 7 x $90 = $630 and earns 63 points.

Only silver and gold members can offset their bill payable by using their points. Each point converts to one dollar.

The class diagram for the Hotel Guest Management System is shown in Figure 1 on the next page.

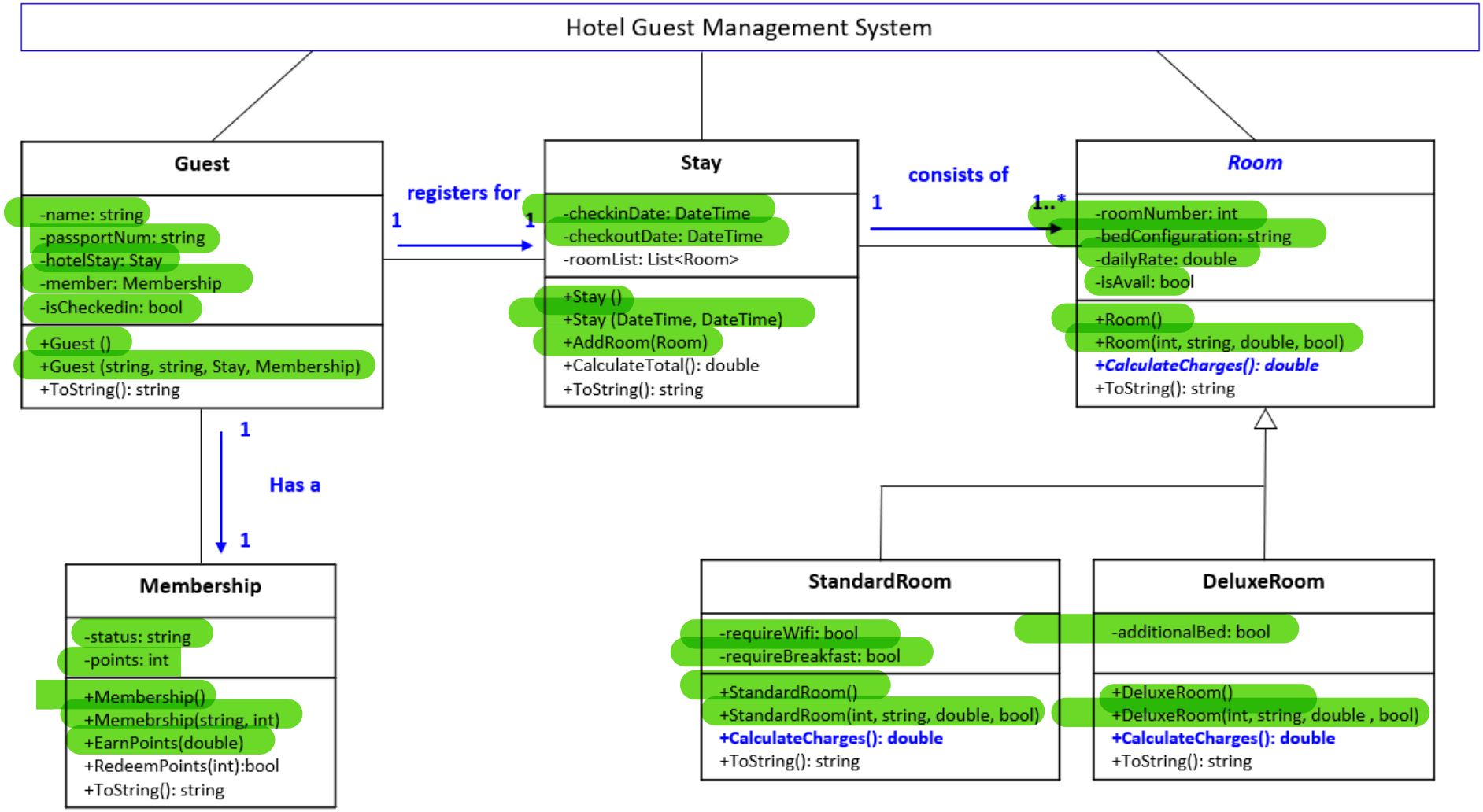

Figure 1: Class Diagram for Hotel Guest Management System

2. CLASS IMPLEMENTATION – 10% GROUP

The team has to divide the work when implementing the classes.

