

# Extended Java: Lab Assignment

## 1

Make sure you have downloaded the IntelliJ project files for the assignment. Use this template to complete your answer.

The coursework is out of 50 marks, 30 marks for question 1 and 20 marks for question 2

### Question 1: BookList (30 marks)

While writing your answer to question 1 make use of BookMain to test your solutions to the various parts of the question. This class won't be marked. You will also use the Book and BookCollection classes in this question.

1. Complete the Book class. It should have the following private properties. Your class should include a constructor with arguments for each of these properties. Also create getters and setters for each property

- title (string)
- author (String)
- ISBN (long)
- pages (int)
- copiesAvailable (int)
- copiesOnLoan (int)

**[2 marks]**

2. Complete the BookCollection constructor so that it loads the file path argument as a File. Then use a Scanner to read in the file and populate the books arrayList. When complete books should have 100 items in it.

**Hint:** You will need to skip over the first line of the file (the column headers), otherwise you will get an error.

**[6 marks]**

3. Complete the function getAuthors so that it returns a HashSet of all the authors in books.

**[5 marks]**

4. Complete the function getLongBooks so it returns an ArrayList of Books which have over 750 words in them

**[5 marks]**

5. Complete the function getBookByTitle so it returns the Book object for the given title. If a title not in the list is given return null.

**[5 marks]**

6. Complete the function getMostPopularBooks which returns an array of the 10 most popular books (That is those that currently have most copies on loan).

**[7 marks]**

## Question 2: Deck of cards (20 marks)

In this question you will complete the Hand class and SortCards class. Spend some time analysing the Card, Deck, Hand and SortCards classes.

### SortCard class [3 marks]

Complete the comparator method compare so that the card objects are compared as follows:

- Where o1 has a higher value property than o2 return 1.
- Where o1 has a lower value property than o2 return -1;
- Where the value property of both o1 and o2 is the same return 0.

**Hint:** Card has a method getNumericValue that will convert the String value of the card to an integer for easy comparison. In this implementation Aces are the highest value.









































### Hand class [17 marks]

Complete the getHandRank method. This should return the correct string from the ranks array of Hand based on the cards that have been dealt.

Base your determination of the rank on the classic poker hands. See the code and the image below for a guide to examples of the 10 ranks these. Using the SortCards class you completed above the cards of the hand are sorted. This will make this task much easier!

Again, use the main method in CardMain to test your code as you develop your solution. You should also make use of the dealSpecificCard method in deck. This will help you to deal specific hands that you can check your code against.

## Poker Hand Rankings

1	Royal Flush	6	Straight						
									
2	Straight Flush	7	Three of a Kind						
									
3	Four of a Kind	8	Two Pair						
									
4	Full House	9	One Pair						
									
5	Flush	10	High Card						
