Activity #1 (Midterm) Laboratory Activity

Create an implementation of the **Stack Data Structure:**

As a guide, you can use the following description of the **Stack Data Structure**.

Stacks are the simplest of all data structures, yet they are also among the most important. They are used in a host of different applications, and as a tool for many more sophisticated data structures and algorithms. Formally, a stack is an abstract data type (ADT) such that an instance S supports the following two methods:

S.push(e):	Add element e to the top of stack S.
S.pop():	Remove and return the top element from the stack S; an error occurs if the stack is empty.

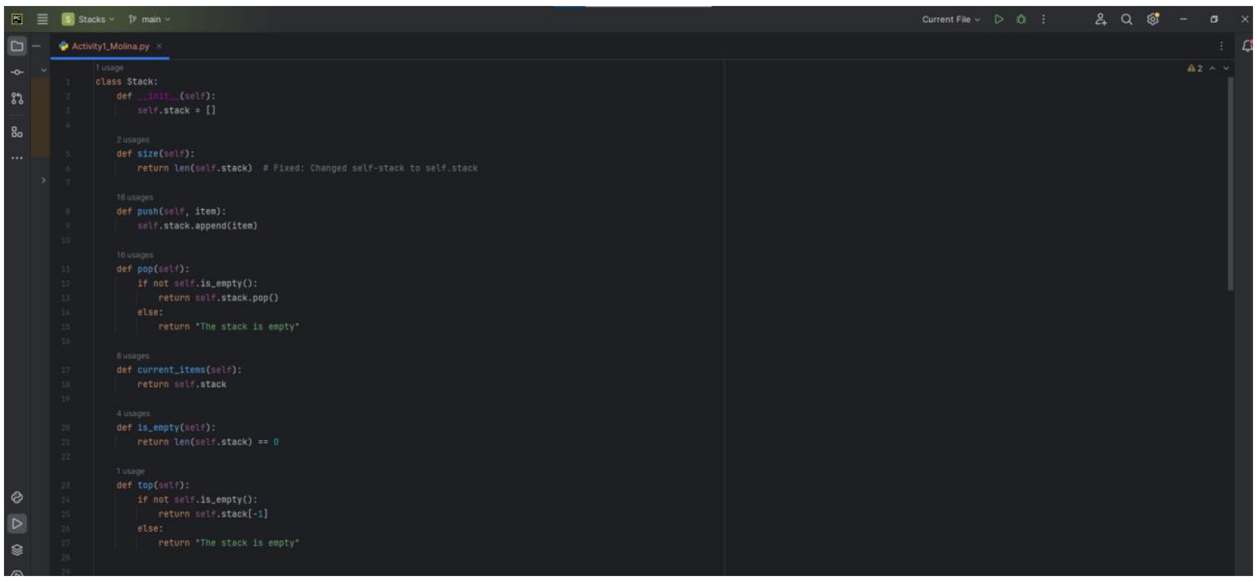
Additionally, let us define the following accessor methods for convenience:

S.top():	Return a reference to the top element of stack S, without removing it; an error occurs if the stack is empty.
S.is_empty():	Return True if stack S does not contain any elements.
len(S):	Return the number of elements in stack S; in Python, we implement this with the special method `__len__`.

