Iterators/Generics – Stack

**Purpose**

This lab was designed to teach you more about stacks, inner classes, iterators and generics.

**Description**

Write a program to implement your own generic Stack class. It should contain a default constructor (array size of 10), push, pop, isEmpty, peek, toString and iterator(can just call listIterator), listIterator(sub interface of iterator) methods. Use dynamic array logic. It should double its size when it reaches the current threshold. It should reduce in size appropriately. You should clear up loitering by setting the removed objects to null. The bulk of this lab lies in the listIterator. I would use an anonymous inner class and let the IDE write all of the method signatures for you. It is **absolutely necessary** to test your code against Java’s stack class to earn a high grade. Be sure to throw exceptions in pop and peek and throw the appropriate exception if you try to modify the stack while in an enhanced for loop. You must also write public E[] getArray() for the autograder.

**Program Shell**

GenericIterableStack.java

**Starting Class**

public class GenericIterableStack // you must use generics and implement Iterable{

private E[ ] ary;

public GenericIterableStack(){}

public E[] getArray(){return ary;}

// add push, pop, peek, isEmpty, toString iterator(can call listIterator) and listIterator

// they must follow the java api!

}