

CMPSCI 182 Program Design and Data Structures  
Project 3 (Stacks)  
30 points total  
Due 10/25/22

## Phase I

The objective of Phase I is to be able to use the Array-based ADT given in the textbook.

To start, create a Project named **Project3**. Within **Project3**, create a package named **phase1**. Copy the code for the **StackArrayBased** class, which starts near the bottom of page 365 and ends near the top of page 367 from the textbook into a Java class file named **StackArrayBased** in the **phase1** package.

Copy the code for the **StackInterface**, which starts near the bottom of page 363 and ends near the top of page 365 from the textbook into a Java interface file named **StackInterface** in the **phase1** package.

Copy the code for the **StackException** class, which is given in its entirety at the top of page 365, from the textbook into a Java class file named **StackException** in the **phase1** package.

The array-based implementation of the ADT stack in the **StackArrayBased** class that you copied from the textbook assumes a maximum stack size of 50 items. Modify this implementation so that each time the stack becomes full, the size of the array is doubled.

Add a Java main class named **StackTester** to the **phase1** package which tests your modified implementation of **StackArrayBased**. This test program should create an object from the **StackArrayBased** class you modified and push and pop Integer-type objects onto and off of the stack. The program should attempt to push more than 50 items onto the stack, more than 100 items onto the stack and more than 200 items onto the stack. The ADT's **push ()** method should detect when the array becomes full and double the size of the array so that more items can be pushed onto the stack. Your test program should also **pop ()** items off the stack to ensure that these items are stored in LIFO (Last In First Out) order **(15 points)**.

## Phase II

The objective of Phase II is to be able to use the Reference-based ADT stack given in the textbook.

Create another package named **phase2** in **Project3**. Copy the code for the **StackReferenceBased** class, which starts near the bottom of page 368 and ends near the bottom of page 369 from the textbook into a Java class file named **StackReferenceBased**.

Copy the **StackInterface** class from the **phase1** package and paste it into the **phase2** package. You may have to add a "package phase2;" statement at the top of this source code file if NetBeans does not refactor the file for you.

Copy the **StackException** class from the **phase1** package and paste it into the **phase2** package. You may have to add a "package phase2;" statement at the top of this source code file if NetBeans does not refactor the file for you.

You will also need to copy a **Node** class into the **phase2** package. You may use the **Node** class given near the middle of page 251 of the textbook, or you may use the **Node** class you wrote for Project 2, or you may use the **Node** class I wrote in lab on 9/13/22 and uploaded to Canvas, **Node.java**. However, be aware that if you use a prewritten **Node** class, you may want to change the data type of the item data field to a primitive **char**.

The section "*Recognizing Strings in a Language*" describes a recognition algorithm for the language  $L$ :

$$L = \{w\$w' : w \text{ is a possibly empty string of characters other than } \$, w' = \text{reverse}(w)\}$$

Implement the pseudocoded recognition algorithm on pages 362 to 363 in a test program named **StringRecognizer**. This test program should declare an object from the **StackReferenceBased** class in the **phase2** package and use the **StackReferenceBased** object to determine if a user-given String is "in the language."

For example, if the test program is run and the user enters a String like "abcd\$abcd", the program should display a message that the user-given String is NOT in the language because "abcd\$abcd" is NOT of the form  $w\$w'$  where  $w'$  is  $\text{reverse}(w)$ . If the test program is run and the user enters a String like "abcd\$dcba", the program should display a message that the user-given String IS in the language because "abcd\$dcba" IS of the form  $w\$w'$  where  $w'$  is  $\text{reverse}(w)$  (**15 points**).

When you have completed both phases, ZIP the entire **Project3** into a single ZIP file and submit this ZIP file to Canvas.