



Your Day 13: Abstract Classes submission got 30.00 points.

[Share](#)[Tweet](#)[Try the Next Challenge](#) | [Try a Random Challenge](#)

Day 13: Abstract Classes

 by [Shafaet](#)

Problem

Submissions

Leaderboard

Discussions

Editorial

Tutorial

Objective

Today, we're taking what we learned yesterday about [Inheritance](#) and extending it to [Abstract Classes](#). Because this is a very specific Object-Oriented concept, submissions are limited to the few languages that use this construct. Check out the [Tutorial](#) tab for learning materials and an instructional video!

Task

Given a *Book* class and a *Solution* class, write a *MyBook* class that does the following:

- Inherits from *Book*
- Has a parameterized constructor taking these **3** parameters:
 - string *title*
 - string *author*
 - int *price*
- Implements the *Book* class' abstract *display()* method so it prints these **3** lines:
 - Title:**, a space, and then the current instance's *title*.
 - Author:**, a space, and then the current instance's *author*.
 - Price:**, a space, and then the current instance's *price*.

Note: Because these classes are being written in the same file, you must not use an access modifier (e.g.: **public**) when declaring *MyBook* or your code will not execute.

Input Format

You are not responsible for reading any input from stdin. The *Solution* class creates a *Book* object and calls the *MyBook* class constructor (passing it the necessary arguments). It then calls the *display* method on the *Book* object.

Output Format

The *void display()* method should print and label the respective *title*, *author*, and *price* of the *MyBook* object'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 following input from stdin is handled by the locked stub code in your editor:

```
The Alchemist
Paulo Coelho
248
```

Sample Output

The following output is printed by your `display()` method:



```
Title: The Alchemist
Author: Paulo Coelho
Price: 248
```




Submissions: 7519

Max Score: 30

Difficulty: Easy

[More](#)

Current Buffer (saved locally, editable)  

Python 2   

```
1  from abc import ABCMeta, abstractmethod
2  class Book:
3      __metaclass__ = ABCMeta
4      def __init__(self,title,author):
5          self.title=title
6          self.author=author
7      @abstractmethod
8      def display(): pass
9
10 #Write MyBook class
11 class MyBook(Book):
12     def __init__(self,title,author,price):
13         Book.__init__(self,title,author)
14         self.price=price
15     def display(self):
16         print 'Title:',title
17         print 'Author:',author
18         print 'Price:',price
19
20 title=raw_input()
21 author=raw_input()
22 price=int(raw_input())
23 new_novel=MyBook(title,author,price)
24 new_novel.display()
```

Line: 14 Col: 21

 Upload Code as File ☐ Test against custom input

Run Code

Submit Code

Congrats, you solved this challenge!

✓ Test Case #0

✓ Test Case #1

✓ Test Case #2

[Next Challenge](#)

Copyright © 2016 HackerRank. All Rights Reserved

Join us on IRC at [#hackerrank](#) on freenode for hugs or bugs.

[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)