OOP with Encapsulation Exercises

Class Activity - Deck of Cards

We will be working on building a program together that allows us to model and simulate the behaviors for a deck of cards.

Individual Exercises

The following are descriptions of several everyday items that you need to implement as classes. The minimal set of attributes and methods are outlined, but you should feel free to add additional ones.

Homework Assignment

Data Members

Attribute	Data Type	Get	Set	Description
totalMarks	int	X	X	The total number of correct marks received on the assignment.
possibleMarks	int	X		The number of possible marks on the assignment.
submitterName	string	X	X	The submitter's name for the assignment.
letterGrade *(derived)*	string	X		The letter grade for the assignment.

Notes

- letterGrade is a derived attribute that is calculated using totalMarks and possibleMarks.
    - For 90% or greater return "A"
    - 80–89% return "B"
    - 70–79% return "C"
    - 60–69% return "D"
    - otherwise return "F"

Constructor

The HomeworkAssignment class has a single constructor. It accepts a single argument possibleMarks.

public HomeworkAssignment(int possibleMarks)

Fruit Tree

Data Members

Attribute	Data Type	Get	Set	Description
typeOfFruit	string	X		The type of fruit on the tree.
piecesOfFruitLeft	int	X		The number of remaining fruit pieces on the tree.

Methods

```
public bool pickFruit(int numberOfPiecesToRemove)
```

Notes

- pickFruit() is a method

- returns true if more fruit was picked or false if no fruit was left to be picked.
- When picking fruit, update the number of remaining pieces by how many were removed

Constructor

The FruitTree class has a single constructor. It accepts two arguments typeOfFruit and startingPiecesOfFruit.

public FruitTree(string typeOfFruit, int startingPiecesOfFruit)

Employee

Data Members

Attribute	Data Type	Get	Set	Description
employeeId	int	X		The employee id.
firstName	string	X		The employee's first name.
lastName	string	X	X	The employee's last name.
fullName *(derived)*	string	X		The employee's full name.
department	string	X	X	The employee's department.
annualSalary	double	X		The employee's annual salary.

Notes

- fullName is a derived attribute that returns lastName, firstName.

Methods

```
public void raiseSalary(double percent)
```

Notes

- raiseSalary(double percent) increases the current annual salary by the percentage provided

Constructor

The Employee class has a single constructor. It accepts four arguments.

public Employee(int employeeId, String firstName, String lastName, double salary)

Airplane

Data Members

Attribute	Data Type	Get	Set	Description
planeNumber	string	X		The six-character plane number.
bookedFirstClassSeats	int	X		The number of already booked first class seats
availableFirstClassSeats *(derived)*	int	X		The number of available first class seats.

totalFirstClassSeats	int	X	The total number of first class seats.
bookedCoachSeats	int	X	The number of already booked first class seats
availableCoachSeats *(derived)*	int	X	The number of available first class seats.
totalCoachSeats	int	X	The total number of first class seats.

Notes

- availableFirstClassSeats is a derived attribute calculated by subtracting bookedFirstClassSeats from totalFirstClassSeats
- availableCoachSeats is a derived attribute calculated by subtracting bookedCoachSeats from totalCoachSeats

Constructors

The Airplane class has a single constructor. It accepts three arguments.

```
Airplane(String planeNumber, int
totalFirstClassSeats, int totalCoachSeats)
```

- planeNumber is the six-character plane number
- totalFirstClassSeats is the initial number of total first class seats.
- totalCoachSeats is the initial number of total coach seats.

Methods

```
bool reserveSeats(bool forFirstClass, int
totalNumberOfSeats)
```

Notes

- reserveSeats() is a method
  - if firstClass is true, reserve the value for totalNumberOfSeats for first class
  - if firstClass is false, reserve the value for totalNumberOfSeats for coach
  - return true if the reservation can be made, false if it cannot

Television

Data Members

Attribute	Data Type	Get	Set	Description
isOn	boolean	X		Whether or not the TV is turned on.
currentChannel	int	X		The value for the current channel. Channel levels go between 3 and 18.
currentVolume	int	X		The current volume level.

Constructors

The Television class does not need a constructor. It can use the **default constructor**.

A new TV is off by default. The channel is set to 3 and the volume level

to 2.

Methods

```
void turnOff()
void turnOn()
void changeChannel(int newChannel)
void channelUp()
void channelDown()
void raiseVolume()
void lowerVolume()
```

Notes

- turnOff() turns off the tv
- turnOn() besides turning the tv on, also resets the channel to 3 and the volume level to 2
- changeChannel(int newChannel) changes the current channel (only if it is on) to the value of newChannel as long as it is between 3 and 18
- channelUp() increases the current channel by 1 (only if it is on). If the value goes past 18, then the current channel should be set to 3.
- channelDown() decreases the current channel by 1 (only if it is on). If the value goes below 3, then the current channel should be set to 18.
- raiseVolume() increases the volume by 1 (only if it is on). The limit is 10
- lowerVolume() decreases the volume by 1 (only if it is on). The limit is 0

Elevator

Data Members

Attribute	Data Type	Get	Set	Description
currentFloor	int	X		The current floor that the elevator is on.
numberOfFloors	int	X		The number of floors available to the elevator.
doorOpen	boolean	X		Whether the elevator door is open or not.

Constructor

The Elevator class has a single constructor that takes one argument. New elevators start on floor 1.

```
Elevator(int totalNumberOffloors)
```

- totalNumberOfFloors indicates how many floors are available to the elevator

Methods

```
void openDoor()
void closeDoor()
void goUp(int desiredFloor)
void goDown(int desiredFloor)
```

Notes

- openDoor() opens the elevator door.
- closeDoor() closes the elevator door.

- goUp(int desiredFloor) sends the elevator upward to the desired floor as long as the door is not open. Cannot go past last floor.
- goDown(int desiredFloor) sends the elevator downward to the desired floor as long as the door is not open. Cannot go past floor 1.