**Java 1: Assignment #3/ Test #3**

Due: TBD

Please remember to read over the assignment rubric before answering these questions. Each question should be written in its own file for upload. When all questions are answered place the source code for each question in a folder, zip it up, then upload it to the assignment 3/Test 3 upload area on the LMS. If there are any concerns, let me know right away. Feel free to work with others to solve the questions but please code them on your own.

**Question 1: [40 marks]**

**(Half the marks of this question goes towards Assignment 3 and the other half of the marks goes toward Test 3)**

**Please note the following question does not require inheritance. This question just uses objects normally.**

**Implement a class called Customer** that represents a passenger on the airline Crow Airlines. Each Customer instance should keep track of its name, its address, its membership number in Crow Airlines’ frequent flyer club, and the number of points accumulated in that club. Thus, a Customer object has four fields or instance variable:

name of type String address of type String membershipNumber of type int membershipPoints of type int

**Implement a class called Flight** that represents a flight on Crow Airlines. Each Flight instance should keep track of its flight number, its origin, its destination, and its departure date and time. Thus, a Flight instance has four fields (instance variables):

flight number of type int origin of type String destination of type String departure of type String

**Implement a class called AirlineTicket** that represents a ticket for a flight on Crow Airlines. Each AirlineTicket instance should keep track of its passenger, its flight, and its price. Thus, an AirlineTicket instance has three fields (instance variables):

passenger of type Customer flight of type Flight price of type double

**For the Customer class**, implement the following instance methods:

* public get (accessing) methods and protected set (modifying) methods for the fields (instance variables)
* a zero-argument constructor (i.e., one that takes no parameters)
* a constructor that accepts a name, address, and membership number, and sets the initial value of the membership points to zero
* a toString method that returns a String representing the Customer objects name, address, membership number, and membership points.
* e.g., "Ming Chiang, 147 Hobson Road, Ottawa, #123456, 1500 points"
* an applyPoints method that takes an AirlineTicket as a parameter, calculates the points value of the ticket, and adds that value to the Customer objectís membership points. The applyPoints method should return the new value of the membership points. The points for a given AirlineTicket object are to be calculated by dividing the price of the AirlineTicket object by 100 and rounding to the nearest integer.

**For the AirlineTicket class**, implement the following instance methods:

* public get (accessing) methods and protected set (modifying) methods for the fields (instance variables)
* a zero-argument constructor (i.e., one that takes no parameters)
* a constructor that accepts a passenger, a flight, and a price
* a toString method that returns a String representing the AirlineTicket object. The String should include the passengers name, the flight number, the origin, the destination, the departure, and the price. e.g., "Ming Chiang, Flight 1030, Ottawa to Calgary, 03/02/99 7:50 pm, $600.00"

**For the Flight class**, implement the following instance methods:

* public get (accessing) methods and protected set (modifying) methods for the fields (instance variables)
* a zero-argument constructor (i.e., one that takes no parameters)
* a constructor that accepts flight number, origin, destination, and a departure
* a toString method that returns a String representing the Flight object. The String should include the flight number, the origin, the destination, and the departure. e.g., "Flight 1030, Ottawa to Calgary, 03/02/99 7:50 pm"

**Testing:**

Be sure to test your code thoroughly. Create a class called TestTicket with several static test methods for testing your Customer, AirlineTicket, and Flight objects. Choose meaningful test cases. Here is an example of a test case.

public static void test1() {

Customer anisa;

AirlineTicket ticketAnisa;

Flight flightCalgary;

anisa = new Customer(); anisa.setName("Anisa Shane");

anisa.setAddress("101 Shangrala Street, Calgary");

anisa.setMembershipNumber(123456); anisa.setMembershipPoints(0);

flightCalgary = new Flight(101, "Ottawa", "Calgary", "03/02/13 7:50 pm");

ticketAnisa = new AirlineTicket(anisa, flightCalgary, 600.00);

System.out.println(anisa);

System.out.println(flightCalgary);

