

Day 13: Abstract Classes!

Welcome to Day 13! Check out [this video](#) reviewing [abstract classes](#), or just jump into the problem. It's important to understand [Inheritance](#) before attempting this challenge.

Following is an example of abstract class in java:

```
abstract class Book{
    String title;
    String author;
    Book(String t,String a){
        title=t;
        author=a;
    }
    abstract void display();
}
```

An abstract class is a class that cannot be instantiated, meaning you cannot create a *new* instance of it. Attempting to instantiate an abstract class (such as *abstract class Book*, provided for you in the editor) like so:

```
Book new_novel=new Book("The Alchemist","Paulo Coelho");
```

results in **error: Book is abstract; cannot be instantiated**. This type of class is only meant to serve as a base or blueprint for connecting the *subclasses* that *inherit* it.

To use an abstract class, you must create its *subclass* and then instantiate the *subclass*. Any *abstract methods* declared in the abstract class (such as *abstract void display()* in *abstract class Book*) must be implemented by the subclass.

In the editor we have provided the abstract *Book* class and a *Solution* class. In the *Solution* class we created instance of a class called *MyBook*. Your task is to write just the *MyBook* class. The class *MyBook* mustn't be *public*.

Note: Since this is a very specific Object-Oriented topic, we have only enabled a few languages for which abstract classes make sense. If you don't find your favorite language, try to experiment with the provided languages or come back for the upcoming challenges.

Input Format

Input from *STDIN* is already handled in the code given in the editor. *MyBook*'s constructor must have the following parameters: *String title*, *String author*, and *int price*.

Output Format

The *void display()* method should print and label the respective *title*, *author*, and *price* of the book's instance (with each value on its own line) like so:

```
Title: $title
Author: $author
Price: $price
```

Note: The `\$` is prepended to variable names to indicate they are placeholders for variables.

Sample Input

The Alchemist
Paulo Coelho
248

Sample Output

Title: The Alchemist
Author: Paulo Coelho
Price: 248