

Erece, Kian Relioe B.

Table

```
1 usage new *
class Stack:
    new *
    def __init__(self):
        self.stack = []

2 usages new *
    def size(self):
        return len(self.stack) # Fixed: Changed self-stack to self.stack

3 usages new *
    def push(self, item):
        self.stack.append(item)

4 usages new *
    def pop(self):
        if not self.is_empty():
            return self.stack.pop()
        else:
            return "Error: Stack is empty!"

5 usages new *
    def current_items(self):
        return self.stack

4 usages new *
    def is_empty(self):
        return len(self.stack) == 0

1 usage new *
    def top(self):
        if not self.is_empty():
            return self.stack[-1]
        else:
            return "The stack is empty"

stack = Stack()
```

```
1 #Table_code
2 stack.push(5)
3 print("Current stack: " + str(stack.current_items()))
4 stack.push(3)
5 print("Current stack: " + str(stack.current_items()))
6 print(f"Current size: {stack.size()}")
7 print(f"Popped value: {stack.pop()}")
8 print(f"Is stack empty? {stack.is_empty()}")
9 print(f"Popped value: {stack.pop()}")
10 print(f"Is stack empty? {stack.is_empty()}")
11 print(f"Popped value: {stack.pop()}")
12 stack.push(7)
13 print("Current stack: " + str(stack.current_items()))
14 stack.push(9)
15 print("Current stack: " + str(stack.current_items()))
16 print(f"Top value: {stack.top()}")
17 stack.push(4)
18 print("Current stack: " + str(stack.current_items()))
19 print(f"Current size: {stack.size()}")
20 print(f"Popped value: {stack.pop()}")
21 stack.push(6)
22 print("Current stack: " + str(stack.current_items()))
23 stack.push(8)
24 print("Current stack: " + str(stack.current_items()))
25 print(f"Popped value: {stack.pop()}")
26 print("Current stack: " + str(stack.current_items()))
```

```

"C:\Program Files\Python312\python.exe" Z:\DSAL601-IDB2\Activity1_ereceMidt.py
Current stack: [5]
Current stack: [5, 3]
Current size: 2
Popped value: 3
Is stack empty? False
Popped value: 5
Is stack empty? True
Popped value: Error: Stack is empty!
Current stack: [7]
Current stack: [7, 9]
Top value: 9
Current stack: [7, 9, 4]
Current size: 3
Popped value: 4
Current stack: [7, 9, 6]
Current stack: [7, 9, 6, 8]
Popped value: 8
Current stack: [7, 9, 6]

```

What values are returned during the following series of stack operations, if executed upon an initially empty stack? **push(5), push(3), pop(), push(2), push(8), pop(), pop(), push(9), push(1), pop(), push(7), push(6), pop(), pop(), push(4), pop(), pop()**.

```

44 #Returned values code
45 stack.push(5)
46 stack.push(3)
47
48 print(f"Popped value: {stack.pop()}")
49
50 stack.push(2)
51 stack.push(8)
52
53 print(f"Popped value: {stack.pop()}")
54 print(f"Popped value: {stack.pop()}")
55
56 stack.push(9)
57 stack.push(1)
58
59 print(f"Popped value: {stack.pop()}")
60
61 stack.push(7)
62 stack.push(6)
63
64 print(f"Popped value: {stack.pop()}")
65 print(f"Popped value: {stack.pop()}")
66
67 stack.push(4)
68
69 print(f"Popped value: {stack.pop()}")
70 print(f"Popped value: {stack.pop()}")

```

```

Popped value: 3
Popped value: 8
Popped value: 2
Popped value: 1
Popped value: 6
Popped value: 7
Popped value: 4
Popped value: 9

```

