

CS 314: Data Structures

April 5, 2021

Sam's Section

Slides available at: www.cs.utexas.edu/~slaberge

Sets Assignment

Look Ahead

- Another partner assignment, post on Piazza if you need a partner!
- As always, start early!
- Also, start thinking about Exam 2 coming up next week
 - Thursday, April 15th
 - Get started studying soon!

Sets Assignment

Style Suggestions

- Be careful of unsafe casts, you shouldn't have any warnings in your code!
 - You may need to use a wildcard generic type `<?>`
- Make sure to have comments for each method
 - (For `ISet` methods, you can copy the comment from the interface!)
- Implement as many methods as you can for `AbstractSet`

Section Problem

Queues

Queues

Help Hours Queue

0: Alice

1: Bob

2: Charlie

Queues

Help Hours Queue

Front 

0: Alice

1: Bob

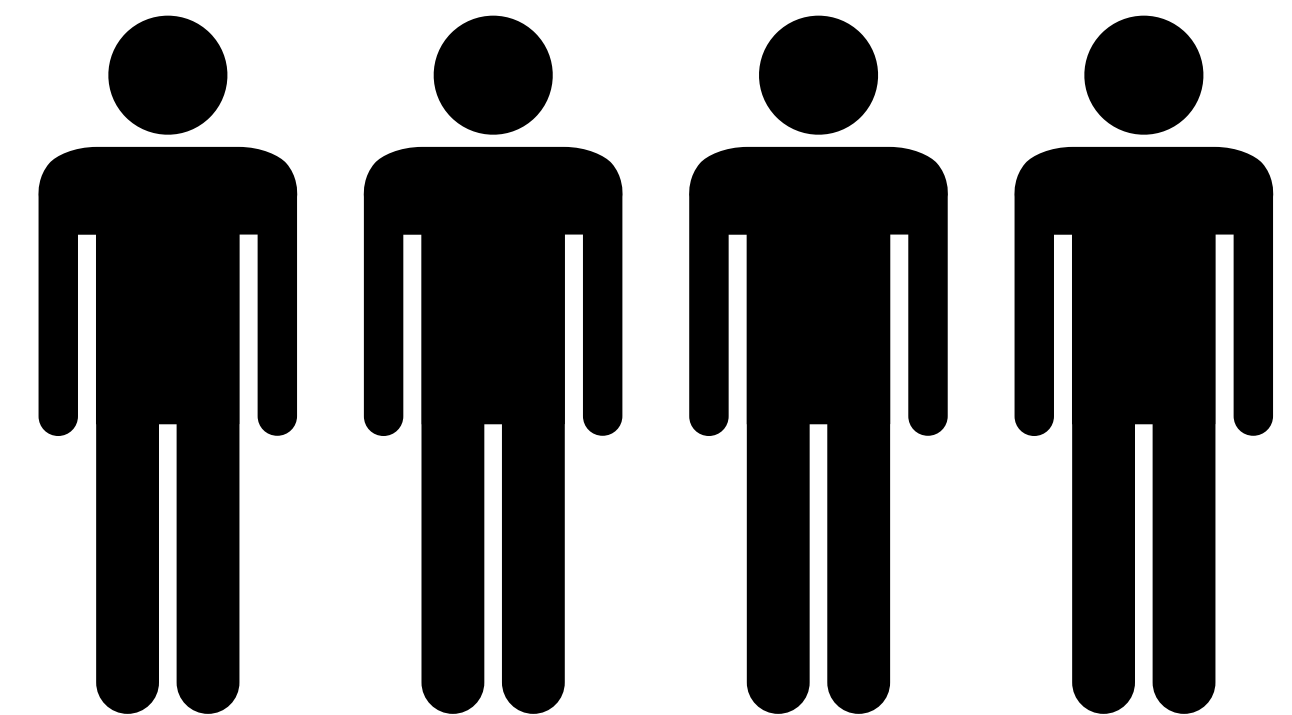
2: Charlie

Back 

Queues

Section Problem

- Another word for “line”
 - As in: “get in line,” “back of the line”, “line up”, etc.
- FIFO: First In, First Out
- Queues are “fair” - next one to be “helped” is whoever’s been waiting the longest



