



Homework Assignment 5

Due date: October 16, 11:55PM

Problem 1

Given the IntegerQueue ADT below:

Informal Specification

The IntegerQueue contains a (possibly empty) collection of objects of type Integer. The queue supports the following operations:

insert	This operation adds a Integer object, given as a parameter, to the end of the queue of integers.
remove	This operation removes and returns a Integer object from the front of the queue of integers. If the queue is empty, null should be returned.
toString	This operation produces a String that contains all Integer objects stored in the queue from front to back separated by a single comma and a single space. If the queue is empty, an empty string should be returned.

State the return value and show the content of the, initially empty, queue of Integer objects after each of the following operations. If any of the operations below is invalid or would cause program to crash, state it and explain what is wrong with it (then proceed with the next step ignoring the invalid line). Assume that queue is a reference of type IntegerQueue.

```
queue.insert( 15 );
queue.insert( 3 );
queue.insert( -15 );
queue.insert( 35 );
queue.remove();
queue.remove();
queue.remove();
queue.insert( 13 );
queue.remove();
queue.remove();
queue.remove();
queue.insert( 3 );
```

Assume that the implementation is array based and follows the efficiency ideas that we discussed in class. Assume that the initial capacity of the array to store the queue is equal to 4 and that its size is doubled whenever the array runs out of room.



Problem 2

Given the CharStack ADT that we discussed in class:

Informal Specification

The CharStack contains a (possibly empty) collection of objects of type Character. The stack supports the following operations:

insert / push	This operation adds a Character object, given as a parameter, to the top of the stack of characters.
remove / pop	This operation removes and returns a Character object from the top of the stack of characters.
peek	This operation returns a Character object from the top of the stack of characters.
toString	This operation produces a meaningful String representation of the stack.

Show the content of the, initially empty, stack of Character objects after each of the following operations. If any of the operations below is invalid or would cause program to crash, state it and explain what is wrong with it. Assume that `stack` is a reference of type CharStack.

```
stack.push('c');
stack.push( new Character('s') );
stack.pop();
char p = 's';
stack.push( p );
stack.push( p );
stack.push( new Character('1') );
stack.peek();
stack.pop();
stack.push('%');
stack.peek();
stack.push('A');
stack.push('X');
stack.pop();
stack.pop();
```

Assume that the implementation is array based and follows the efficiency ideas that we discussed in class. Assume that the initial capacity of the array to store the stack is equal to 4 and that its size is doubled whenever the array runs out of room.

Problem 3

The GenericList interface and its GenericLinkedList implementation used in class (see the source code for lecture 5) provide an insertion method that always adds a new node to the back of the list. Assuming that the data item that is stored in the GenericNode object implements the Comparable interface (i.e. has standard definitions of `compareTo(...)`), write the method

```
void orderedInsert( T item )
```

that adds a new node to the list in a sorted order (from smallest to largest).

Make sure to document your code (tell the reader what you are doing).



How and what to submit

You can (but do not have to) use the template provided at

<https://docs.google.com/a/nyu.edu/document/d/1AKwqpEwzzChJmioeDoyrNLifSmS2PXVD8sG43Y402jk/edit?usp=sharing>

to complete your solution.

The completed solutions should be submitted as a single PDF document to NYU Classes.

Warning: Make sure you submit a PDF document. We will ignore any files that are not PDF (that includes text files, Word documents, Google docs, RTF files, etc.).