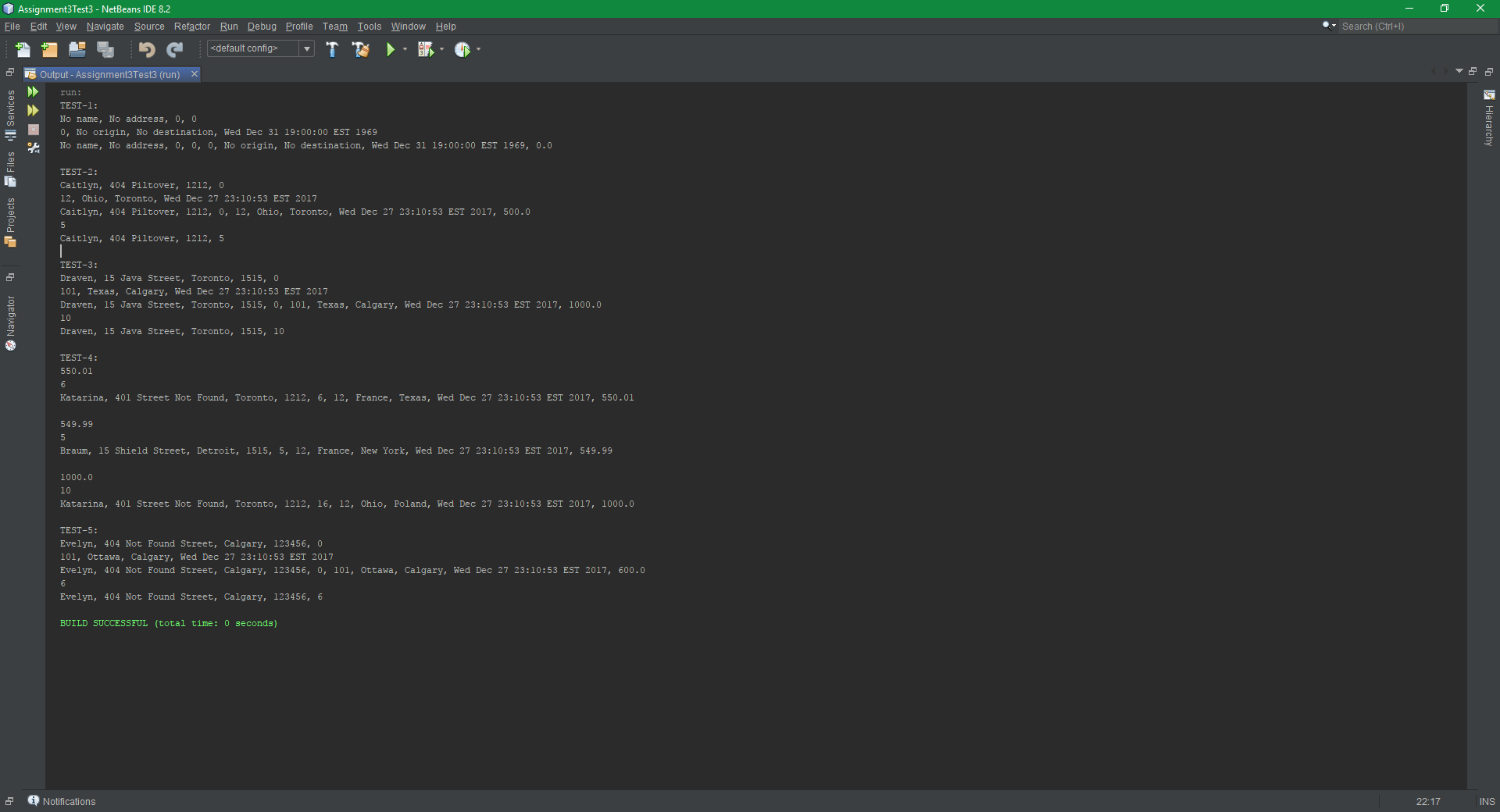
System.out.println(ticketAnisa);

System.out.println(anisa.applyPoints(ticketAnisa));

System.out.println(anisa);

}

**Submission:** Submit a printout of your Customer, Airline Ticket, and Flight classes and their methods along with a copy of the output generated by the testing. Include comments in your code and ensure that all of the output is clearly explained. Be sure to focus on encapsulation. Make everything private that it is reasonable to do so. Testing will be marked! Here is an example of the programs output.



**Question 2: [10 marks]**

**(Half the marks of this question goes towards Assignment 3 and the other half of the marks goes toward Test 3)**

Write a program that meets that following requirements:

* Creates an array with 100 randomly chosen integers.
* Prompts the user to enter the index of the array, and then displays the corresponding element value. If the specified index is out of bounds, display the message Index Out of Bounds!
* If the input is not of the valid type display the message Not a valid index! Please enter an integer!
* Here is an example of the program’s output.

