

CSCE 314 [Sections 202, 502] Programming Languages – Fall 2016

Anandi Dutta

Assignment 6

Assigned on Monday, November 21, 2016

Electronic submission to eCampus due at 23:59, Friday, 12/02/2016

By electronically submitting this assignment to eCampus by logging in to your account, you are signing electronically on the following Aggie Honor Code: "On my honor, as an Aggie, I have neither given nor received any unauthorized aid on any portion of the academic work included in this assignment."

Note 1: This homework set is individual homework, not a team-based effort. Discussion of the concept is encouraged, but actual write-up of the solutions must be done individually.

Note 2: Turn in one yourLastName-yourFirstName-hw5.tar or yourLastName-yourFirstName-hw5.zip file on eCampus, nothing else. What to submit is detailed below.

Note 3: All Java code that you submit must compile without errors using javac of Java 8 (that is installed in the departmental servers, linux.cse.tamu.edu and compute.cse.tamu.edu). If your code does not compile, you will likely receive zero points for this assignment.

Note 4: Remember to put the head comment in your files, including your name, your UIN, and acknowledgements of any help received in doing this assignment. Again, remember the honor code.

Note 5: Problem [1] contains 50 points. Each method contains 6 points each, GenericStack contains 30 points total. The test program contains 20 points.

Problem [2] contains 50 points total. The implementation contains 35 points, the test contains 15 points.

Problem [3] contains 15 points.

[1] Define a GenericStack class that extends ArrayList. Therefore, you are implementing GenericStack using inheritance. Write a test program that prompts the user to enter five strings and displays them in reverse order. In your GenericStack class you need to write these methods: getSize(), peek(), push(), pop(), isEmpty().

getSize() :: it will provide you the stack size

peek() :: it will provide you the object of the top of the object without removing it

push() :: user will use this method to push an item onto top of the stack

pop() :: this method is used to remove the object at the top of this stack and returns that object as the value of this function

isEmpty :: this method will check if the stack is empty or not.

[2] Implement the merge sort algorithm using Java. This is an open-ended HW. We are presenting our requirements as clients, it is your job to fulfill that. You need to write a test program that will ask the user to provide an array of characters. Then your code will sort the array and provide the sorted array to the user. You cannot use any built-in sort code from Java API, you have to write your own merge sort code.

Hint: You might want to use the `Comparable` interface.

Hint: You might want to have a method like this: it will do the swapping for your sorting algorithm.

```
private static void exch(Comparable[] a, int i, int j)
```

Hint: You might want to have another helper method allowing you to compare an item with another

```
private static boolean less(Comparable v, Comparable w)
```

For Section 202; 15 points; Extra Credit for Section 502

[3] Write a program incorporating this generic method that returns the maximum element in a two dimensional array. Also, write a test for it.

```
public static <E extends Comparable<E>> E max(E[] [] list)
```