CIS 3515 Assignment 7

Instructions: Create a new application called **BookShelf**, that will take shape over the next few labs.

Your application will incorporate a dynamic (responsive) UI, that will display a **single fragment** at a time if the application is being run **on a small screen (cell phone) in portrait mode**, but must display **two fragments simultaneously** if being run **in landscape mode, or on a large device (tablet), regardless of orientation** (you can create a virtual tablet device to test this configuration).

1. Create a **Book** class in your application that contains the following fields for a book:

    1. title: String – The title of the book

    2. author: String – The author of the book

2. Create a **BookList** class in your application that is used to store a set of Book objects. This class should provide the standard set of collection-like operations, such as:

    1. add(Book) : void

    2. remove(Book) : void

    3. get(int) : Book

    4. size(): int

    5. Any other operation you might find useful

3. In your activity:

    1. Create an instance of your BookList class, and populate it with a random assortment of Book objects. Simply pick some random book titles, real or made up – you should have at least 10 books. You can store and fetch that information from a string resource, or simply create a helper method or helper class to generate the books for you – all that matters is that in the end you have an BookList object that contains 10 Book objects, where each Book has a title and author value.

4. Create 3 layout files for your activity for the following scenarios:

    1. small-portrait: this will contain a single display container (this could be the default)

    2. small-landscape: this will contain 2 display containers

    3. tablet: this will contain 2 display containers

5. Create two fragments: BookListFragment, and BookDetailsFragment.

   1. BookListFragment will contain a ListView that, when provided with an BookList object, will display the book title and book author for each book in its ListView. The title and author should each be displayed in their own TextView – that is, each item in the ListView will contain 2 TextViews, one showing the book title, and the other showing the book author. When the user clicks one of the books in the ListView, the fragment should invoke a method in its parent with the **index** of the book that was clicked. The fragment should have a factory method (**newInstance(BookList bookList)**) that creates a fragment using the provided books to set up the fragment's initial state

   2. BookDetailsFragment will have the ability to display a book title and author when provided with a book object. It will have a single public API method, **displayBook(Book book)**, which will display the book title in a TextView with a large font size, and the author in a TextView with a slightly smaller font size. Additionally it should have a factory method (**newInstance(Book book)**) that creates a fragment using a book to set up the fragment's initial state.

6. Your activity will then use the Master-Detail pattern when determining when to display a certain fragment.

   1. If the activity is using a layout that contains a single container (portrait mode on a small screen), it should display an instance of the BookListFragment at startup. When the user clicks a book from the BookListFragment, an instance of BookDetailsFragment should be launched to replace the BookListFragment, displaying information on the book that was clicked. The user should then be able to reverse this operation and select a new book.

   2. If the activity is using a layout that contains two containers, then one container is used to display a **BookListFragment**, and the other to display a **BookDetailsFragment**. Clicking a book title in the BookListFragment should have the BooksDetailsFragment display the book title, but it should not load a new instance for each click. Instead, a single instance of BookDetailsFragment should be loaded, and it should change the book that is displayed whenever the user clicks a new book from the BookListFragment.

   3. If the switches their mobile device from one orientation to another, your application should not recreate fragments if they already exist. Instead it should continue to use the fragments that we previously attached to the activity.

   4. If the user is viewing a selected book in Portrait mode (on a small screen), and then switches the device to Landscape mode, the selected book should be shown automatically. The same is true if the user switches from Landscape to Portrait after selecting a book – the selected book should be visible first instead of the list of books.

7. **COMMIT and PUSH to Github FREQUENTLY! Do NOT do a single commit and push at the end of development. Upload your Github link to Canvas.**

Rubric

Application has configuration for small-portrait, large portrait, and landscape orientations	10%
Both BookListFragment and BookDetailsFragment have factory methods that can create new fragment instances	10%
BookListFragment has ListView that shows both title and author in their own TextViews	10%
BookListFragment properly communicates selected book to parent activity	10%
Proper master-detail implementation when in small-portrait mode (New instance of BookDetailsFragment being created when book is clicked)	15%
Proper master-detail implementation when in large-portrait or landscape mode (Single instance of BookDetailsFragment being updated when book is clicked)	15%
Fragments are retained and reused (when possible) during configuration changes instead of creating new instances	15%
Book selection is remembered across orientation changes	15%