Self-check 4.1 questions

1. Results are:
2. The String “Jane” gets pushed into the stack.
3. Top equals the String “Jane”.
4. The String Joseph gets pushed into the stack.
5. NextTop equals the String Joseph and gets removed from or popped out of the stack.
6. The top of names gets printed and then it gets popped out of the stack.
7. If peek was used into of pop in question 2, the top of names would stay in the stack and get printed.

Self-check 4.3 1 & 3 questions

1. The first two pushes add \* and i to the ArrayList, the underlying data structure

The first Pop gives char ch1 i and removes it from stack.

The second pop just removes \* and returns it but nothing happens with it.

The third push add a space to the end of the ArrayList.

The Char ch3 gets the space from the top of the stack with no removal.

3. The first two pushes add i and s to the end of the LinkedList, the underlying data structure

The first Pop gives char ch1 s and removes it from stack by shifting the node past s’s node.

The second pop just removes i and returns it but nothing happens with it.

The third push add a space to the end of the LinkedList.

The Char ch3 gets the space from the top of the stack with no removal.

Self-check 4.4 question 1

First, 4 and 7 is pushed in the stack. Then, with + operand, it pushes in the plus. Which adds 4+7 to make 11. The 5 is then added to the stack. Then with \* operand being pushed in, the 5 is multiplied to the 11, making 55. 3 is next added and then 3 subtracted from 55 due to the – operand being pushed. Next, 6 is added to the stack and then 55 is divided by 6 because of the / operand being pushed to the stack. The value is 52/6 which is 8.667.

First, 13 and 2 are pushed. Second, 13 and 2 are multiplied because the \* operand is pushed in. Third, 5 is pushed and 26 is divided by 5 since the / is pushed. Fourth, 6, 2, and 5 are added to the stack. The first has the five multiplied to the 6 and 2 which does nothing right now, due to the pushing of the \*. Then, the 6 and 2 subtract each other (by the pushing of -), making 4 times 5, since the operands make 5\*(6-2). Then finally, the 5.2 is added to 20(by the pushing of +), making 25.2.

First, the 5 and 4 are pushed. Then, the 5 is divided by 4 to make 1.25 because of the / operand being pushed. Second, 6 and 7 are pushed. Then, 6 and 7 are subtracted to make -1 because of the – operand being pushed. Third, the 4 and 2 are pushed. Then 4 is divided by 2 to make 2 because of the / operand. Fourth, - is pushed into the stack which has 1.25 subtracted by -1. Finally, -2.25 is multiplied by 2 by the \* operand being added. That makes -5 as the solution.