INTI International University & Colleges™

UNIVERSITY OF WOLLONGONG AUSTRALIA

FACULTY OF ENGINEERING AND INFORMATION SCIENCES

SUBJECT'S INFORMATION:			
Subject:	CSIT121 Object-Oriented Design and Programming		
Session:	Autumn 2019 (February)		
Programme / Section:	Computer Science and IT		
Lecturer:	Ms. Siti Hawa		
Coursework Type *(tick appropriate box)*	✓ Individual Assignment ☐ Group Assignment ☐ Project ☐ Lab Task ☐ Seminar / Tutorial Paper ☐ Others		
Coursework Title:	**Assignment 2**	Coursework Percentage:	10%

ASSESSMENT CRITERIA:		
Correctness	The application should produce the correct result as stated in the specification.	2 marks
Class design and implementation	Design of class that follows the specification provided. Class relationship is implemented. Use of class variables for shared attributes.	4 marks
Main Application	Correct implementation of the main method with appropriate control flow and data structures used.	3 marks
Readability and Output	Appropriate comments are included. Meaningful identifiers used. Proper indentation and line spacing used. Well formatted output and creativity shown.	1 mark

SUBMISSION:

All completed work should be submitted online through Moodle before or on the due date provided.

SUBMIT AS EARLY AS POSSIBLE. ONLY ONE SUBMISSION IS ALLOWED. IF RE-SUBMISSION IS NECESSARY, YOU ARE REQUIRED TO REMOVE THE EARLIER SUBMISSION AND THIS MUST BE DONE BEFORE THE DUE DATE. OTHERWISE YOU WILL BE PENALIZED FOR LATE SUBMISSON.

DUE DATE:	**Monday, 20th May 2019 (11:55 pm)**

PENALTIES FOR LATE SUBMISSION:

Penalties apply to all late work, except if student academic consideration has been granted. Late submissions will attract a penalty of 25% of the assessment mark per day including the weekend. Work more than (3) days late will be awarded a mark of zero.

PLAGIARISM:

When you submit an assessment task, you are declaring the following

1. It is your own work and you did not collaborate with or copy from others.
2. You have read and understand your responsibilities under the University of Wollongong's policy on plagiarism.
3. You have not plagiarised from published work (including the internet). Where you have used the work from others, you have referenced it in the text and provided a reference list at the end ot the assignment.

Plagiarism will not be tolerated. Students are responsible for submitting original work for assessment, without plagiarising or cheating, abiding by the University's policies on Plagiarism as set out in the University Handbook under University Policy Directory and in Faculty handbooks and subject guides. under University Policy Directory and in Faculty handbooks and subject guides.

COURSEWORK SPECIFICATION

OBJECTIVES

This assignment aims to provide you with some experience in writing codes using Java programming language that covers the following topics:

- Inheritance and Polymorphism
- Object Interactions

Remember that:

1. **All programs should be able to run on the lab's computers.**
2. **You must put the following information on the header of each text and source file you will be submitting in this assignment:**
   Student's full name:
   Student's ID:
   Modification Date:
   Purpose of this file (or program):
3. **Assignments that are not able to be compiled will result in zero mark given to the assignment.**
4. **You must only use the Java features that have already been covered in the lectures**