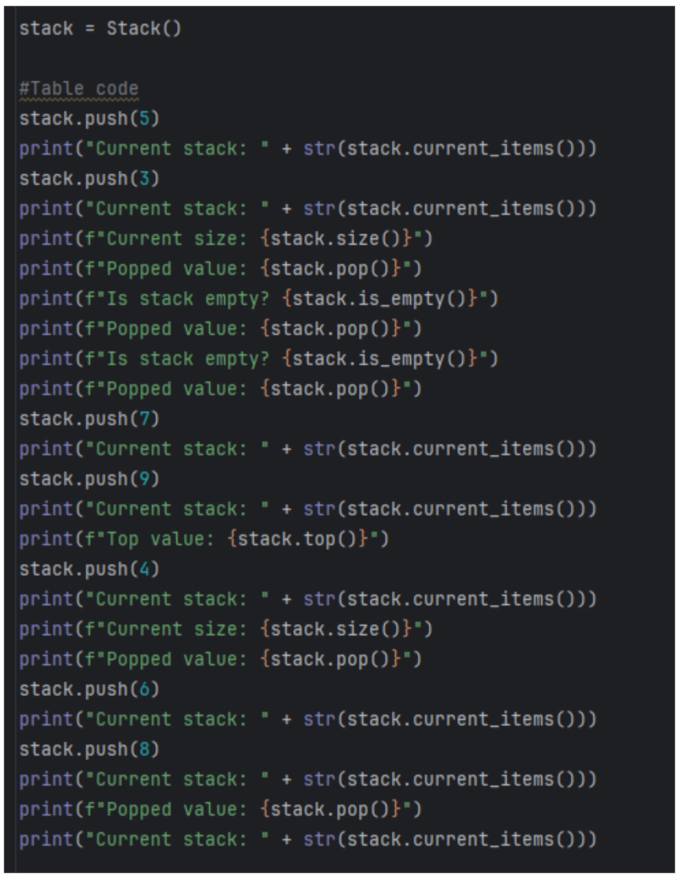
By convention, we assume that a newly created stack is empty, and that there is no a priori bound on the capacity of the stack. Elements added to the stack can have arbitrary type.

Next, simulate the **Stack Data Structure** using the table below:

Operation
S.push(5)
S.push(3)
len(S)
S.pop()
S.is_empty()
S.pop()
S.is_empty()
S.pop()
S.push(7)
S.push(9)
S.top()
S.push(4)
len(S)
S.pop()
S.push(6)
S.push(8)
S.pop()



