# Challenge 1: Implement an Abstract Method in a Base Class

Can you implement an abstract method of a base class? A solution is placed in the solution section to help you, but we would suggest that you try to solve it on your own first.

WE'LL COVER THE FOLLOWING



## Problem Statement #

We have already implemented a `Book` class which has an abstract method, `GetDetails()`, a parameterized constructor, and three private fields having their respective properties with `get` accessors defined:

- `_name` with the property, `Name`
- `_author` with the property, `Author`
- `_price` with the property, `Price`

Write a `MyBook` class that inherits from the `Book` class and has a parameterized constructor taking these parameters:

- **string title**
- **string author**
- **string price**

Implement the `Book` class' `getdetails()` method in the `MyBook` class so that it

returns the `MyBook` details.

### Input

```
    Call to MyBook constructor passing name, author, and price.
    Call to `GetDetails()` method to return the details of a book.
```

### Output

```
Returns the details of the book.
```

### Sample Input

```
Book myBook = new MyBook("Harry Potter", "J.k. Rowling", "100");
```

### Sample Output

```
"Harry Potter, J.k. Rowling, 100"
```

## Coding Exercise

First, take a close look and design a step-by-step algorithm before jumping to the implementation. This problem is designed for practice, so try to solve it on your own. If you get stuck, you can always refer to the solution provided in the solution review.

**Good luck!**

```
// Abstarct Book Class
abstract class Book {

  // Private Fields
  private string _name;
  private string _author;
  private string _price;

  protected string Name{
    get {return this._name;}
  }
   protected string Author{
    get {return this._author;}
  }
   protected string Price{
    get {return this._price;}
  }

  // Parameterized Constructor
```

```
  public Book(string name, string author, string price) {
    this._name = name;
    this._author = author;
    this._price = price;
  }

  // Abstract Method
  public abstract string GetDetails();

}

// Class MyBook extending Book Class
class MyBook : Book {

  // Parameterized Constructor
  public MyBook(string name, string author, string price)
    : base(name, author, price)
  { }

  // Overrideing the getDetails Abstract Method of the Base Class
  public override string GetDetails() {
    // Write your code here
    return "";
  }

}
```

The solution will be explained in the next lesson.