OOP with Encapsulation 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.

Notes

- X in the set column indicates it **should have a set accessor**.
- Nothing in the set column indicates the property is a derived property.
- Readonly properties do not require a set accessor.
- Private setters require the property to have private set.

Homework Assignment

Properties

Property	Data Type	Get	Set	Description
TotalMarks	int	X	X	Gets or sets the total number of correct marks received on the assignment.
PossibleMarks	int	X	readonly	Gets the number of possible marks on the assignment.
SubmitterName	string	X	readonly	Gets or sets the submitters name for the assignment.
LetterGrade	string	X		Gets the letter grade for the assignment.

Notes

- PossibleMarks and SubmitterName are readonly properties that can only be set from the constructor.
- LetterGrade is a derived property 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"
    - *hint*: PossibleMarks and TotalMarks are integers, what happens when a smaller integer is divided by a larger integer

Constructor

The HomeworkAssignment class has a single constructor. It accepts two arguments possibleMarks and submitterName.

public HomeworkAssignment(int possibleMarks, string submitterName)

Fruit Tree

Properties

Property	Data Type	Get	Set	Description
TypeOfFruit	string	X	readonly	Gets the type of fruit on the tree.
PiecesOfFruitLeft	int	X	private	Gets the number of remaining fruit pieces on the tree.

Notes

Methods

```
public bool PickFruit(int numberOfPiecesToRemove)
```

Notes

- PickFruit() is a method
  - returns true if more fruit was picked or false if PiecesOfFruitLeft is 0.
  - When picking fruit, update the PiecesOfFruitLeft by removing 1.

Constructor

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

public FruitTree(string typeOfFruit, int startingPiecesOfFruit)

Employee

Properties

Property	Data Type	Get	Set	Description
EmployeeId	int	X	readonly	Gets the employee id.
FirstName	string	X	readonly	Gets the employee's first name.
LastName	string	X	X	Gets or sets the employee's last name.
FullName	string	X		Gets the employee's full name.
Department	string	X	X	Gets or sets the employee's department.
AnnualSalary	double	X	private	Gets the employee's annual salary.

Notes

- FullName is a derived property that returns LastName, FirstName.

Methods

```
public void RaiseSalary(double percent)
```

Notes

- RaiseSalary(double percent) increases the current annual salary by the percentage provided (e.g. 5.5 represents 5.5%)

Constructor

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

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

Airplane

Properties

Property	Data Type	Get	Set	Description
PlaneNumber	string	X	readonly	Gets the six-character plane number.
BookedFirstClassSeats	int	X	private	Gets the number of already booked first class seats
AvailableFirstClassSeats	int	X		Gets the number of available first class seats.
TotalFirstClassSeats	int	X	readonly	Gets the number of total first class seats.
BookedCoachSeats	int	X	private	Gets the number of already booked coach seats
AvailableCoachSeats	int	X		Gets the number of available coach seats.

TotalCoachSeats	int	X	readonly	Gets the number of total coach seats.

Notes

- AvailableFirstClassSeats is a derived property calculated by subtracting BookedFirstClassSeats from TotalFirstClassSeats
- AvailableCoachSeats is a derived property calculated by subtracting BookedCoachSeats from TotalCoachSeats

Constructors

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

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

- planeNumber is the assigned plane number to the airplane.
- 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 forFirstClass is true, add totalNumberOfSeats to the value for BookedFirstClassSeats
    - if forFirstClass is false, add totalNumberOfSeats to the value for BookedCoachSeats
    - return true if there were enough seats to make the reservation

Television

Properties

Property	Data Type	Get	Set	Description
IsOn	bool	X	private	Gets whether or not the TV is turned on.
CurrentChannel	int	X	private	Gets the value for the current channel. Channel levels go between 3 and 18.
CurrentVolume	int	X	private	Gets 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() inceases 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

Properties

Property	Data Type	Get	Set	Description
CurrentLevel	int	X	private	Gets the current floor that the elevator is on.
NumberOfLevels	int	X	readonly	Gets the number of levels available to the elevator.
DoorIsOpen	bool	X	private	Gets if 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 numberOfLevels)
```

- numberOfLevels 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.