

CS 1073

FR04A

Assignment 4

Ebrahim Arefi

3621326

I. A Stateroom class:

```
/**
 * This class represents a stateroom on the cruise ship.
 *
 * @author Ebrahim Arefi, 3621326
 */
public class Stateroom {

    /**
     * The section name of stateroom.
     */
    private String sectionName;

    /**
     * The room number of the stateroom.
     */
    private int roomNum;

    /**
     * The price per week for the stateroom.
     */
    private double pricePerWeek;

    /**
     * This method constructs a Stateroom object with the specified
     * section name, room number, and weekly price.
     *
     * @param sectionNameIn the section name of the stateroom.
     * @param roomNumIn     the room number of the stateroom.
     * @param pricePerWeekIn the price per week of the stateroom.
     */
    public Stateroom(String sectionNameIn, int roomNumIn, double pricePerWeekIn) {
        sectionName = sectionNameIn;
        roomNum = roomNumIn;
        pricePerWeek = pricePerWeekIn;
    }

    /**
     * This method retrieves the section name of the stateroom.
     *
     * @return the section name.
     */
    public String getName() {
        return sectionName;
    }
}
```

```
}

/**
 * This method retrieves the room number of the stateroom.
 *
 * @return the room number.
 */
public int getRoomNum() {
    return roomNum;
}

/**
 * This method retrieves the weekly price of the stateroom.
 *
 * @return the weekly price.
 */
public double getWeeklyPrice() {
    return pricePerWeek;
}
}
```

II. A Cruiser class:

```
/**
 * This class shows the information of a cruiser.
 * Each cruiser has a name, activity fees, shore excursion fees,
 * a discount, and a stateroom.
 *
 * @author Ebrahim Arefi, 3621326
 */
public class Cruiser {

    /**
     * The name of the cruiser.
     */
    private String name;

    /**
     * The shore excursion fee for the cruiser.
     */
    private double shoreExcursionFee;

    /**
     * The activity fee for the cruiser.
     */
    private double activityFee;

    /**
     * The discount for the cruiser.
     */
    private double discount;

    /**
     * The stateroom where the cruiser is staying.
     */
    private Stateroom stateroom;

    /**
     * This method constructs a Cruiser object with the specified
     * name, shore excursion fee, activity fee, discount, and stateroom.
     *
     * @param name          the name of the cruiser.
     * @param shoreExcursionFee the fee for shore excursions.
     * @param activityFee    the fee for activities.
     * @param discount       the discount applied.
     * @param stateroom      the stateroom assigned.
     */
}
```

```

    */
    public Cruiser(String name, double shoreExcursionFee, double activityFee,
double discount, Stateroom stateroom) {

        this.name = name;
        this.shoreExcursionFee = shoreExcursionFee;
        this.activityFee = activityFee;
        this.discount = discount;
        this.stateroom = stateroom;
    }

    /**
     * mutator methods:
     * Method is called to change the name of the cruiser.
     *
     * @param name the new name.
     */
    public void setName(String name) {
        this.name = name;
    }

    /**
     * Mutator methods:
     * Method is called to set the shore excursion fee.
     *
     * @param shoreExcursionFee the new shore excursion fee.
     */
    public void setShoreExcursionFee(double shoreExcursionFee) {
        this.shoreExcursionFee = shoreExcursionFee;
    }

    /**
     * mutator methods:
     * Method is called to set the activity fee.
     *
     * @param activityFee the new activity fee.
     */
    public void setActivityFee(double activityFee) {
        this.activityFee = activityFee;
    }

    /**
     * mutator methods:
     * Method is set to change the discount amount.
     *
     * @param discount the new discount.

```

```

    */
    public void setDiscount(double discount) {
        this.discount = discount;
    }

    /**
     * mutator methods:
     * Method is called to set/change the stateroom.
     *
     * @param stateroom the new stateroom.
     */
    public void setStateroom(Stateroom stateroom) {
        this.stateroom = stateroom;
    }

    /**
     * mutator methods:
     * Method calculates the total amount the cruiser
     * must pay for the cruise.
     *
     * @return the total payment owing.
     */
    public double getTotalPaymentOwing() {
        return shoreExcursionFee + activityFee + discount;
    }

    /**
     * This method retrieves and returns the cruiser informations.
     *
     * @return a string with the cruiser details.
     */
    public String getCruiserDetails() {

        return name + "\n\t- " + stateroom.getName() + "\n\t- " +
stateroom.getRoomNum();
    }

}

```

III. A driver class:

```
/**
 * This is a driver class to test the Stateroom and Cruiser classes.
 * It creates objects and prints the details.
 *
 * @author Ebrahim Arefi, 3621326
 */
public class As4TestDriver {
    public static void main(String[] args) {

        Stateroom sea = new Stateroom("Sea Breeze", 248, 1265.95);
        Cruiser tamara = new Cruiser("Tamara Glass", 456.50, 274.70, 85.00, sea);

        Stateroom odyssey = new Stateroom("Odyssey", 231, 1400.75);
        Cruiser ohas = new Cruiser("Ohas Natarajan", 325.50, 245.95, 0, odyssey);

        Stateroom nautilus = new Stateroom("Nautilus", 173, 1374.50);
        Cruiser vinson = new Cruiser("Vinson Tadeja", 395.50, 115.80, 105.00,
nautilus);

        tamara.setActivityFee(304.55);

        Stateroom zephyr = new Stateroom("Zephyr", 936, 1630.95);
        ohas.setStateroom(zephyr);

        vinson.setShoreExcursionFee(489.75);

        tamara.setDiscount(105.00);

        System.out.println();
        System.out.println("-----");
        System.out.println(tamara.getCruiserDetails() + "\n\tTotal amount owing: "
+ tamara.getTotalPaymentOwing());
        System.out.println("-----");
        System.out.println(ohas.getCruiserDetails() + "\n\tTotal amount owing: "
+ ohas.getTotalPaymentOwing());
        System.out.println("-----");
        System.out.println(vinson.getCruiserDetails() + "\n\tTotal amount owing: "
+ vinson.getTotalPaymentOwing());
        System.out.println("-----");
        System.out.println();
    }
}
```

As4Output:

```
ebi@Ebis-MBP AS4 % Javac As4TestDriver.java
ebi@Ebis-MBP AS4 % Java As4TestDriver
```

Tamara Glass

- Sea Breeze
- 248

Total amount owing: 866.05

Ohas Natarajan

- Zephyr
- 936

Total amount owing: 571.45

Vinson Tadeja

- Nautilus
- 173

Total amount owing: 710.55

ebi@Ebis-MBP AS4 %