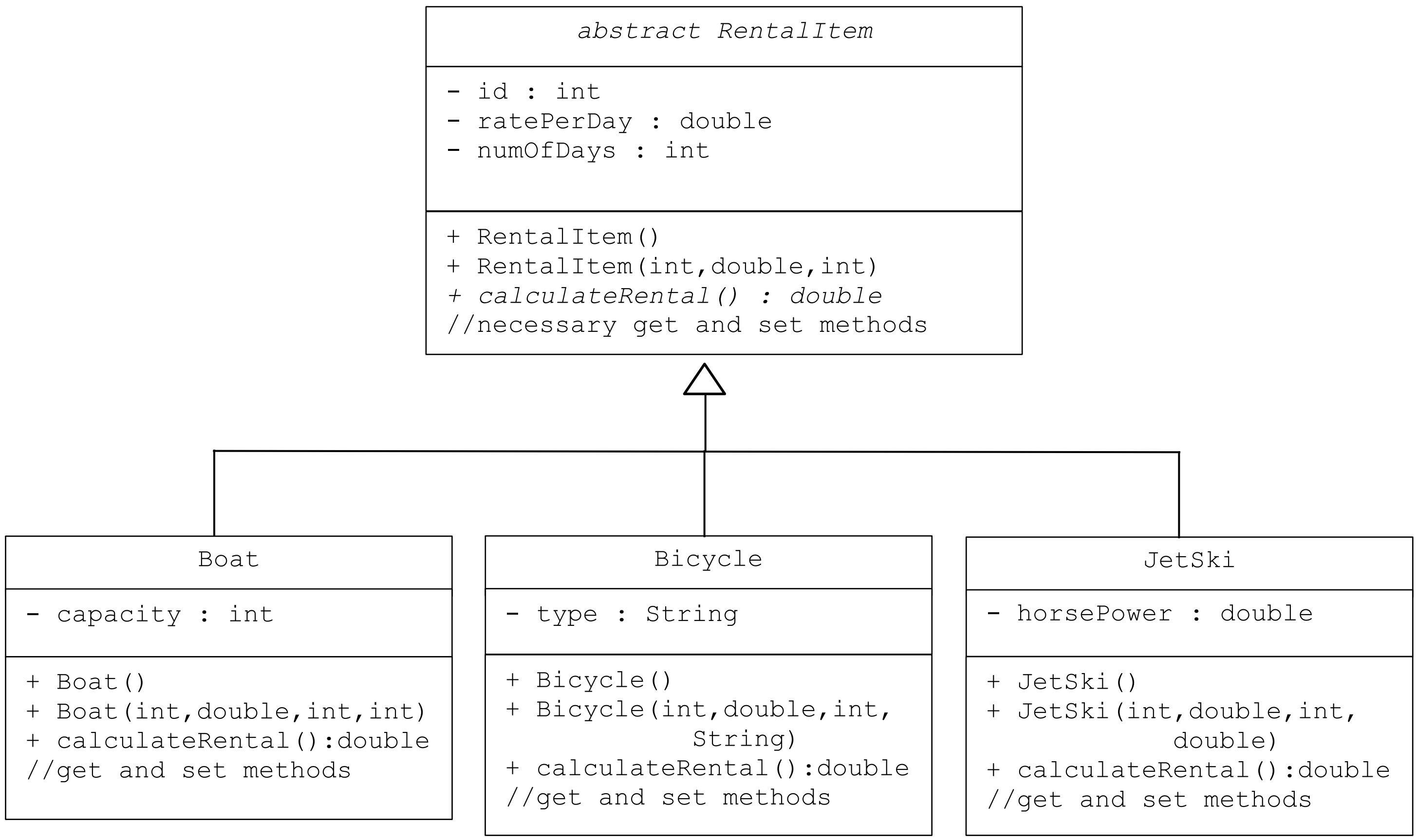
Question:

A resort near Pangkor Island offers rental items to their customers at a price. The customer may rent a number of rental items which can include a boat, a bicycle, or a jet ski. You are required to write a Java application that keeps track of the items rented by each customer and also able to calculate the total rental amount due.

To implement this, you need to declare several classes. The classes includes the `RentalItem` class which is inherited by the `Boat` class, the `Bicycle` class, and the `JetSki` class. A `Customer` class is also required to keep track of the customers renting the items. Lastly, you also need another class called `Rental` that will relate the `Customer` object with a number of `RentalItem` object.

