ASSESSMENT BRIEF

COURSE: Bachelor of Information Technology/ bachelor of Information Technology with a specialisation in Cyber Security	
Unit:	Object Oriented Design and Programming
Unit Code:	OODP101
Type of Assessment:	Assessment 3 – Solution to programming problem by group of 3-4 students
Length/Duration:	20 Hours
Unit Learning Outcomes addressed:	Upon successful completion of this unit students should be able to: 1. Demonstrate basic knowledge of object-oriented programming concepts and programming problems 2. Analyse and dissect simple design and programming problem 3. Implement a well-designed modularized solution to small programming problems 4. Develop and/or implement testing schedules
Submission Date:	Week 11 and 12
Assessment Task:	A group of 3-4 students will work together to provide a quality solution to programming problem using JAVA programming language,
Total Mark:	Assignment and Demonstration 30 Marks
Weighting:	30% of the unit total marks

Students are advised that **submission of an Assessment Task past the due date without a formally signed approved Assignment Extension Form** (Kent Website *MyKent Student Link> FORM – Assignment Extension Application Form – Student Login Required)* **or previously approved application for other extenuating circumstances impacting course of study, incurs a 5% penalty per calendar day**, calculated by deduction from the total mark.
For example. An Assessment Task marked out of 40 will incur a 2 mark penalty for each calendar day.

More information, please refer to *(*Kent Website *MyKent Student Link> POLICY – Assessment Policy & Procedures – Student Login Required)*

ASSESSMENT DESCRIPTION:

Your task will be to design and develop a programming solution to a problem using JAVA programming language.

Background

The system that you will create is a Hotel Management System (HMS). The HMS is a tool used by a hotel that manages rooms, amenities and their bookings by customer. The hotel has different types of rooms to cater the needs of different customers and has various amenities to choose from.

Each room has a unique ID, features, price and location in the hotel. For each hotel room, there are a

number of available amenities for ordering by guests. (For example: - Room no 1 has one queen bed, $200 price and at extreme left corner of ground floor and have included amenities like wifi, TV with Netflix, Laundry, Breakfast, Lunch, Dinner, Coffee machine, AC, Heater etc. Following are the number of amenities available in hotel rooms:

1. Free WIFI
2. TV with Netflix
3. Laundry containing washer and dryer
4. Air Conditioner
5. Heater
6. Coffee machine
7. Breakfast
8. Lunch
9. Dinner
10. Undercover barbeque area
11. Electric Blankets
12. Toasters
13. Microwave

This IT system, HMS, will enable the manager to customise and save different types of rooms to the system as well as view the stored list of rooms. It will be possible to inspect details for each room. The details will include specifics of the rooms as well as specific details regarding amenities that can be added to rooms. Each amenity will have an ID, name, description and basic price associated with it.

There are following room types in the hotel that you have to incorporate in your system:

1. A deluxe queen room is suitable for couples, has queen bed in it, number of amenities and 8% discount is available on its original price.
2. A deluxe spa is suitable for travellers or any needing relaxation, has one self-contained spa, one queen bed, number of amenities and surcharge of $80 is added on its original price for spa facility.
3. A family suite is suitable for family, has one queen bed and two single beds, a number of amenities and 20% discount is available on its original price.
4. Two bedroom family suite is suitable for big family, has two self-contained two bedrooms with queen beds, attached bathrooms, number of amenities and 5% discount is available on its original price.

Customer can book more than one room at a time and can add various amenities into their rooms according to their requirements. The HMS will give discount to customers according to their status. When customer book rooms, system will ask the status of customer and discounts will be given according to following information.

1. Active customers are the one who visits hotel at least once a week and will get 8% discount on every order they made.
2. VIP customers are the one who have been visiting hotel from more than one year and will get 12% discount on all orders.
3. New Customers will not get any discounts.

Once a room is booked, HMS will calculate the total of order and will issue invoice to customer.

Requirement 1: Class Diagram

This section expects you to create a class diagram after reading the coding requirements. Your class diagram should have proper relationships between classes, all possible attributes, constructors and methods that your class files will have. You need to use correct notation which is discussed in class.

