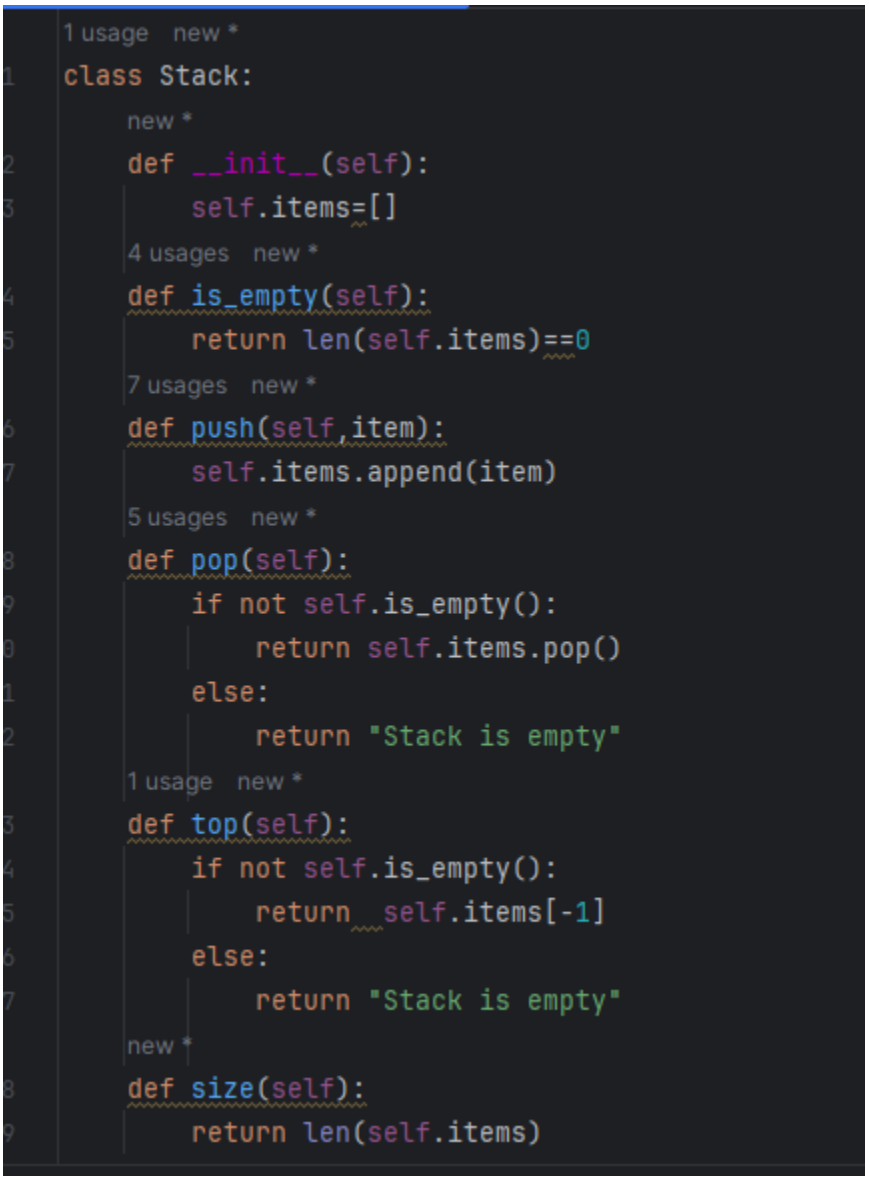
Richmond_Al P. Broqueza 09/27/2024

IDB2 DSALGO1

1.



```
1 usage  new *
class Stack:
    new *
    def __init__(self):
        self.items=[]
    4 usages  new *
    def is_empty(self):
        return len(self.items)==0
    7 usages  new *
    def push(self,item):
        self.items.append(item)
    5 usages  new *
    def pop(self):
        if not self.is_empty():
            return self.items.pop()
        else:
            return "Stack is empty"
    1 usage  new *
    def top(self):
        if not self.is_empty():
            return  self.items[-1]
        else:
            return "Stack is empty"
    new *
    def size(self):
        return len(self.items)
```

```
S = Stack()
S.push(5)
print("Pushed: 5")
S.push(3)
print("Pushed: 3")
print("Length:",len(S.items))
print("Popped Element:",S.pop())
print ("Stack empty?",S.is_empty())
print("Popped Element",S.pop())
print ("Stack empty?",S.is_empty())
print("Popped Element",S.pop())
S.push(7)
print("Pushed: 7")
S.push(9)
print("Pushed: 9")
print("Top Element:",S.top())
S.push(4)
print("Pushed: 4")
print("Popped Element",S.pop())
print("Length: ",len(S.items))
S.push(6)
print("Pushed: 6")
S.push(8)
print("Pushed: 8")
print("Popped Element",S.pop())
print(S.items)
```

