1. Problem : Given a list of books, perform filtering and mapping operations on the books using Java 8 stream api and lambda functions.

Consider the following Books data that is already created in the program

|  |  |  |
| --- | --- | --- |
| Title | Pages | Price |
| Book 1 | 900 | 450 |
| Book 2 | 560 | 1000 |
| Book 3 | 400 | 600 |

Function Description:

1. getBooksByPriceGt : Takes list of books and price as input parameters and is suppose return a list of Book objects whose price is greater than the passed input value

Test cases

1. Input: price: 100, books: <check the above table>

Output: [Book 1, Book 2, Book 3]

1. Input: 2000

Output: []

1. Input: price: 500, books: <check the above table>

Output: [Book 2, Book 3]

1. getBooksByPriceLtSortedByPage : Takes a list of books and price as input parameters and is supposed to return a list of Book objects whose price is less than the passed input price value. Also the final results returned must be sorted in the descending order of pages.

Test cases

1. Input: price: 100, books: <check the above table>

Output: []

1. Input: price: 2000, books: <check the above table>

Output: [Book 1, Book 2, Book 3]

1. Input: price: 700, books: <check the above table>

Output: [Book 1, Book 3]

1. getBooksByPagesGtSortedByPriceTitleUpperCased : Takes a list of books and pages as input parameters and is supposed to return a list of Book objects whose pages are greater than the passed input pages value. Also the final results returned must be sorted in the descending order of price and the title of each of the Book objects in the list returned must be upper case.

Test cases

1. Input: pages: 500, books: <check the above table>

Output: [BOOK 2, BOOK 1]

1. Input: pages: 2000, books: <check the above table>

Output: []

1. Input: pages: 100, books: <check the above table>

Output: [BOOK 2, BOOK 3, BOOK 1]