Requirement 2: Coding

2.1 Create Room class which will have basic attributes as given in background information.

a) A default constructor which assigns each instance variable a default value.
b) A constructor with parameters which assign values to each instance variable. Note that the values to initialise in the objects should be passed in using arguments when the constructor is called.
c) Author get and set methods for your classes for instance variables where appropriate.
d) This class has array of amenities, so you need to add proper methods to add the items into the array, remove the items and retrieve the item from the array.
e) Room class should have calculatePrice() which will add the price of all amenities and its own price in it.
f) You need to write proper toString() method to display the details of room and amenities in it.

2.2 Create sub-classes of room class as given in background information.
You must provide the following in each particular specialised room class.

a. All the room classes have (at least) the following attributes: RoomID (int), features (String), Description (String), Price(double) which will be inherited from super class and one additional attribute of your own choice.

b. A default constructor which assigns each instance variable a default value. The String variables (e.g. features) should be initialised to “unknown”, the price should be initialised to a minimum default price and any object fields initialised to null.

c. A constructor with parameters which assign values to each instance variable. Note that the values to initialise in objects should be passed in using arguments when the constructor is called.

d. Author get and set methods for your classes for instance variables where appropriate.

e. Write a toString() method in each class that will return a String containing all the relevant data for each of your objects.

f. In each of your specialised room classes, override the setPrice() method to perform an appropriate price calculation based on the amenities in the object.

2.3 Create a class for amenities which will have attributes ID (int), name (String), Description (String), price (double). These amenities will be stored in each room in form of array or arrayList.

a. A default constructor which assigns an instance variable a default value.

b. A constructor with parameters which assign values to each instance variable.

c. Write get and set methods for your classes for instance variables where appropriate.

d. Write a toString() method in this class that will return a String containing all the relevant data for each of your objects

2.4 Create a customer class which will have name (String), discount (double) and status (String) as its attributes.

a. A default constructor which assigns an instance variable a default value. Note: You do not want customer to do any manipulations with the discount value. It will be calculated according to the status.

b. A constructor with parameters which assign values to each instance variable.

c. Write get and set methods for your classes for instance variables where appropriate.

d. Write a toString() method in each class that will return a String containing all the relevant data for each of your objects

e. setDiscount() method will calculate and set the discount according to status of customer. s

2.5 In HotelDriver class, create a main method and do as follows:

a) Create two objects of each room type.

b) Create all amenities as given in background information. You can create more of your choice.

c) Display a welcome message which should have following information in it:

i. The welcome message should have a row of "*" at the top and the bottom, just long enough to extend over the text. *Hint: Use a loop for this.*

ii. The first line of the message should read "WELCOME TO HOTEL MANAGEMENT SYSTEM"

iii. The second line of the message should be blank.

iv. The third line should read "Developed by" followed by your names and a comma, then "student ID", then your student ids of all group members.

v. The fourth line should display "OODP101 Object Oriented Design and Programming Assessment Task 4" vi. The fifth line should display the current date and time of system.

vii. The sixth line should be blank, and the seventh line should be another row of "*"

d) Ask user if they want to place order. If yes, ask the name and status of customer and save it because you will need it for calculating the final price and printing the final output.

e) Display all created room types and ask user to enter the room they want to book using room ID.

f) Once room is selected, display the list of amenities to user that they can add in their booking. Ask user to select amenities with the help of ID and user can add more than one amenity in one room.

g) Populate each room with all selected amenities. Your room class is having array or arrayList of objects and methods to add these amenities in the rooms. Use those methods here to add amenities in the room.

h) Ask user if they want to book another room, repeat the steps from a) to e).

i) When user wish to finish order, program should calculate the total of all room price, amenities price and deducting customer discount according to the status they entered.

j) Display the total amount due with all the details of order.

2.6 Document all methods in your code using comments and by following proper coding standards.

Requirement 3:- Demonstration

3.1 All group member must demonstrate their work done to their tutor during week 12 class. Your tutor can ask you any questions from the whole assessment.

ASSESSMENT SUBMISSION:

Group leader will submit java project (coding), class diagram using a word file and individual contribution statement. All these should be put inside a zip/compressed file for submission. Java project (coding) should be exported from eclipse.
All other members will submit their own individual contribution statement.

This assignment must be submitted online in Moodle.

For assistance please speak to our Academic Learning Skills Coordinators, in Sydney (als.syd@kent.edu.au) or in Melbourne (als.mel@kent.edu.au). They can help you with understanding the task, draft checking, structure, referencing and other assignment-related matters.

GENERAL NOTES FOR ASSESSMENT TASKS

Content for Assessment Task papers should incorporate a formal introduction, main points and conclusion.