3. BASIC FEATURES – 50% INDIVIDUAL

Your program is required to create rooms, guests and stays from the 3 given data files at the onset. After which it displays a menu for user to choose to perform each of the feature describe below repeatedly until user chooses to exit from the menu.

1) List all guests

- *display the information of all the guests*

2) List all available rooms

- *display the information of all available rooms*

3) Register guest

- *prompt user for the following information for the guest: name, passport number*
- *create a guest object with the information given*
- *create membership object*
- *assign membership object to the guest*
- *add the guest object to the guest list*
- *append the guest information to the **guest.csv** file*
- *display a message to indicate registration status*

4) Check-in guest

- *list the guests*
- *prompt user to select a guest and retrieve the selected guest*
- *prompt user to enter check in date*
- *prompt user to enter check out date*
- *create stay object with the information given*
- *list all available rooms*
- *prompt user to select a room and retrieve the selected room*
- *prompt user for a response depending on the type of room selected:*
  - *Standard Room: require wifi [Y/N] & require breakfast [Y/N]*
  - *Deluxe Room: additional bed [Y/N]*
- *update the user response to the room object*
- *update the availability of the room*
- *add the room object to the room list of the stay*
- *prompt user whether want to select another room, if yes, repeat the above steps from list all available rooms. If no, proceed to next step*
- *assign the stay object to the guest*
- *update check in status of the guest*
- *display a message to indicate check in successfully*

5) Display stay details of a guest

- *List the guests*
- *prompt user to select a guest and retrieve the selected guest*
- *retrieve the stay object of the guest*
- *display all the details of the stay including check in date, check out date and all rooms details that he/she has checked in*

6) Extends the stay by numbers of day

- *List the guests*
- *prompt user to select a guest and retrieve the selected guest*
- *check if the guest is checked in*
- *prompt user for the number of day to extend*
- *retrieve the stay object of the guest*
- *compute and update the check out date of the stay*

- **Validations** (and feedback)
  - *The program should handle all invalid entries by the user*
    e.g. invalid option, invalid date, invalid S/No etc.
  - *If user made a mistake in the entry, the program should inform the user via appropriate feedback*

IMPORTANT INSTRUCTIONS:

- *A student is to implement features 1, 3 & 5, and another for features 2, 4 & 6.*
- *Individual student without a team is required to implement features 1, 2, 4 & 6.*

4. ADVANCED FEATURES – 20% INDIVIDUAL

(a) Display monthly charged amounts breakdown & total charged amounts for the year

- *prompt the user for the year*
- *retrieve the stay object and determine the month & year based on checked out date*
- *compute and display the monthly charged amounts breakdown & the total charged amounts for the input year*
- *Sample mock up screenshot:*

```
Enter the year: 2022

Jan 2022:   $...
Feb 2022:   $...
Mar 2022:   $...
Apr 2022:   $...
May 2022:   $...
Jun 2022:   $...
Jul 2022:   $...
Aug 2022:   $...
Sep 2022:   $...
Oct 2022:   $...
Nov 2022:   $...
Dec 2022:   $...

Total:      $...
```

Note: The month is based on the check out date. For example, if guest checked in on 28 Oct 2022 & checked out on 2 Nov 2022. This charged amounts falls under Nov 2022.

(b) Check out guest

- *list the guests*
- *prompt user to select a guest and retrieve the selected guest*
- *display the total bill amount*
- *display the membership status & points*
- *check membership status to determine if the guest can redeem points*
- *prompt user for the number of points to offset the total bill amount*
- *redeem points*
- *display the final total bill amount*
- *prompt user to press any key to make payment*
- *earn points*
- *while earning points, the member status will be upgraded accordingly.*
- *update check out status of the guest*

(c) Recommend a feature to be implemented (bonus mark is only awarded if advanced feature is completed)

- *You may gain up to 5 bonus marks if you propose and successfully implement an additional feature. Check with your tutor with your idea before implementing.*
- *This is team effort*

IMPORTANT INSTRUCTIONS:

