

CS119 – Lab 10
Due Date: April 19

Purpose: The Stack ADT is used throughout computer science. In this lab you will implement this ADT and then use it to check if parentheses are matched properly in a string.

Knowledge: This lab will help you become familiar with the following content knowledge:

- How to implement and use a Stack ADT

Task: Follow the steps in this lab carefully to complete the assignments. Copy the lab10 folder and complete the following assignments. Start by testing out the example functions for queues and make sure you understand them.

Assignment 1:

In a separate file, create a module for the ADT `Stack` and write the operations `empty`, `isEmpty`, `top`, `push`, and `pop`.

Criteria for Success: In the `Example10` file import your module. Create an empty stack and verify that it is empty with the `isEmpty` function. Create a stack by pushing a couple of values on the stack and verify that it is not empty and that the top of the stack is correct. Pop a value off the stack and verify the new top. Check that repeatedly popping off all the values leads to an empty stack.

Assignment 2:

In the `Example10` file implement the function
`matchingParens :: String -> Bool`

This function determines if the parentheses in the string are balanced. For example, `"(()())"` is balanced but `"()()"` is not. We can check for balance by marching through the characters and pushing a left paren onto a stack when encountered and popping a paren off the stack on a right paren of the matching type.

To accomplish this you will want to write a helper function which takes two parameters: your string and a `Stack Char`. This function will return a boolean value on whether or not the string contains matching parens. Then all that `matchingParens` needs to do is call this helper function, passing in an empty stack.

Criteria for Success: Test your function on `"(()())"`, `"()()"`, and `"()"` and verify your results.

Submit all your files in Canvas for grading.