Appropriate academic writing and referencing are inevitable academic skills that you must develop and demonstrate in work being presented for assessment. The content of high quality work presented by a student must be fully referenced within-text citations and a Reference List at the end. Kent strongly recommends you refer to the Academic Learning Support Workshop materials available on the Kent Learning Management System (Moodle). For details please click the link http://moodle.kent.edu.au/kentmoodle/mod/folder/view.php?id=3606 and download the file titled "Harvard Referencing Workbook". This Moodle Site is the location for Workbooks and information that are presented to Kent Students in the ALS Workshops conducted at the beginning of each Trimester.

Kent recommends a minimum of **FIVE (5)** references in work being presented for assessment. Unless otherwise specifically instructed by your Lecturer or as detailed in the Unit Outline for the specific Assessment Task, any paper with less than five (5) references may be deemed not meeting a satisfactory standard and possibly be failed.

Content in Assessment tasks that includes sources that are not properly referenced according to the "*Harvard Referencing Workbook*" will be penalised.

Marks will be deducted for failure to adhere to the word count if this is specifically stated for the Assessment Task in the Unit Outline. As a general rule there is an allowable discretionary variance to

the word count in that it is generally accepted that a student may go over or under by 10% than the stated length.
Students are not allowed to use AI tools for the preparation of their submissions.

GENERAL NOTES FOR REFERENCING

References are assessed for their quality. Students should draw on quality academic sources, such as books, chapters from edited books, journals etc. The textbook for the Unit of study can be used as a reference, but not the Lecturer Notes. The Assessor will want to see evidence that a student is capable of conducting their own research. Also, in order to help Assessors determine a student's understanding of the work they cite, all in-text references (not just direct quotes) must include the specific page number(s) if shown in the original. Before preparing your Assessment Task or own contribution, please review this 'YouTube' video (Avoiding Plagiarism through Referencing) by clicking on the following link: link: http://moodle.kent.edu.au/kentmoodle/mod/folder/view.php?id=3606

A search for peer-reviewed journal articles may also assist students. These type of journal articles can be located in the online journal databases and can be accessed from the Kent Library homepage. Wikipedia, online dictionaries and online encyclopaedias are acceptable as a starting point to gain knowledge about a topic, but should not be over-used – these should constitute no more than 10% of your total list of references/sources. Additional information and literature can be used where these are produced by legitimate sources, such as government departments, research institutes such as the National Health and Medical Research Council (NHMRC), or international organisations such as the World Health Organisation (WHO). Legitimate organisations and government departments produce peer reviewed reports and articles and are therefore very useful and mostly very current. The content of the following link explains why it is not acceptable to use non-peer reviewed websites (Why can't I just Google?): https://www.youtube.com/watch?v=N39mnu1Pkgw (Thank
you to La Trobe University for access to this video).

MARKING GUIDE (RUBRIC):

Your answers for the final examination questions will be assessed as per the following marking criteria. Please read carefully each section/level and marks weightage.

Criteria	High Distinction	Distinction	Credit	Pass	Fail
			Requirement 1- Design		
Class Diagram (5 Marks)	***Excellent(5 Marks)*** Class diagram have all classes, proper relationships between classes, all possible attributes, constructors and methods that java files are having. Correct notation has been used.	***Very Good (4 Marks)*** Class diagram have all classes but NO proper relationships between classes, all possible attributes, constructors and methods that java files are having. Correct notation has been used	***Good(3 Marks)*** Class diagram have all classes, proper relationships between classes BUT missing all possible attributes, constructors and methods that java files are having. Correct notation has been used	***Satisfactory(2 Marks)*** Class diagram have few classes, NO proper relationships between classes, only few attributes, constructors and methods that java files are having. Issues with notation	***Not Satisfactory (0 Marks)*** Not attempted

Requirement 2-Coding					
2.1 Room Class (2.5 Marks)	***Excellent (2.5 Marks)***Room Class has: - All mentioned attributes - Default constructor and Parametrized constructor - Getters and Setters - toString() - Method to add, retrieve and remove the elements of array - calculatePrice() which adding the price of all amenties in it	***Very Good (2 Marks)*** Room Class is missing or having issues in ONE of the following: - All mentioned attributes - Default constructor and Parametrized constructor - Getters and Setters - toString() - Method to add, retrieve and remove the elements of array - calculatePrice() which adding the price of all amenities in it	***Good (1.5 Marks)*** Room Class is missing or having issues in TWO of the following: - All mentioned attributes - Default constructor and Parametrized constructor - Getters and Setters - toString() - Method to add, retrieve and remove the elements of array calculatePrice() which adding the price of all amenities in it	***Satisfactory (1 Mark)*** Room Class is missing or having issues in MORE THAN TWO of the following: - All mentioned attributes - Default constructor and Parametrized constructor - Getters and Setters - toString() - Method to add, retrieve and remove the elements of array calculatePrice() which adding the price of all amenities in it	***Not Satisfactory (0 Marks)*** Not attempted