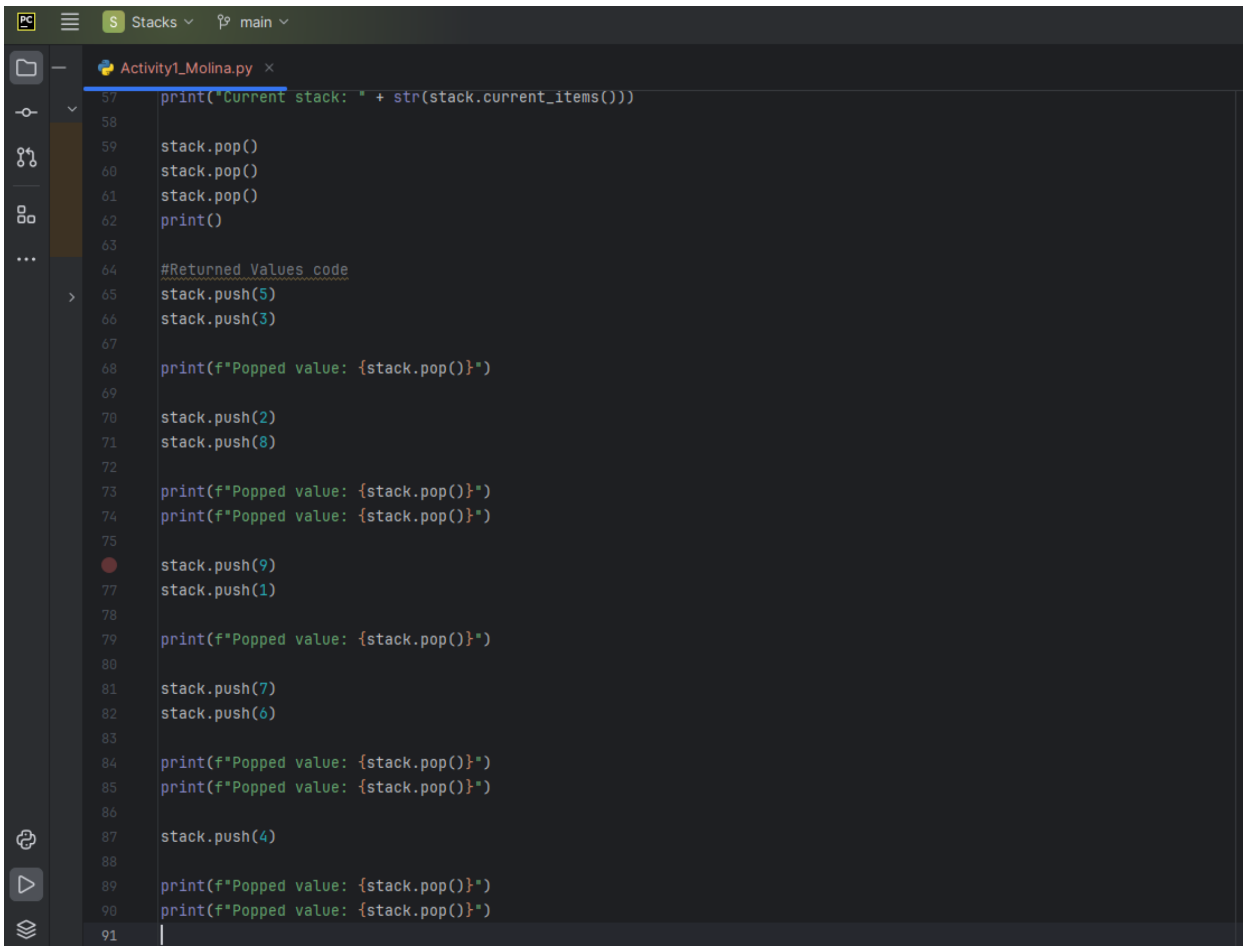
```python
class Stack:
    def __init__(self):
        self.stack = []

    def size(self):
        return len(self.stack)  # Fixed: Changed self-stack to self.stack

    def push(self, item):
        self.stack.append(item)

    def pop(self):
        if not self.is_empty():
            return self.stack.pop()
        else:
            return "The stack is empty"

    def current_items(self):
        return self.stack

    def is_empty(self):
        return len(self.stack) == 0

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



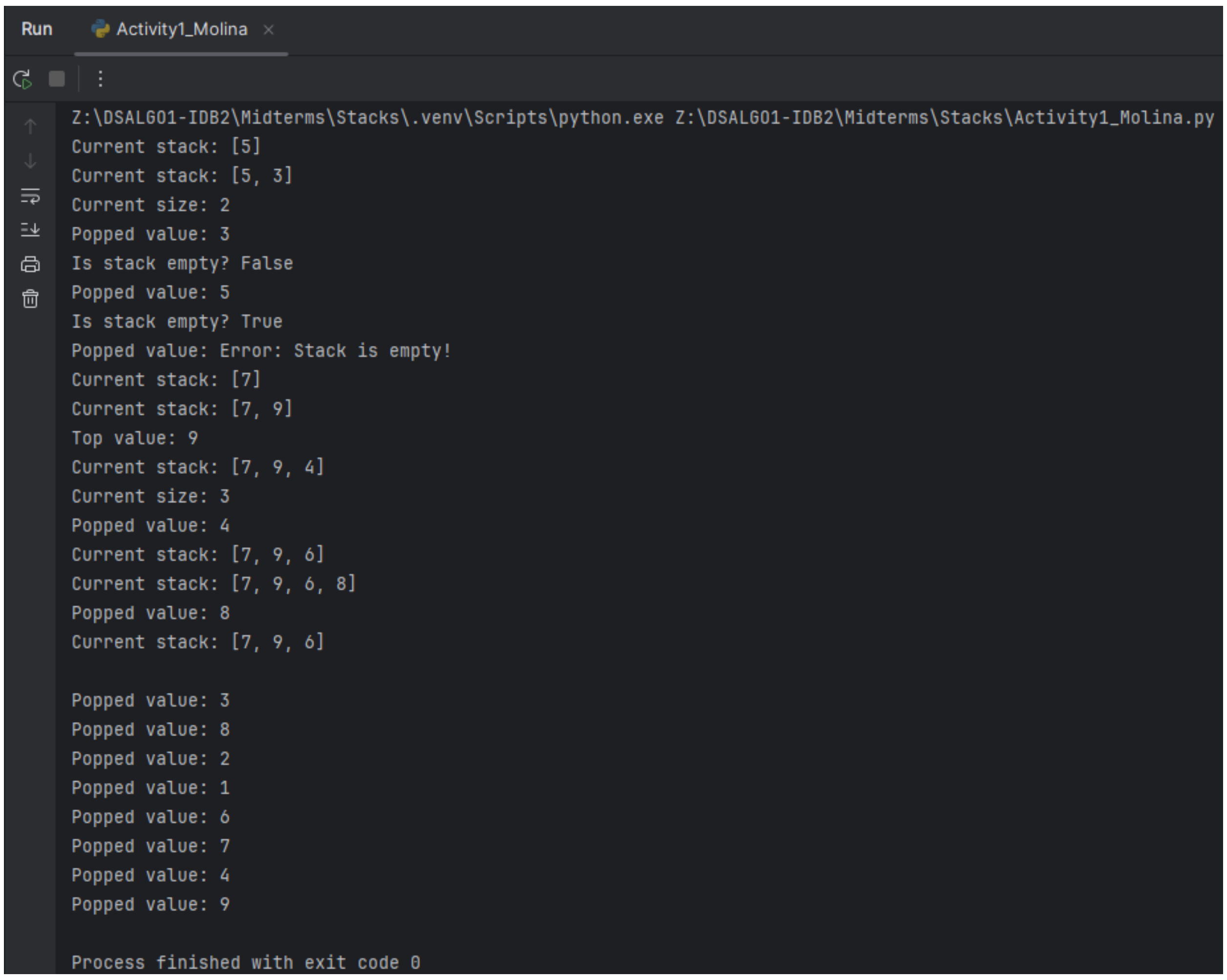
```
stack = Stack()

