Re: EN.605.202.81.SP21 Data Structures: Lab 2 question

Richard Cost <rcost@jhu.edu>
Fri 2021-03-19 8:58 PM

To: Peter Rasmussen <prasmus3@jhu.edu>

Peter, this solution is fine.

____

From: Peter Rasmussen <prasmus3@jhu.edu>
Sent: Friday, March 19, 2021 7:04 AM
To: Richard Cost
Subject: EN.605.202.81.SP21 Data Structures: Lab 2 question

Hi Dr. Cost,

First, I wanted to let you know my Lab 1 was graded - thank you for facilitating that.

Regarding Lab 2: I want to make sure the implementation I have below doesn't violate any of the restrictions we may have on using Python lists. The below function - which currently doesn't handle edge cases and errors but will - recursively converts a prefix expression (e.g., ['+', 'a', 'b']) into its postfix equivalent. My question is - should I create another class to limit the functionality of the list class (a queue comes to mind)? I use pop (front element), append (push to back) and then also reorder the terms right before the return statement.

```
def convert_prefix_to_postfix(prefix: list)->str:
    postfix = []
    while len(prefix) > 0:
        if prefix[0] in '+-*/$' and len(postfix) == 0:
            postfix.append(prefix.pop(0))
        elif prefix[0] in '+-*/$':
            indent += '   '
            postfix += convert_prefix_to_postfix(prefix)
        else:
            postfix.append(prefix.pop(0))
    op_term = postfix[1] + postfix[2] + postfix[0]
    postfix = [op_term] + postfix[3:]
    return postfix
```

Thanks!

Peter Rasmussen
Data Scientist
202.644.1575