The classes are partially described in the following diagrams:

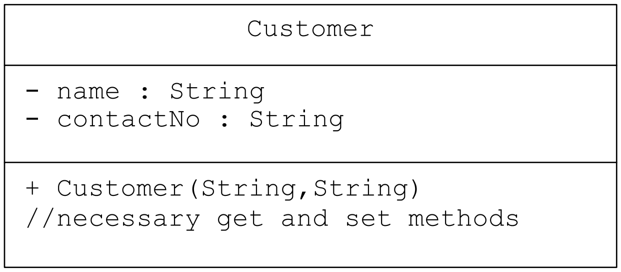
The `RentalItem` class and its subclasses.


abstract RentalItem
- id : int
- ratePerDay : double
- numOfDays : int
+ RentalItem()
+ RentalItem(int,double,int)
+ calculateRental() : double
//necessary get and set methods
Boat
- capacity : int
+ Boat()
+ Boat(int,double,int,int)
+ calculateRental():double
//get and set methods
Bicycle
- type : String
+ Bicycle()
+ Bicycle(int,double,int, String)
+ calculateRental():double
//get and set methods
JetSki
- horsePower : double
+ JetSki()
+ JetSki(int,double,int, double)
+ calculateRental():double
//get and set methods

Include the necessary accessors and mutators to the classes. The calculateRental() method should perform different calculation to get the rental amount for each object. The calculation is as follows:

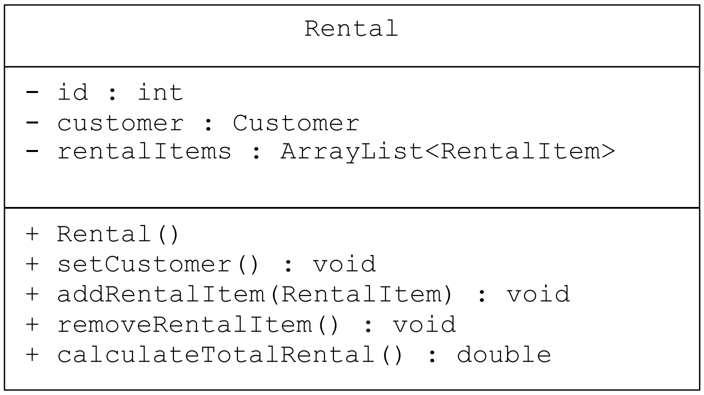
Boat	Rental is calculated based on capacity. A boat for 10 passengers and more is charged with the rate per day plus additional charges of RM50 and a boat for less than 10 passengers has no additional charges.
Bicycle	A mountain bike has additional charges of RM10 than the normal rate per day. A kids bike is given half the price of the normal rate per day. Other bikes are charged at the normal rate.
JetSki	A jet ski with 250 horse power or below has no additional charge. Others will be charged at 1.5 of the normal rate.

The Customer class.



```
Customer
- name : String
- contactNo : String
+ Customer(String,String)
//necessary get and set methods
```

The Rental Class



```
Rental
- id : int
- customer : Customer
- rentalItems : ArrayList<RentalItem>
+ Rental()
+ setCustomer() : void
+ addRentalItem(RentalItem) : void
+ removeRentalItem() : void
+ calculateTotalRental() : double
```

The Rental class will include the object of class Customer and an ArrayList of RentalItem objects.

The rentalItems ArrayList should be declared as RentalItem type so that it can store any number of objects from different subclasses of RentalItem class (either Boat, Bicycle, or JetSki). Each customer can rent as many of these items as they wish.

The addRentalItem(RentalItem) method should receive a RentalItem object and add it to the ArrayList. The calculateTotalRental() method should add the rental amount from each rental items and return the value.

Lastly, write a main() method that keeps several Rental objects. Use suitable menu driven application to allow the user to add customer, add rental items, remove rental items, display all rental details and total rental charges, and display a single rental details and its total charges.

You are required to ensure that all errors are handled properly. Use exception handling technique and/or condition validation as necessary. Include meaningful comments in your programs and make your programs readable. Be creative in displaying your outputs.