# Term Project – Stack Checkpoint

# Warming-up: Java Stack class [Checked: ]

Java already provides a built-in stack. Run StackDriver program that use the Java Stack class and answer the following questions.

1. What is the output?
2. What are contents of the stack after operations?
3. What is the error message when you try to pop() when the stack is empty()?
4. Add elements repeatedly and check how the capacity changes. Describe your findings.
5. In this time, add elements by 100,000 and check the capacity first. Then remove all elements by using clear() and check the capacity. Describe your findings.
6. How can you reduce the capacity when necessary?
7. What is the default stack capacity in Java to begin with? Can it be zero to begin with?

# Part I: Creating StackOfInts class

## Step 1: Stack of Ints [Checked: ]

**Checkpoint:** After you implement StackOfInts including some missing methods, run it while replacing the first line of the warming-up program, **StackDriver.java**.

## Step 2 & 3: Increasing capacity, trimToSize() [Score: /0.5]

**Checkpoint:** The expected output is shown below. This output should be the same as you **alternate** the first two lines of code in the program **StackDriver2.java**.

## Step 4: Decreasing capacity automatically [Score: /0.5]

**Checkpoint:** The capacity does not decrease even though its size became zero.

**Checkpoint**: After you implement the pop() method in **StackOfInts** as instructed, then the capacity becomes smaller as its size becomes smaller.

## Step 5: Printing elements in Stack [Score: /0.5]

**Checkpoint:** When you override toString() method, you must use either StringBuilder or StringBuffer class, **not String**. Use @Override annotation in Java. Run **StackDriver4.java**

## Step 6: Throwing `EmptyStackException` [Score: /0.5]

**Checkpoint:** Run **StackDriver5.java** with StackOfInts. Your output should be the same Exception object thrown.

# Part II: Creating a generic and iterable stack

## Step 1: Using java.util.ArrayList [Score: /0.5]

**Checkpoint:** Implement StackGeneric.java and run it with **StackDriver7.java**.

## Step 2: Java Generics [Score: /1.0]

**Checkpoint:** Complete the StackGeneric2.java, run it with **StackDriver8.java** and pass the actual type arguments.

## Step 3-1: Iterable interface using anonymous inner class

For an example, implement the next StringIterable class first, then continue on this step.

## StringIterable: An iterable interface example [Score: /1.0]

Method I: Using an own separate class

**Checkpoint:** Check the StringIterable class using the StringIterableDriver.java program..

Method II: Using anonymous inner class

**Checkpoint:** Check **StringIterable2.java**, using the same program **StringIterableDriver.java**

## Step 3-2: Iterable interface using anonymous inner class

## Method I [Score: /0.5], Method II [Score: /0.5]

Method I: Using a separate class;

**Checkpoint:** The **StackDriver10.java** now should work with StackGeneric3.

Method II: Using an anonymous inner class while overriding iterator() method

**Checkpoint:** The StackDriver10.java should also work with StackGeneric4.

## Step 4: The simplest solution for this case^^ [Score: /0.5]

**Checkpoint:** Fill the blank with an appropriate code. **StackDriver10.java** should also work with StackGeneric5.

# Part III: Make StackOfInts class iterable [Score: 2.0]

Checkpoint: Run **StackOfIntsIterable.java** with the **StackDriver11.java** program

## Files to submit

Submit the following source files on time. Use **lab9** folder in Piazza

- For Part I: StackOfInts.java
- An example: StringIterable.java, StringIterable2.java
- For Part II; StackGeneric.java, StackGeneric2.java, StackGeneric3.java, StackGeneric4.java, StackGeneric5.java
- For Part III: StackOfIntsIterable.java
- TermProjectStackScores.docx with **Warming-up questions and scores recorded.**

## Due and Grade points

- Due: 11:55 pm, Saturday, Nov 24, 2018
- Grade points:
  - Warming-up questions: 0.0
  - Part I: 2 points, Part II: 3 points, Part III: 2 points
  - An example - StringIterable : 1 point
  - Max penality for wrong or inflated grading: -2.0
- My total score: _____ by myself