#Table code
stack.push(5)
print("Current stack: " + str(stack.current_items()))
stack.push(3)
print("Current stack: " + str(stack.current_items()))
print(f"Current size: {stack.size()}")
print(f"Popped value: {stack.pop()}")
print(f"Is stack empty? {stack.is_empty()}")
print(f"Popped value: {stack.pop()}")
print(f"Is stack empty? {stack.is_empty()}")
print(f"Popped value: {stack.pop()}")
stack.push(7)
print("Current stack: " + str(stack.current_items()))
stack.push(9)
print("Current stack: " + str(stack.current_items()))
print(f"Top value: {stack.top()}")
stack.push(4)
print("Current stack: " + str(stack.current_items()))
print(f"Current size: {stack.size()}")
print(f"Popped value: {stack.pop()}")
stack.push(6)
print("Current stack: " + str(stack.current_items()))
stack.push(8)
print("Current stack: " + str(stack.current_items()))
print(f"Popped value: {stack.pop()}")
print("Current stack: " + str(stack.current_items()))
```

Lastly, simulate the following operations to answer the question listed.

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()**.



```
print("Current stack: " + str(stack.current_items()))

stack.pop()
stack.pop()
stack.pop()
print()

#Returned Values code
stack.push(5)
stack.push(3)

print(f"Popped value: {stack.pop()}")

stack.push(2)
stack.push(8)

print(f"Popped value: {stack.pop()}")
print(f"Popped value: {stack.pop()}")

stack.push(9)
stack.push(1)

print(f"Popped value: {stack.pop()}")

stack.push(7)
stack.push(6)

print(f"Popped value: {stack.pop()}")
print(f"Popped value: {stack.pop()}")

stack.push(4)

print(f"Popped value: {stack.pop()}")
print(f"Popped value: {stack.pop()}")
```



```
Z:\DSALG01-IDB2\Midterms\Stacks\.venv\Scripts\python.exe Z:\DSALG01-IDB2\Midterms\Stacks\Activity1_Molina.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]

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

Process finished with exit code 0
```