WIA1002/WIB1002 Data Structures

Tutorial: Stack & Problem Solving with Stacks

Question 1

The stack method that returns an element from the stack without removing it is _____________.

a. pop

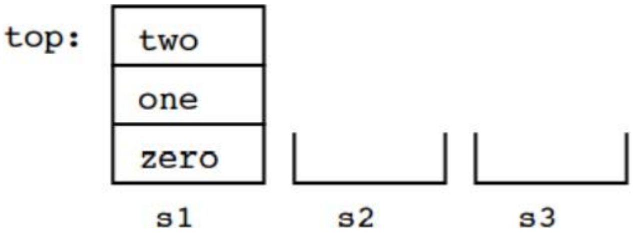
b. push

c. peek

d. spy

Question 2

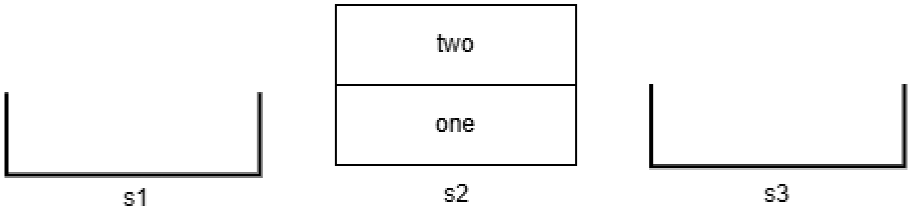
We have three stacks, s1, s2 and s3, that can contain data of type String. Here are their initial contents:



As you can see, initially s2 and s3 are empty. Here is a sequence of operations on the three stacks:

```
s2.push(s1.pop());
s3.push(s1.pop());
s1.pop();
s1.push(s2.pop());
s2.push(s3.pop());
s2.push(s1.pop());
```

Draw the contents of the three stacks after the operations are complete.



Question 3:

Over time, the elements 1, 2, and 3 are pushed onto a stack **in that order**. For each of the following, indicate (yes or no) whether the sequence could be created by popping operations. If yes, list the sequence of push() and pop() operations that produces the sequence.

(a) 1-2-3 (b) 2-3-1 (c) 3-2-1 (d) 1-3-2

(a) Yes

```
push(1)
pop() → 1
push(2)
pop() → 2
push(3)
pop() → 3
```

(b) Yes

```
push(1)
push(2)
pop() → 2
push(3)
pop() → 3
pop() → 1
```

(c) Yes

```
push(1)
push(2)
push(3)
pop() → 3
pop() → 2
pop() → 1
```

(d) Yes

```
push(1)
pop() → 1
push(2)
push(3)
pop() → 3
pop() → 2
```

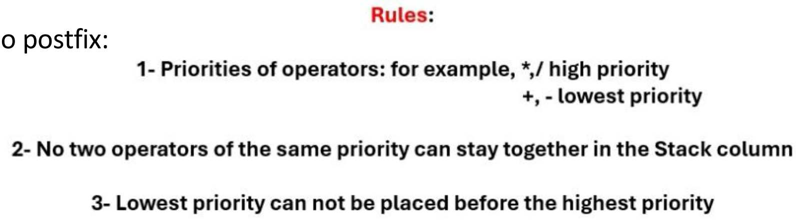
Question 4:

Convert the following infix expressions to postfix:

a) a + b * c

b) a * b – c/d

c) a + (b*c + d)/e



Rules:

1- Priorities of operators: for example, *,/ high priority
+, - lowest priority

2- No two operators of the same priority can stay together in the Stack column

3- Lowest priority can not be placed before the highest priority

a) a b c * +

b) a b * c d / -

c) a b c * d + e / +

Question 5:

Write the following expressions in infix form:

a) a b + c *

b) a b c + *

a) (a + b) * c

b) a * (b + c)

Question 6:

Which of the following is an application of stack?

A. finding factorial

B. tower of Hanoi

C. infix to postfix

D. all of the above

- **Finding factorial:** Can be implemented using recursion and recursion internally uses a stack (call stack)
- **Tower of Hanoi**: Typically solved recursively so again it relies on the stack structure during execution
- **Infix to postfix**: Requires an explicit stack to hold operators and manage precedence during conversion