Queues

Queue Operations

- `enqueue(E elem)`: Add an element to the back of the queue

Join the Queue

Name

Email

EID

Short Problem Description

[Join the Queue!](#)

`enqueue("Dave")`



Help Hours Queue

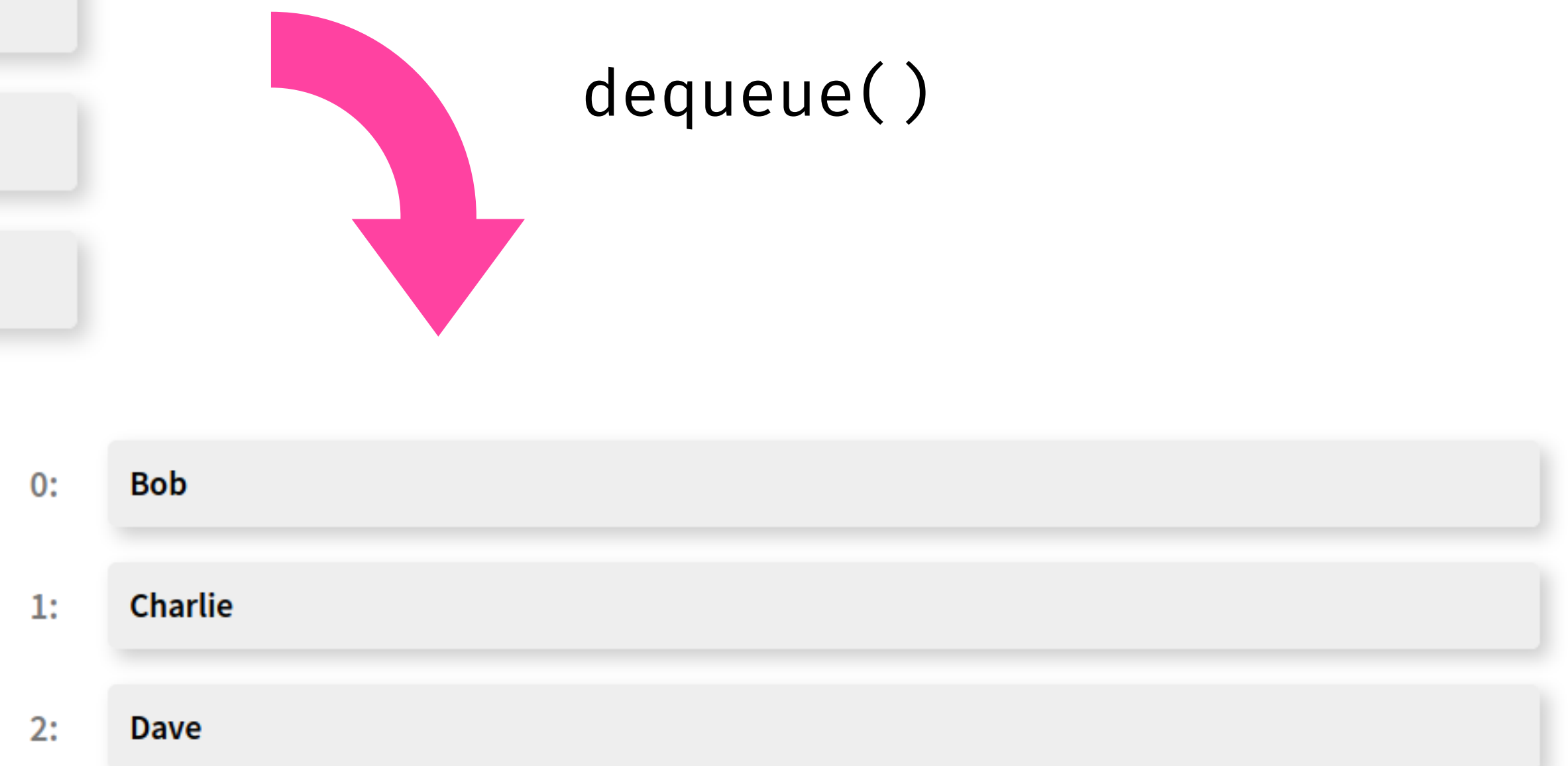
0:	Alice
1:	Bob
2:	Charlie
3:	Dave



Queues

Queue Operations

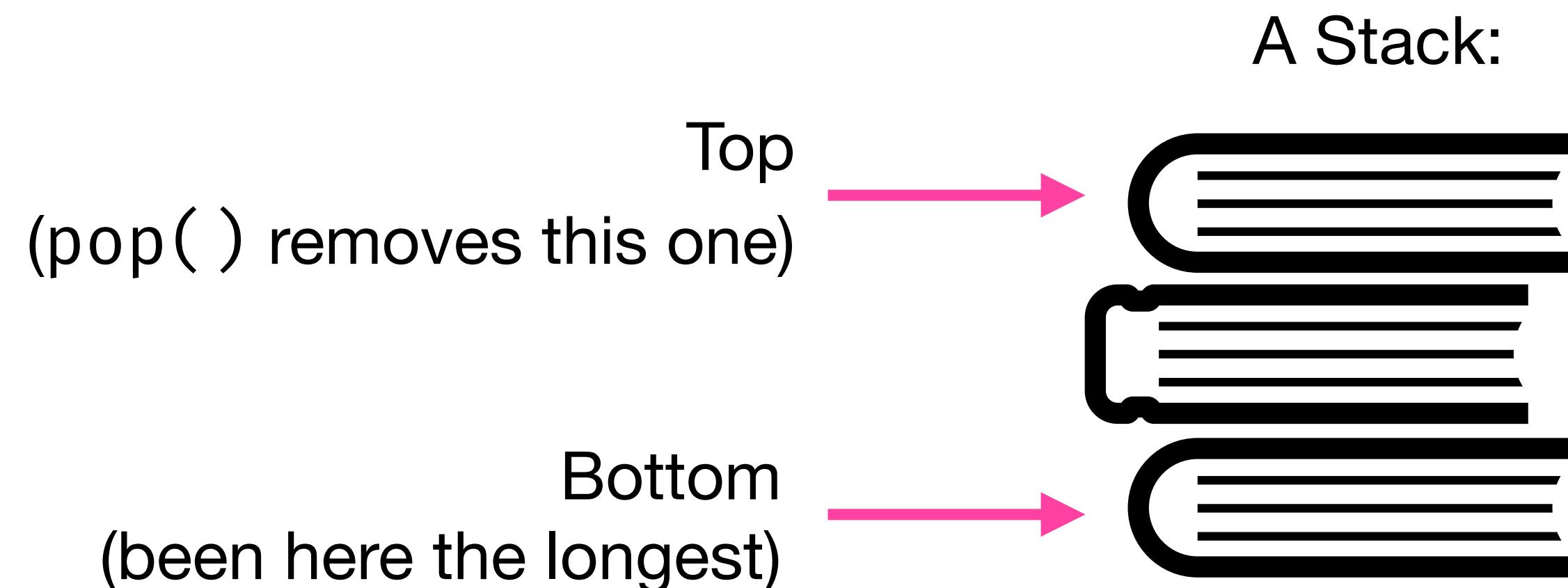
- `dequeue()`: Removes element at the front of the queue



