Assignment Array Stack<E> CSIS-2420

Learning Objectives:

- Deepen your understanding of arrays and understand their performance implications
- Follow the guidelines provided in the slides
- Implement the functionality specified

Turning In:

Make sure to include a comment with your name and assignment number on top of the source code file.
Submit a jar file that **includes ArrayStack.java** via Canvas.

Max Points: 20

Description:

Write a class **ArrayStack<E>** based on the guidelines provided in sides 12 - 23

- The ArrayStack should have no fixed size limit
- the pop method should avoid loitering, and
- the resizing of the array should be done efficiently as recommended in the resource provided.
- Sedgewick only mentions that popping from an empty stack should result in an exception.
  I want you to throw an **`EmptyStackException`**.
- **There is one thing I want you to do differently though:**
  Your ArrayStack needs to be able to accept elements of any reference type not just Strings

Hint:
Because of erasure Java doesn't allow you to create an array of type **`T`** with the new operator (`new` **`T[1]`** won't work)
You'll need to create an array of type **`Object`** and cast it to **`T[]`** like this: **`(T[]) new Object[1]`**
This is not ideal because it is an unchecked cast but it is the best you can do.
Eclipse will issue a warning. You an suppress the warning. However, you want to suppress the warning only where needed and **not** for the whole class.
Hover over the yellow wiggly line. Eclipse will offer suggestions where to place the SuppressWarnings annotation.
Choose the most restrictive scope (where possible right above the variable declaration)

All you need to turn in for this assignment is the class ArrayStack<E>.
You do need to test your own code though to ensure that it works.
I include some unit tests that should help you get started. Note though that they don't check for proper re-sizing.
So pay special attention to that areay.