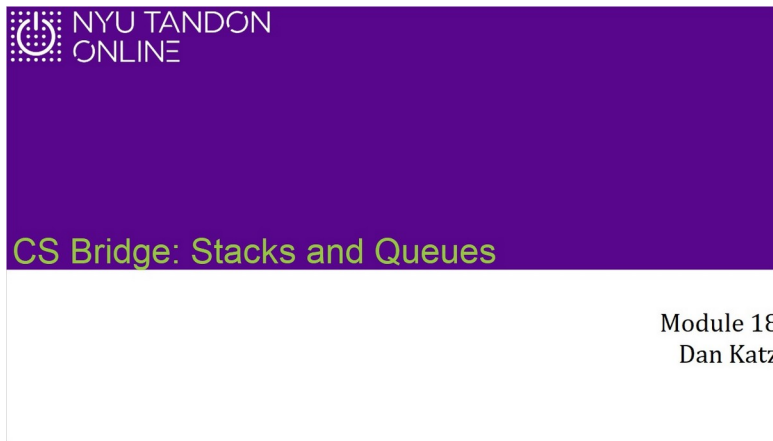


CS Bridge Module 18 Stacks & Queues

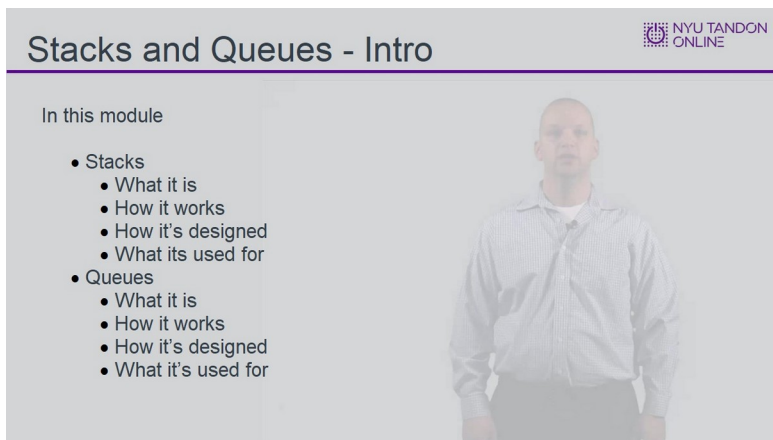
1. Stacks and Queues

1.1 Title Slide



Notes:


1.2 Stacks and Queues - Intro



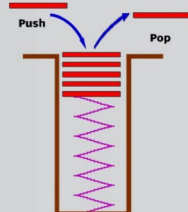
Notes:


1.3 Stack

Stack




- Last in, First out (or First in, Last out) FILO or LIFO
 - The last item inserted is the first removed
- Accessor functions are usually “push,” “pop” and “top”
- Supplemental functions
 - clear
 - isEmpty
 - size



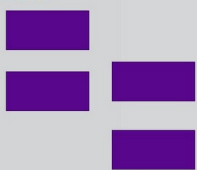


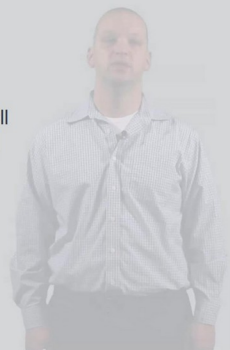
1.4 Stack, How It Works

Stack, How It Works



- Push simply adds to the front
- Pop simply removes from the front
- Supplemental functions need to be provided to all as applicable
 - clear
 - isEmpty
 - size





1.5 Stack Storage

Stack Storage

NYU TANDON ONLINE

Comparison of storage methods

	List	Array
push	$O(1)$	$O(1)/O(N)$
pop	$O(1)$	$O(1)$
top	$O(1)$	$O(1)$
clear	$O(N)$	$O(1)$
isEmpty	$O(1)$	$O(1)$
size	$O(1)$	$O(1)$

1.6 Stack Storage - Continued

Stack Storage - Continued

NYU TANDON ONLINE

- Since we recognize that pop will happen more often than clear, the list makes considerably more sense to use than the array.
- We will use the built-in STL list class.
 - Push will call push_front
 - Pop will call pop_front
 - Other functions will be mapped to equivalent STL functions

1.7 Stack Code

Stack Code

NYU TANDON ONLINE

```
#include <iostream>
#include <list>

using namespace std;

template<class T>
class Stack{
    list<T> data;
public:
    void push(T newItem){ data.push_front(newItem); }
    T pop(){ return data.pop_front(); }
    T top()const { return *data.begin(); }
    bool isEmpty()const{ return data.size() == 0; }
    int size() const{ return data.size(); }
    void clear(){ data.clear(); }
};
```

1.8 Stack, What Is It Used For

Stack, What Is It Used For



- Compilers
 - Used for pattern matching, ({ matches })
 - When an open is encountered, it is pushed
 - When a close is encountered, it is compared to the open at the top and, if a match, the top is popped.
- Math Infix to post-fix conversion
- Math post-fix to value evaluation



1.9 Stack, What Is It Used For

Stack, What Is It Used For



```
int main() {  
    int arr[] = { 2, 3, 4, 5 };  
    cout << arr[myfunc()] << endl;  
}
```

First item in the stack

More in the stack

Compiler is here



1.10 Order of Insertions

Order of Insertions



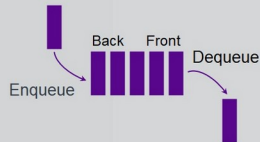
- push('(') – stack now: (
- top – compares '(' to ')'
- pop – stack now: (empty)
- push('{') – stack now: {
- push('[') – stack now: {[
- top – compares '[' to ']'
- pop – stack now: {
- push('(') – stack now: {(
- top – compares '(' to ')'
- pop – stack now: {
- push('[') – stack now: {[
- push('(') – stack now: {[(
- top – compares '(' to ')'

1.11 Queue, What Is It

Queue, What Is It



- First in, first out (FIFO) data structure
 - The first item which is enqueued if the first item dequeued
 - