

PolyU COMP2021 Assignment 2

OO Programming

Goal: By doing this assignment, you will understand more about class hierarchy and polymorphism.

The assignment will be checked using **JDK version 1.7.0**.

In this assignment, you will use a java program to build a very simple bookstore inventory system.

Suppose there are 8 kinds of books in the bookstore:

Biographies – Books which portray persons life

History – Well-written, well-researched books for history students and history enthusiasts

Horror – Books which involve scary stories

Journals – Many publications issued at stated intervals

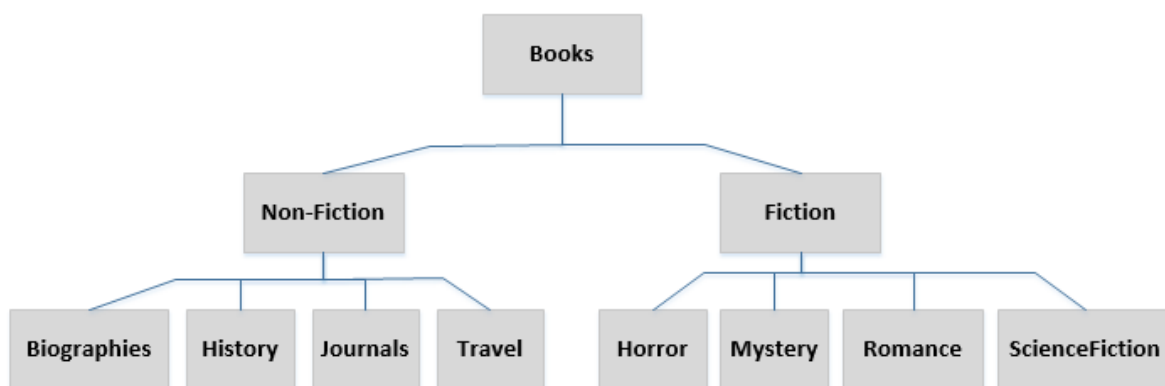
Mystery – Books emphasize on the puzzle or suspense element

Romance – Books which involve contemporary love stories

Science Fiction – Books which is a genre of fiction dealing with imaginative content

Travel – Books which meet the needs of adventure travelers

They belong to two different categories: **Fiction** book, **Non-Fiction** book:



To begin with

You can first enter into **demo** folder in console type **java Assign2Demo** or double click Assign2Demo.

We will see something like:

The screenshot shows a Java Swing window titled "GUI". Inside, there are three input fields labeled "Book Name", "Price", and "Quantity". To the right of the "Price" field is a label "The total value of all books is: 0". To the right of the "Quantity" field is a label "The total number of all books is: 0". Below these are two rows of buttons: the first row contains "Biographies", "History", "Travel", and "Journals"; the second row contains "Science Fiction", "Mystery", "Horror", and "Romance". Below the buttons are two checkboxes: "☐ Non-Fiction" and "☐ Fiction". At the bottom are two large buttons: a green one labeled "Increase Price" and a red one labeled "Decrease Price".

You can add a book by typing the name, price and quantity in the appropriate field, then click a button such as **Biographies**. After that, you will see “the total number of all books” and “the total value of all books” are updated. By choosing either category “Non-Fiction” or “Fiction”, clicking buttons **Increase Price** or **Decrease Price** will increase or decrease the price(s) of all added book by a **hidden** percentage. As a result, “the total value of all books” shall be increased or decreased by a certain amount.

The Assignment Part

You are given 4 sets of files:

- ReadMe.pdf (this file)
- **/src (the .java files for you to start your assignment)**
 - some .class files are given (without src)
 - e.g., NonFiction.class is given
- /javadoc (shows the API of all given classes)
- /demo (execute “java Assign2Demo” in this directory and you will see what we expect you to implement)

Your job: Implement 9 .java files under the /src directory:

- (1) Biographies.java
- (2) History.java
- (3) Horror.java
- (4) Journals.java
- (5) Mystery.java,
- (6) Romance.java
- (7) ScienceFiction.java
- (8) Travel.java
- (9) SetOfActions.java

Submit these 9 .java source files through learn.polyu.edu.hk assignment box in .zip format.

If you successfully finish this assignment ...

That means you have implemented 9 .java files. So, if you compile like this under /src:
**javac SetOfActions.java Biographies.java History.java Horror.java Journals.java
Mystery.java Romance.java ScienceFiction.java Travel.java**

And then
java Assign2Demo

It will behave the same as it is in the **demo**.

You shall use the demo GUI to check the correctness of your implementation.

Note that:

- You don't need to worry about building the GUI in this assignment
 - They are given to you as
 - Assign2Demo.class
 - The class with the main method
 - CreateGUI*.class
 - The classes that render the GUI; invoked by Assign2Demo.class
 - SetOfActions.java
 - Whenever a button X is clicked in the GUI, Assign2Demo.class and CreateGUI.class will invoke the method addX in this file
 - In the demo example above, if "Biographies" button is clicked, then the method addBiographies will be invoked with the following parameters: addBiographies (<empty list>, "Albert Einstein", 150,10)
 - Continuing, if "Horror" button is next clicked with a name "Scary Adventure", price 100 and quantity 12, then the method addHorror will be invoked with the following parameters: addHorror (<Biographies Albert Einstein>, "Scary Adventure", 100, 12)
 - That is, the ArrayList st holds the current list of created Books
- **The price adjustment *formula* of each book type (i.e., NonFiction and Fiction) is not given to you. But the price adjustment *implementation* is actually encapsulated somewhere. Think about it.**
- **You cannot use any tool to examine the .class content. We can detect that and you will get 0 marks by doing that.**

Deadline: 9 Oct 2015, 11:00

Late Penalty

late x day: your score = raw score * (100 - 20x)%

Grading Policy:

Two measures:

- 1) Number of classes you have implemented correctly (primary)
- 2) Submission time (secondary)

The grading policy is similar to assignment 1, please refer to the following table:

Implemented Classes	Score
1	11 - 20
2	21 - 30
3	31 - 40
4	41 - 50
5	51 - 60
6	61 - 70
7	71 - 80
8	81 - 90
9	91-100

For students who passed the same number of classes, the score will be differentiated by the submission time as we did in assignment 1.

Hint

You can focus on writing just one class at first. For example, you can write **Biographies.java** first. In order to make the compiling work, you can first comment out addHistory, addHorror, addJournals, addMystery, addRomance, addScienceFiction, addTravel methods of **SetOfActions.java**, then compile the **Biographies.java** and **SetOfActions.java** and run the GUI. By that time, you can test only **Biographies** button.

Plagiarism

It is easy to detect the similarity of source files, and cases will be strictly handled according to the University's regulation, so please don't risk doing that.