Queues

Compared to Stacks

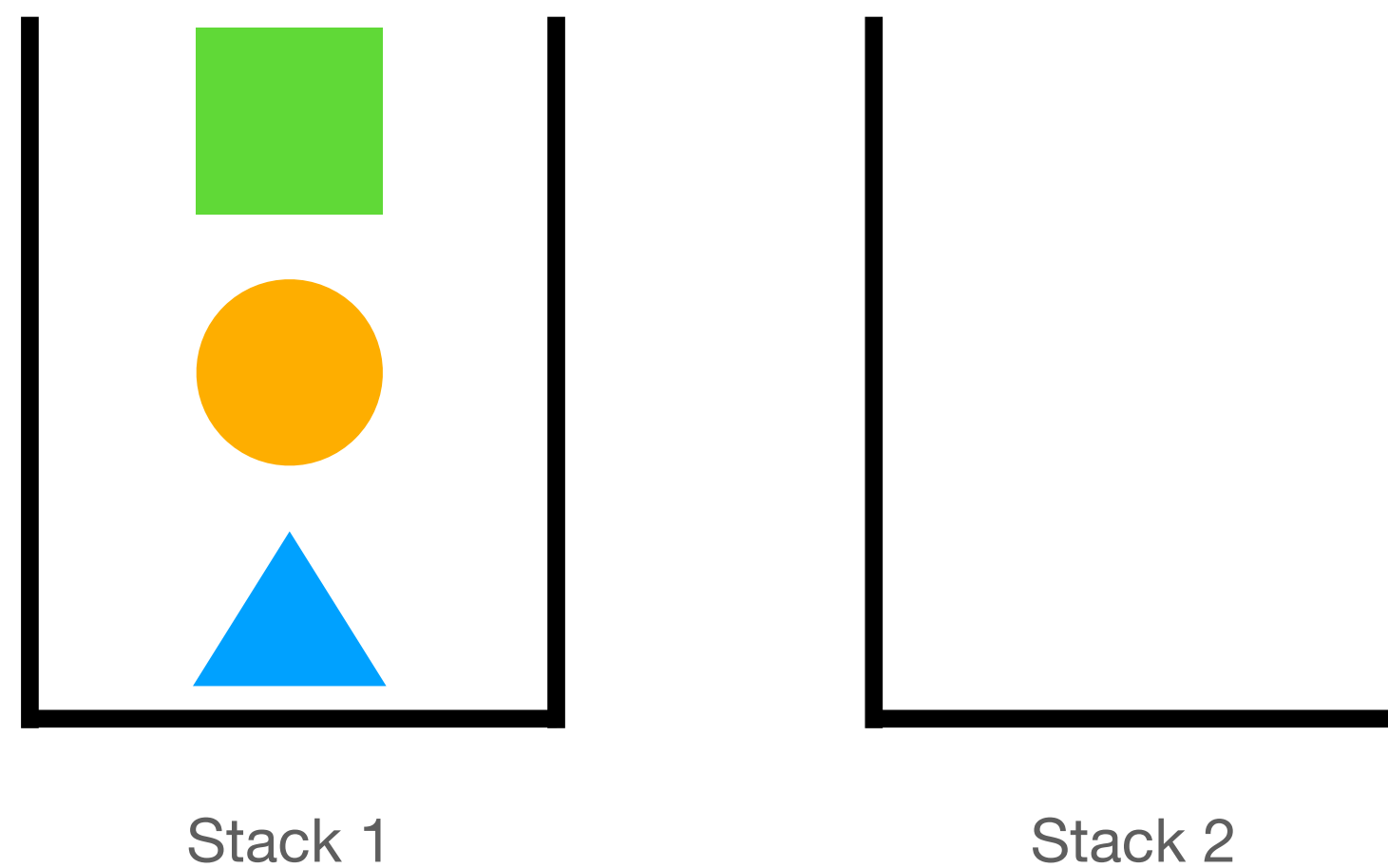
- Queues are FIFO: First In, First Out
 - `dequeue()`: Remove the item that's been in the queue the longest
- Stacks are LIFO: Last In, First Out
 - `pop()`: Remove the item on top (the one that's been there the *least* amount of time)



Queues

Section Problem

- Implement a Queue using two Stacks
- Recall how you can “reverse” a stack:

