

- Don't forget to set your Eclipse workspace and working set.
- You must submit the JAR file, exported (with source code), from your Eclipse project.
- You must check your JAR file to make sure all the source files (.java files) are present. It can be opened with file compression programs such as 7-zip or Winrar.
- Failure to export properly will result in your work not getting marked.

1) To submit:

- Export your project to a JAR file, with source code.
- Name your JAR file ID\_Week06\_Q2.jar. For example, 6623110021\_Week06\_Q2.jar
- Submit the JAR file on MyCourseville.

(21 marks total -> will be scaled to equal to other homework)

Stack implementations are provided.

1. (7 marks) Create a class StackUtility. This class is a utility class that can manipulate a given stack. Write method:

```
public static MyStack removeRange(MyStack s, int i, int j) throws Exception {
```

This method removes data from position i to j (inclusive) from the stack (the stack must actually change).

- Let position 0 be the top of the stack.
- You can assume that the stack has at least 1 data.
- You can assume i and j never indicate positions outside the stack, and i comes before j.
- i and j can be the same position.
- This method must work on both StackArray and StackLinkedList.
- **This method can only use methods originally defined in MyStack (you can call constructor though). You are not allowed to modify StackArray and StackLinkedList in order to complete this question (you get 0 mark if you do not follow this instruction).**

Example:

If your stack, s, originally looks like:

11
9
6
8
12

removeRange(s, 1, 2) will return



11
8
12

2. (14 marks) In this question we are actually going to write the removeRange method directly in class StackLinkedList and StackArray. The actual stack object must be modified.

```
public void removeRange(int i, int j) (can throws Exception in the linked list version)
```

- Let position 0 be the top of the stack.
- You can assume that the stack has at least 1 data.

- You can assume i and j never indicate positions outside the stack, and i comes before j.
- i and j can be the same position.
- **Do not change the JUnit file.**
- **Do not call methods define in MyStack and CDLinkedList. You must manipulate the array or linked list directly.**
- **If you don't follow this instruction in bold, you get 0 mark.**
- JUnit Test cases used when marking will be different from the JUnit you have, but they test the same logic. So make sure your program works for any possible inputs.