Richard Hayes Crowley

03/23/2021

CSC\_157\_Lab\_014

**SOURCE CODE:**

*# Richard Hayes Crowley*

*# CSC\_157\_Lab\_14*

*import* stack

*# a stack object is declared*

s = stack.Stack()

*# verify if the stack is empty*

print(s.isEmpty())

*# some items are pushed onto the stack*

s.push(10)

s.push("james")

*# determine the top element of the stack*

print(s.peek())

*# an item is placed onto the stack*

s.push(True)

*# the current stack size is requested*

print(s.size())

*# an item is placed onto the stack*

s.push("luke")

*# the status of the stack is verified*

print(s.isEmpty())

*# an item is placed onto the stack*

s.push(20)

*# some items are removed from the stack*

print(s.pop())

print(s.pop())

*# an item is placed onto the stack*

s.push("katherine")

*# the current stack size is requested*

print(s.size())

*# item 20 is added to the stack*

s.push(20)

*# item " clerk " is pushed on the stack*

s.push("clerk")

*# item 15.00 is added to the stack*

s.push(15.00)

*# item 50 is pushed on the stack*

s.push(50)

*# item " manager " is added to the stack*

s.push("manager")

*# item 25.00 is added to the stack*

s.push(25.00)

*# the stack size is requested*

print(s.size())

*# an item is popped from the stack*

s.pop()

*# the stack size is requested*

print(s.size())

*# an item is popped from the stack*

s.pop()

*# item " assistant " is added to the stack*

s.push("assistant")

*# an item is popped from the stack*

s.pop()

*# item " clerk " is added to the stack*

s.push("clerk")

*# two items are popped from the stack*

s.pop()

s.pop()

*# a request is made to verify that is stack is not empty*

print(s.isEmpty())

*# the stack size is requested*

print(s.size())

*# a request is made to view the top - most element in the stack*

print(s.peek())

*# the stack size is requested*

print(s.size())

*# Going to cause an overflow*

s.push("one")

s.push("two")

s.push("three")

print("Attempting to push 'four' will throw an exception because we are capacity")

s.push("four")

*# Stack overflow! Please pop an item before pushing*

print(f"Return from popping out the last item, should be 'three': {s.pop()}")

s.push("four")

*# Four will now display on top of stack*

print(

f"Pushed 'four' after popping 'three', peeking at top of stack returns: {s.peek()} ")

**OUTPUT:**

Text

Description automatically generated