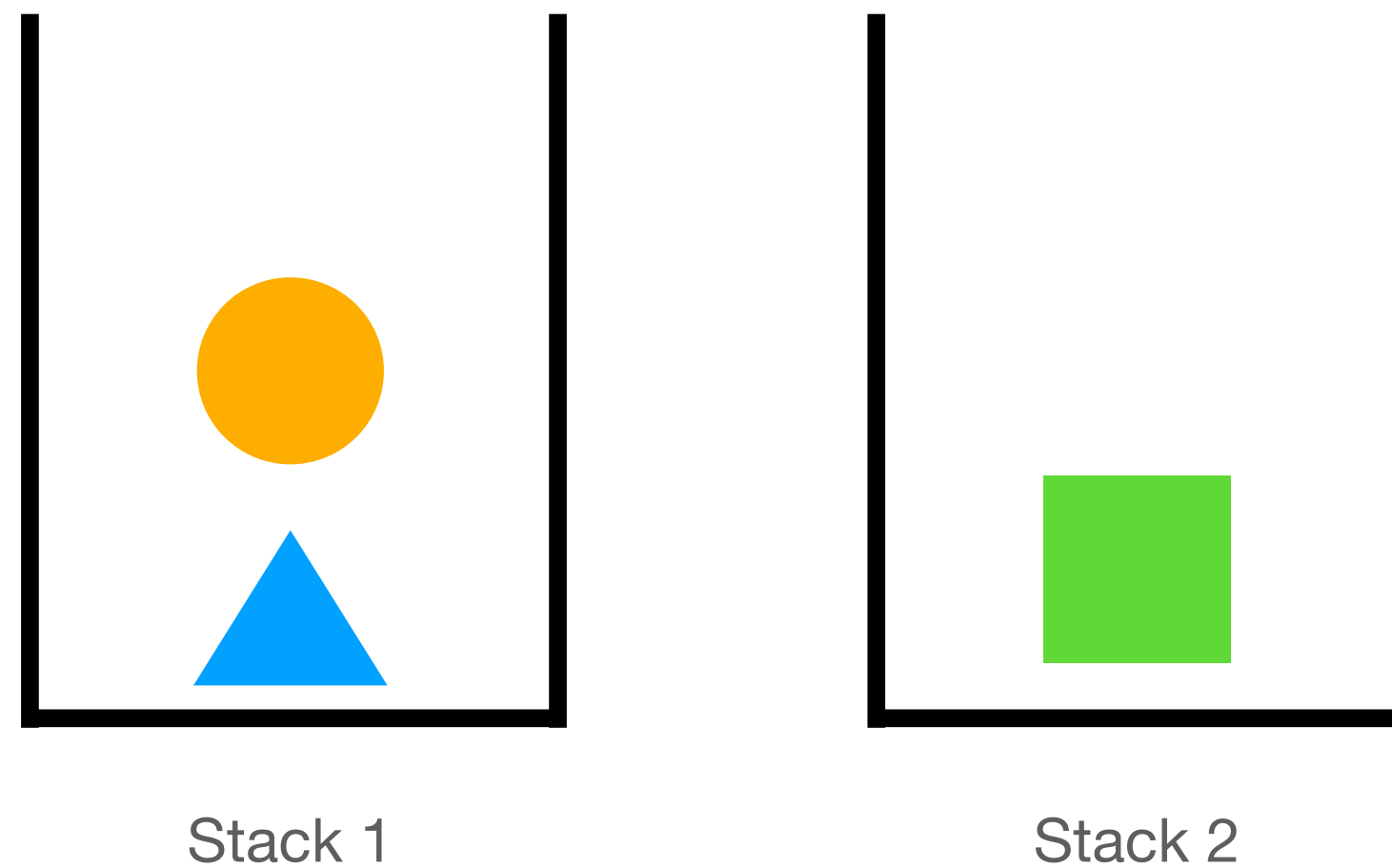


Queues

Section Problem

- Implement a Queue using two Stacks
- Recall how you can “reverse” a stack:

```
stack2.push( stack1.pop() )
```

