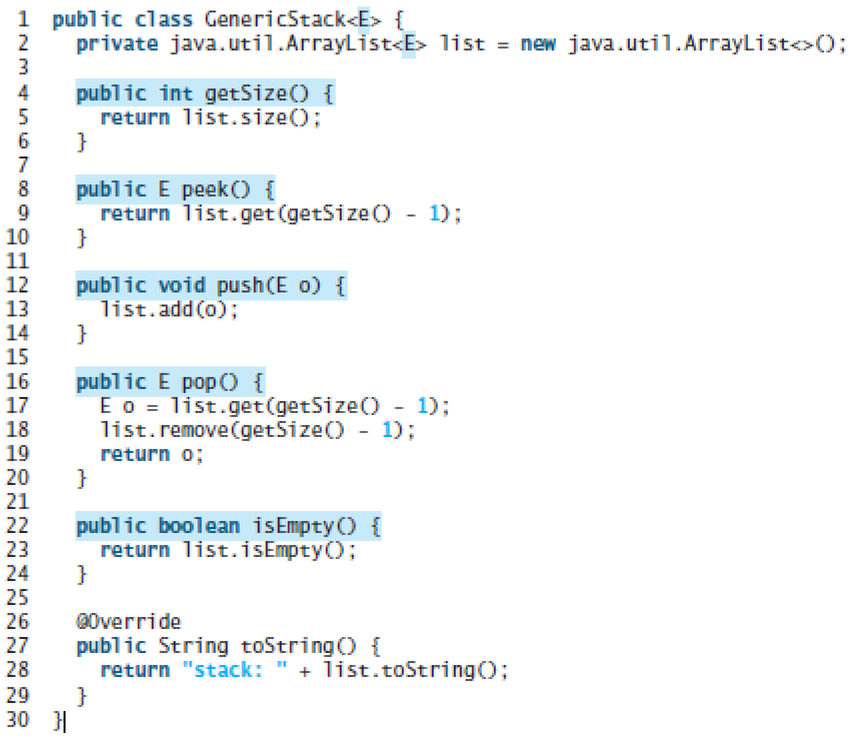
1. **(2points)** Revise the following code:



```
public class GenericStack<E> {
  private java.util.ArrayList<E> list = new java.util.ArrayList<>();

  public int getSize() {
    return list.size();
  }

  public E peek() {
    return list.get(getSize() - 1);
  }

  public void push(E o) {
    list.add(o);
  }

  public E pop() {
    E o = list.get(getSize() - 1);
    list.remove(getSize() - 1);
    return o;
  }

  public boolean isEmpty() {
    return list.isEmpty();
  }

  @Override
  public String toString() {
    return "stack: " + list.toString();
  }
}
```

Revise the GenericStack class above to implement it using an array rather than an ArrayList:

- You must have a constructor to construct a stack with the default initial capacity.
- You must have another constructor to construct a stack with a specified initial capacity.
- You should check the array size before adding a new element to the stack. If the array is full, create a new array that doubles the current array size and copy the elements from the current array to the new array.

2. **(2 points)** Write the following method that returns a new ArrayList. The new list contains the non-duplicate elements from the original list.

```
public static <E> ArrayList<E> removeDuplicates(ArrayList<E> list)
```

Your method must remove the duplicates when inserted into the following code:

```
import java.util.ArrayList;

public class Exercise19_03 {
  public static void main(String[] args) {
    ArrayList<Integer> list = new ArrayList<Integer>();
    list.add(14);
    list.add(24);
    list.add(14);
    list.add(42);
    list.add(25);

    ArrayList<Integer> newList = removeDuplicates(list);

    System.out.print(newList);
  }

 public static <E> ArrayList<E> removeDuplicates(ArrayList<E> list){

   /* Your implementation here */

  }
}
```

3. **(1 point)** Can you define a custom generic exception class? Why?