OUTPUT:

```
Pushed: 5
Pushed: 3
Length: 2
Popped Element: 3
Stack empty? False
Popped Element 5
Stack empty? True
Popped Element Stack is empty
Pushed: 7
Pushed: 9
```

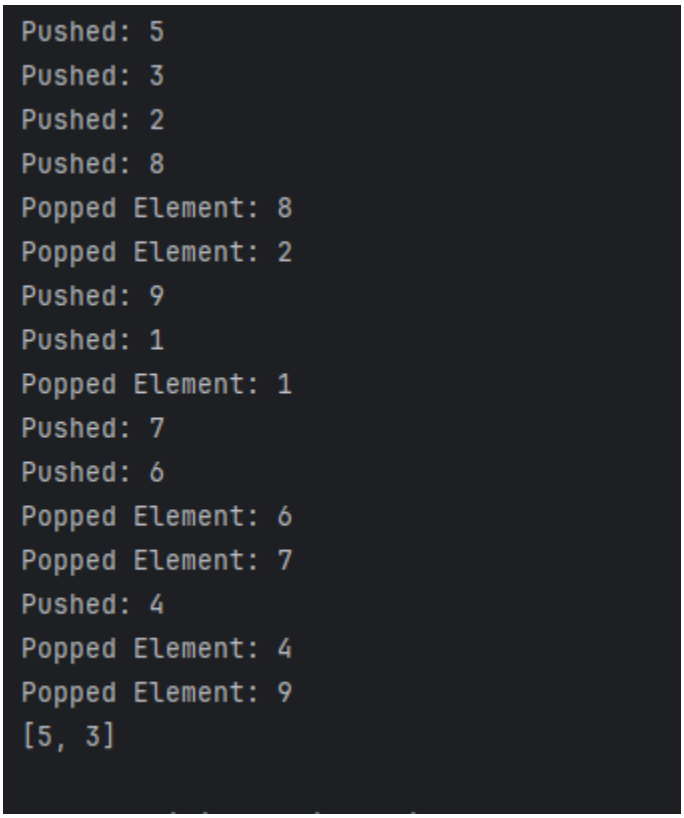
```
Top Element: 9
Pushed: 4
Popped Element 4
Length:  2
Pushed: 6
Pushed: 8
Popped Element 8
[7, 9, 6]
```

2.

```
1 usage  new *
class Stack:
    new *
    def __init__(self):
        self.items=[]
    4 usages  new *
    def is_empty(self):
        return len(self.items)==0
    7 usages  new *
    def push(self,item):
        self.items.append(item)
    5 usages  new *
    def pop(self):
        if not self.is_empty():
            return self.items.pop()
        else:
            return "Stack is empty"
    1 usage  new *
    def top(self):
        if not self.is_empty():
            return  self.items[-1]
        else:
            return "Stack is empty"
    new *
    def size(self):
        return len(self.items)
```

```
S=Stack()
print()
print()
S.push(5)
print("Pushed: 5" )
S.push(3)
print("Pushed: 3" )
S.push(2)
print("Pushed: 2" )
S.push(8)
print("Pushed: 8" )
print("Popped Element:",S.pop())
print("Popped Element:",S.pop())
S.push(9)
print("Pushed: 9" )
S.push(1)
print("Pushed: 1" )
print("Popped Element:",S.pop())
S.push(7)
print("Pushed: 7" )
S.push(6)
print("Pushed: 6" )
print("Popped Element:",S.pop())
print("Popped Element:",S.pop())
S.push(4)
print("Pushed: 4" )
print("Popped Element:",S.pop())
print("Popped Element:",S.pop())
print (S.items)
```

OUTPUT:


Pushed: 5
Pushed: 3
Pushed: 2
Pushed: 8
Popped Element: 8
Popped Element: 2
Pushed: 9
Pushed: 1
Popped Element: 1
Pushed: 7
Pushed: 6
Popped Element: 6
Popped Element: 7
Pushed: 4
Popped Element: 4
Popped Element: 9
[5, 3]