

AP Computer Science Lab

Airport and Flight Lab

This question involves a system to manage flights at an airport. Flights are represented by the following Flight class.

```
public class Flight {
    // There may be instance variables, constructors, and methods
    that are not shown.

    /** Returns the number of passengers on the flight */
    public int getNumPassengers()
    { /* implementation not shown */ }

    /** Returns the price of a seat on the flight. All seats on a
    flight have the same price.
    *   Postcondition: The value returned is greater than or equal
    to 0.0.
    */
    public double getPrice()
    { /* implementation not shown */ }

    /** Returns the capacity of the flight (the maximum number of
    passengers the flight can carry)
    */
    public int getCapacity()
    { /* implementation not shown */ }
}
```

The flights into and out of an airport are stored in an Airport object, which contains a list of all such flights. You will write two methods of the Airport class.

```
public class Airport {
    // There may be instance variables, constructors, and methods
    that are not shown.

    /** A list of the flights into and out of this airport
    *   Guaranteed not to be null and to contain only non-null
    entries
    */

    private ArrayList<Flight> allFlights;
```

```

    /** Returns the revenue generated by all flights at the airport,
    as described in part (a) */
    public double getTotalRevenue()
    { /* to be implemented in part (a) */ }

    /** Updates the list of flights by removing certain flights and
    returns the total number of
    * passengers whose flights were removed, as described in part
    (b)
    */

    public int updateFlights()
    { /* to be implemented in part (b) */ }
}

```

(a) Write the Airport method `getTotalRevenue`. The method returns the total revenue for all flights into and out of the airport. Revenue for a flight is the product of the number of passengers on the flight and the price of a seat on the flight. All seats on a flight have the same price.

Some flights sell more seats than there is capacity for, since passengers sometimes cancel or modify reservations. If there are more passengers on a flight than the flight has capacity for, the revenue for the flight is the product of the capacity and the price of a seat on the flight.

For example, assume that `capitalHub` has been declared as an Airport object and `ArrayList allFlights` contains the following flights.

Number of Passengers: 25 Price: 50.00 Capacity: 30	Number of Passengers: 10 Price: 100.50 Capacity: 60	Number of Passengers: 50 Price: 200.00 Capacity: 40	Number of Passengers: 20 Price: 100.00 Capacity: 120
--	---	---	--

The following table shows the revenue that would be generated by each flight in `allFlights`.

Number of Passengers	Price of a Seat	Capacity	Revenue Calculation
25	\$50.00	30	$25 \times \$50.00 = \$1,250.00$
10	\$100.50	60	$10 \times \$100.50 = \$1,005.00$
50	\$200.00	40	$40 \times \$200.00 = \$8,000.00$
20	\$100.00	120	$20 \times \$100.00 = \$2,000.00$

The call `capitalHub.getTotalRevenue()` should return the value **12255.0**.

Complete method `getTotalRevenue`.

```
/** Returns the revenue generated by all flights at the airport, as
described in part (a) */
public double getTotalRevenue()
```

(b) Write the Airport method `updateFlights`. The method removes from the `ArrayList` `allFlights` any flight where the number of passengers is less than 20 percent of the total capacity. The method should return the total number of passengers whose flight was removed.

For example, assume that `capitalHub` has been declared as an `Airport` object and `ArrayList` `allFlights` contains the following flights.

Number of Passengers: 25 Price: 50.00 Capacity: 30	Number of Passengers: 10 Price: 100.50 Capacity: 60	Number of Passengers: 50 Price: 200.00 Capacity: 40	Number of Passengers: 20 Price: 100.00 Capacity: 120
--	---	---	--

The call `capitalHub.updateFlights()` should return the value 30, and after the method finished executing, the `ArrayList` `allFlights` should contain the following two flights.

Number of Passengers: 25 Price: 50.00 Capacity: 30	Number of Passengers: 50 Price: 200.00 Capacity: 40
--	---

Complete method `updateFlights`.

```
/** Updates the list of flights by removing certain flights and returns the total number of
 * passengers whose flights were removed, as described in part (b)
 */
public int updateFlights()
```