Solution Review: Implement an Abstract Method in a Base Class

This review provides a detailed analysis to solve the 'Implement an Abstract Method in a Base Class' challenge.

WE'LL COVER THE FOLLOWING

Solution

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

  // Protected fields
  protected String name;
  protected String author;
  protected String 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();

}

// MyBook class extending Book class
class MyBook extends Book {

  // Parameterized constructor
  public MyBook(String name, String author, String price) {
    super(name, author, price); // Calling base class constructor
  }

  // Override the getDetails method of the Base Class
  public String getDetails() {
    return name + ", " + author + ", " + price;
  }
```

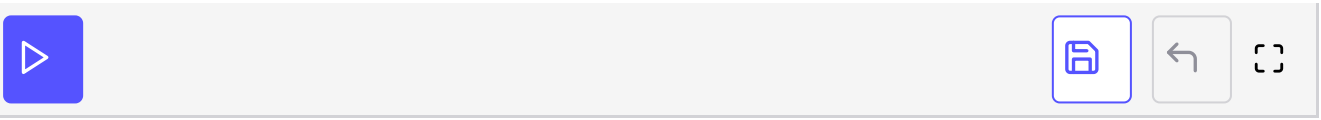
```
}

class Demo {

  public static void main(String args[]) {
    Book myBook = new MyBook("Harry Potter", "J.k. Rowling", "100");
    System.out.println(myBook.getDetails());

  }

}
```



Explanation

- **Line 23:** Extended `MyBook` class from the `Book` class.
- **Line 28:** Called the base class Constructor.
- **Line 32:** The abstract method `getDetails()` is overridden.
- **Line 34:** Implemented the Overridden abstract method `getDetails()`.