- *A student is to implement feature (a) and another implements feature (b).*
- *Individual student without a team can choose to implement either one of the features.*
- *Please note that you should implement the advanced features only AFTER all the basic features have been fully implemented and working.*
- *NO MARKS will be awarded for the advanced features if the basic features have NOT been fully implemented and working.*
- *Marks will be deducted if you are not able to show your understanding of the program, both basic and advanced features (if applicable), during the presentation.*

5. ACTIVITY PLAN

Suggestions for Getting Started

There are many ways that you could complete this assignment. It is most important part to think about the entire assignment first so that it is easy to integrate the various parts.

a) Analysis

1. Understand the program specification and the requirements before attempting the assignment.
   e.g. the relationships between the classes
   the use of the attributes in each class

b) Program Design

2. Work out the User Interface needed to get the user input for suitable output.
3. Work out the main logic of the program using Object-Oriented programming techniques;

   i.e. use the inheritance and the association of the classes properly.
4. You are required to use suitable classes and methods appropriately for this assignment.

   Marks will be deducted for inefficient use of the classes/methods or improper use of classes/methods

c) Implementation & Testing

5. Determine the order in which the classes are to be implemented. (Certain classes need to be implemented before other classes can be implemented)
6. Implement the classes **ONE** at a time.
7. Test your program logic to make sure that it works.

   You must prepare test data to see that your program works correctly. All data entry should be validated and illegal data entry should be highlighted to the user so that the user can enter correct data.

6. DELIVERABLES

You are to develop the project using Console App. You should name your project as **`Snnnnnnnn_PRG2Assignment`** *(note: **`Snnnnnnnn`** is your Student Number)*.

You are required to submit your work in **TWO** stages:

Stage 1 – Due date: Wednesday, Week 15 (25 January 2023) @ 2359 hrs

- ALL the 6 classes (source files) shown in the class diagram and at least 1 basic feature to your PRG2Assignment Network folder (ictspace.ict.np.edu.sg\PRG2Assignment).

  Create a folder called Stage1 in your PRG2Assignment network folder and upload your project to this folder.

  In each of your source file, you MUST include a block comment at the top stating your **`student number`** and **`name`** as shown below:

```
//============================================================
// Student Number : S12345678
// Student Name   : Vivian Ng
//============================================================
```

  Note: If a student did not submit his/her stage 1 work by the deadline, 20 marks will be deducted.

Stage 2 – Due date: Wednesday, Week 16 (01 February 2023) @ 8.00 am

- ALL the classes (source files) that you have written for the whole assignment to your PRG2Assignment Network folder (ictspace.ict.np.edu.sg\PRG2Assignment).

  Create a folder called Stage2 in your PRG2Assignment network folder and upload your project to this folder.

  In each of your source file, you MUST include a block comment at the top stating your **`student number`** and **`name`** as shown below:

```
//============================================================
// Student Number : S12345678
// Student Name   : Vivian Ng
//============================================================
```

Presentation – During PRG2 lessons

- You are to present your application to your tutor during the PRG2 classes right after the submission deadline.

7. GRADING CRITERIA

This assignment constitutes 30% of this module.

Performance Criteria for grading the assignment is as described below. Marks awarded will be based on **program code** as well as student's degree of understanding of work done as assessed during the **presentation**.

Grading criteria for the program is given below.

A Grade

- Program implements the *Basic Features* successfully
- Program implements all the basic *input validations* successfully
- Program implements the two *Advanced Features* successfully
- Program demonstrates good design with the correct use of methods
- Program provides strong evidence of good programming practice
- Program has been tested adequately

B Grade

- Program implements the *Basic Features* successfully
- Program implements some basic *input validations* successfully
- Program implements one *Advanced Features* successfully
- Program attempts to use methods
- Program provides sufficient evidence of good programming practice
- Program has been tested adequately

C Grade

- Program implements the *Basic Features* successfully
- Program provides some evidence of good programming practice
- Program has been tested adequately

D Grade

- Program implements the *Basic Features* successfully
- Program has been tested adequately
- Score at least a 'D' in the presentation

NOTE

- *Evidence of good programming practice include appropriate use of methods, the use of meaningful variable names, proper indentation of code, appropriate and useful comments, adoption of standard naming conventions etc.*
- *Basic Input validation refers to the checking of the inputs entered by the user. e.g. invalid option, invalid date*