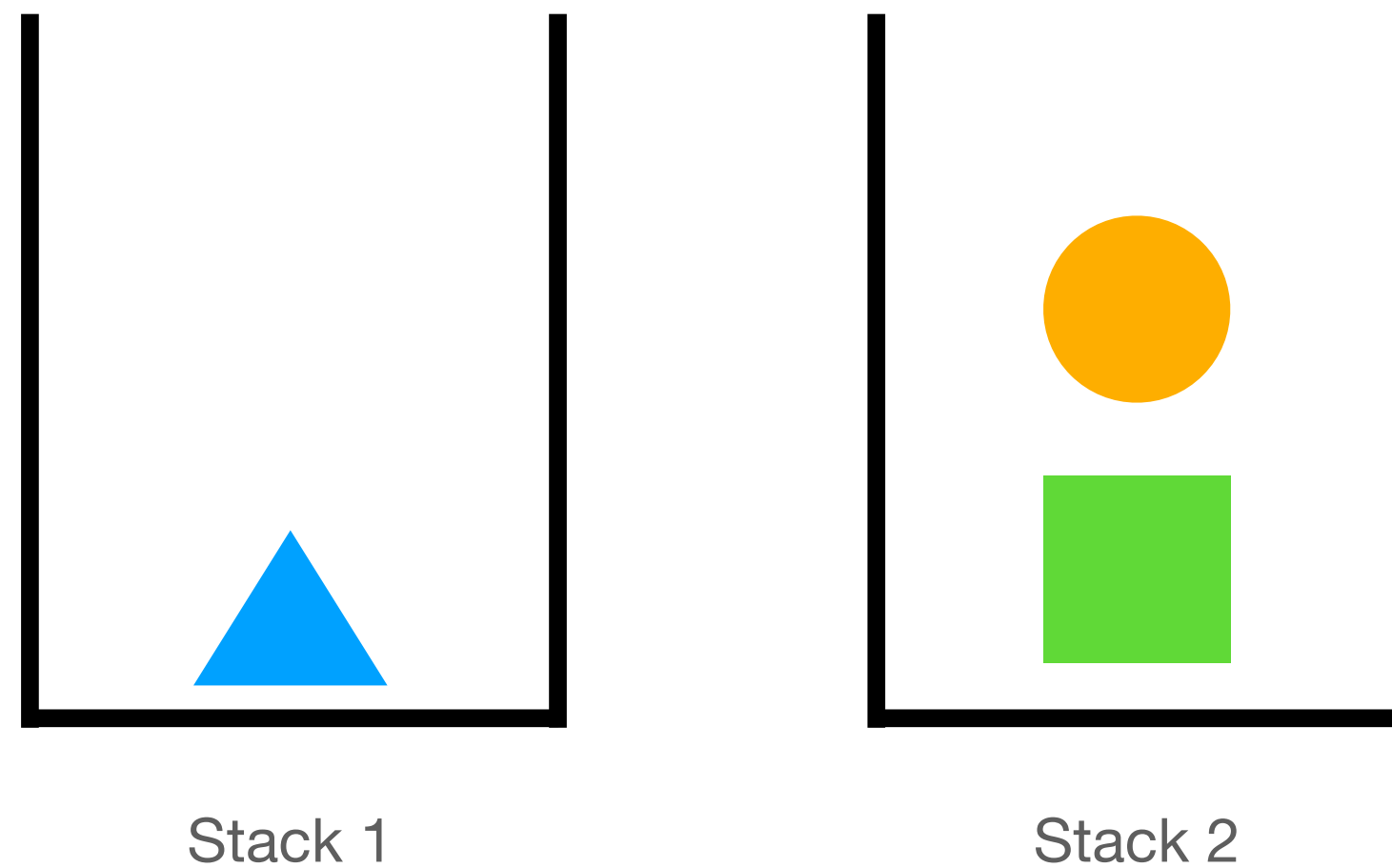


Queues

Section Problem

- Implement a Queue using two Stacks
- Recall how you can “reverse” a stack:

```
stack2.push( stack1.pop() )
```