2.2 SubClasses of Room Class (8 Marks)	***Excellent (8 Marks)*** All sub classes have: - All mentioned attributes - Default constructor and Parametrized constructor - Getters and Setters including overridden calculatePrice() according to condition	***Very Good (6.5 Marks)*** One or two sub classes are missing or having issues in one or two of the following: - All mentioned attributes - Default constructor and Parametrized constructor - Getters and Setters including overridden calculatePrice() according to condition	***Good(4.5 Marks)*** Three sub classes are missing or having issues in one or two of the following: - All mentioned attributes - Default constructor and Parametrized constructor - Getters and Setters including overridden calculatePrice() according to condition	***Satisfactory (2.5 Marks)*** All sub classes are missing or having issues in more than two of the following: - All mentioned attributes - Default constructor and Parametrized constructor - Getters and Setters including overridden calculatePrice() according to condition	***Not Satisfactory (0 Marks)*** Not attempted
2.3 Amenities Class (2.5 Marks)	***Excellent (2.5 Marks)*** Amenities Class has: - All mentioned attributes - Default constructor and Parametrized constructor - Getters and Setters - toString()	***Very Good (2 Marks)*** Amenities Class missing or having issues in ONE of the following: - All mentioned attributes - Default constructor and Parametrized constructor - Getters and Setters - toString()	***Good (1.5 Marks)*** Amenities Class missing or having issues in TWO of the following: - All mentioned attributes - Default constructor and Parametrized constructor - Getters and Setters - toString()	***Satisfactory (1 Mark)*** Amenities Class is missing or having issues in MORE THAN TWO of the following: - All mentioned attributes - Default constructor and Parametrized constructor - Getters and Setters - toString()	***Not Satisfactory (0 Marks)*** Not attempted

2.4 Customer Class (2 Marks)	***Excellent (2 Marks)*** Customer Class has: - All mentioned attributes - Default constructor and Parametrized constructor - Getters and Setters - toString()	***Very Good (1.5 Marks)*** Customer Class is missing or having issues in ONE of the following: - All mentioned attributes - Default constructor and Parametrized constructor - Getters and Setters - toString()	***Good (1 Marks)*** Customer Class is missing or having issues in TWO of the following: - All mentioned attributes - Default constructor and Parametrized constructor - Getters and Setters - toString()	***Satisfactory (0.5 Marks)*** Customer Class is missing or having issues in MORE THAN TWO of the following: - All mentioned attributes - Default constructor and Parametrized constructor - Getters and Setters - toString()	***Not Satisfactory (0 Marks)*** Not attempted
2.5 Hotel Driver Driver Class (6 Marks)	***Excellent (6 Marks)*** Hotel Driver Class consist of: - Creation of rooms as per requirements - Creation of all amenities as per requirements - Population of each room with amenities - Welcome message and option to ask for booking rooms -Complete order and proper invoice as per requirements	***Very Good (5 Marks)*** - Minor issues in creation of rooms and amenities or in booking rooms	***Good (4 Marks)*** - Creation of rooms and amenities is done as per requirements but population have minor issues which is having impact on booking rooms	***Satisfactory (2 Marks)*** Creation of rooms and amenities and its popuation is done as per requirements but order cannot be completed or giving wrong invoice to user due to logic error in code	***Not Satisfactory (0 Marks)*** Not attempted
Demonstration and group work (4 marks)	**Excellent (4 marks)** Answered all questions and student is well known of assessment requirements and group work. Group cohesion is evident in demonstration	***Very Good (3 Marks)*** Answered few questions only but contributed well to group work	***Good (3 Marks)*** Answered all questions in demonstration but contribution in group work is somewhere missing	**Satisfactory (1 Mark)** Just attended the demonstration but could not answer any questions clearly	***Not Satisfactory (0 Mark)*** Not attempted

Total Marks ***/30***					