

## QUIZ CONCEPTUAL

In which of the following scenarios, would a stack **NOT** be the ideal data structure to model the situation?

A line in a grocery store.



Correct! A line in a grocery models a First In, First Out structure in that the first person to enter the line will also be the first person to leave the line. A stack should model a Last In, First Out structur... [Show more](#)

Dirty plates in a sink.

Using the back and forward arrows in an internet browser to navigate through pages.

If I “peeked” at the following stack, what value would be returned, and what would the new stack look like from top to bottom?

```
Stack:
|55|
|60|
|50|
----
Top Value: 55
Bottom Value: 50
```

Return Value: 55, Stack: [55, 60, 50]



Correct! “Peeking” into the stack only returns the value from the top of the stack. “Peeking” does NOT remove any value from the stack.

In what case would “stack overflow” occur?

Popping data from an empty stack.

Pushing data onto an already full stack.



Correct! If the stack is full, then there is no room in the stack. Therefore, “pushing” data onto the stack will result in stack overflow.

Popping data from a full stack.

Pushing data onto an empty stack.

If I “popped” a value from the following stack, what value would be returned, and what would the new stack look like from top to bottom?

```
Stack:
|55|
|60|
|50|
----
Top Value: 55
Bottom Value: 50
```

Return Value: 55, Stack: [60, 50]



Correct! “Popping” from the stack removes and returns the value from the top of the stack.

If the following stack was implemented using a linked list (with the head node of the linked list representing the top of the stack), what would the linked list look like if a 40 pound gym plate was “pushed” onto the stack?

```
Stack:
|55|
|60|
|50|
----
Top Value: 55
Bottom Value: 50
Value to Add: 40
```

40 → 55 → 60

55 → 60 → 50 → 40

60 → 50 → 40

40 → 55 → 60 → 50



Correct! Because stacks follow a Last In First Out structure, the value “pushed” onto the stack will become the head node in a linked list.