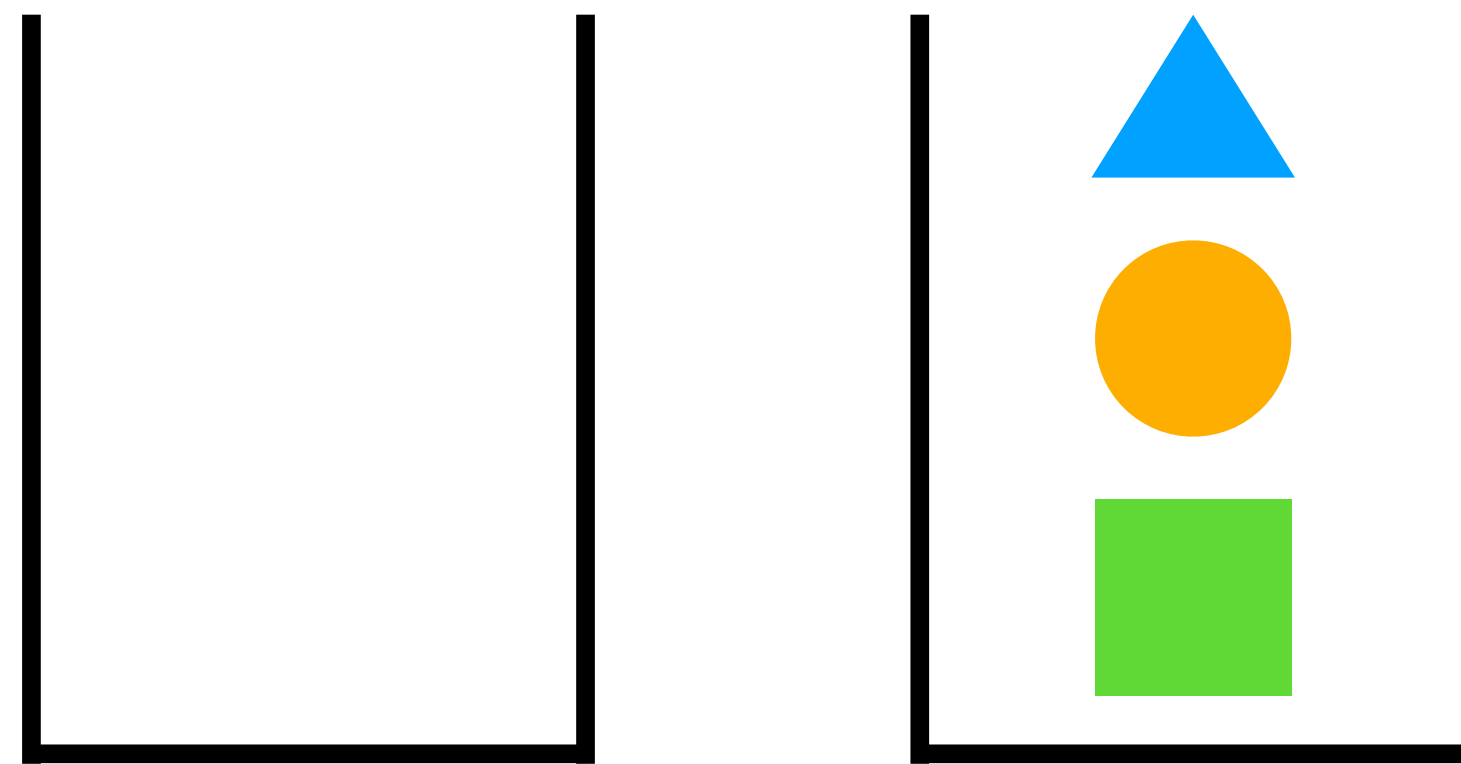


Queues

Section Problem

- Implement a Queue using two Stacks
- Recall how you can “reverse” a stack:

```
stack2.push( stack1.pop() )
```



Stack 1

Stack 2

Queues

Section Problem

- Implement a Queue using two Stacks
- Recall how you can “reverse” a stack:

```
stack2.push( stack1.pop() )
```

