

# DATA STRUCTURES: STACKS

**Course Name:** AP Computer Science A  
**Unit/Theme:** Data Structures

**Time Frame:** 86 minutes (2 periods)  
**Grade Level:** 11-12

## CONTENT AND SKILLS

### Learning Objectives:

- Students will be able to create their own implementation of a stack data structure and utilize stacks in their own program.

### Students will be able to...

- Explain what LIFO (or FILO) order is, how it works, and how it relates to stacks.
- Relate stacks to objects or experiences from their own lives.
- Explain how data is added, accessed, and removed from stacks.
- Implement their own stack class in Java using an underlying ArrayList.
- Utilize stacks as a data structure to manage data in Java programs.

### Supports For Diverse Learners

- Students are asked to think about stacks in relation to their own experiences
- Extension opportunities are provided for students that they can tailor to their own interests
- Information is presented to students in multiple representations
- Students are provided with ways to check their own work

## NYS Computer Science and Digital Fluency Standards

List all standards and how learners will meet the standard

- 9-12.IC.7: Investigate the use of computer science in multiple fields.
- 9-12.CT.4: Implement a program using a combination of student-defined and third-party functions to organize the computation.
- 9-12.CT.7: Design or remix a program that utilizes a data structure to maintain changes to related pieces of data.
- 9-12.DL.2: Communicate and work collaboratively with others using digital tools to support individual learning and contribute to the learning of others.

## INSTRUCTIONAL PLAN

List the steps of the lesson, including instructions for the students.

- Do Now: 'What comes to your mind when you hear the word "stack"? In your own words, what is a "data structure"? Take a minute to write down your thoughts, submit it on Google Classroom, and share with the person sitting next to you'
- We review the Do Now as a class, asking students to volunteer to share their answers to the Do Now for a classwide discussion. If students seem hesitant to share, instead pull up

their responses on Google Classroom and highlight some student answers.

- After reviewing the Do Now, go into presenting the slides, making sure to ask the class the questions written in yellow.
- On Slide 9, visit the link for and show pushing and popping onto the stack in real-time. Make sure to call on students for what values should be added to the stack.
- After the demonstration, students should take time to discuss with each other and then write down their answers to the questions on Slide 10 in Google Classroom. Afterwards, we discuss the students' answers as a class.
- After Slide 11, do a live demonstration of the back and forward stacks of a browser. Open a New Tab, and ask students for suggestions on what sites to visit. After visiting each site, make sure to show it in the back button history and how pressing the back button takes us to the sites we visited most recently. Show how the forward button also takes us to the site we visited most recently, but show that it gets cleared out if we visit a new page.
- Then, do a live demonstration of the Browser app that they are going to be building for their lab.
- Next, go over the last few slides of the presentation and let students know their tasks for the day
- Post the lab assignment on Google Classroom and let students work.
- Monitor the check-in list to see when students are ready for a check-in. For the check-in, go over what they have written for the WebStack class to see if the code makes sense for what the class should do. Point out any corrections or minor improvements that students may want to make to their program.
- In addition, go around the class and monitor student progress as well as answer student questions as needed.

### ASSESSMENT CRITERIA

- Part 1 of the lab will be assessed using the following criteria.
  - The StackTester class provided for them passes all 13 tests
  - Their toString() method prints out webpage names in the reverse order (the last thing in the stack should be the first thing printed)
  - Note: Both criteria are assessed in class during the check-in
- Part 2 of the lab will be assessed using the following criteria
  - The displayCurrentPage() method prints out the information of the currentPage variable
  - The goToWebpage() method:
    - Searches the internet ArrayList for the input name
    - adds the currentPage to the browserBack stack
    - clears the browserForward stack
    - sets the currentPage to the desired page if found, otherwise sets it to an error page
  - The goBack() method:
    - Pushes currentPage to the browserForward stack
    - Pops from the browserBack stack to set the new value of currentPage
    - displays the current page
    - explains none of the above is possible if the browserBack stack is empty

- The goForward() method:
  - Pushes currentPage to the browserBack stack
  - Pops from the browserForward stack to set the new value of currentPage
  - displays the current page
  - explains none of the above is possible if the browserForward stack is empty
- The viewBrowserBackAndForward() method:
  - Displays the content of both browserBack and browserForward stacks
  - Displays there is nothing to show if both stacks are empty
- Students are encouraged to check each other's work, and are able to check their own work by using the demo video or reviewing the sample solution (posted after class)

### **MATERIALS / RESOURCES**

- Laptops/Computers
- Repl.it, IntelliJ, or another IDE for coding collaboratively
- SmartBoard, Google Meets, projectors, or other way of presenting slides to students
- Google Classroom for posting assignments and resources
- Other materials needed for the lesson can be found [here](#)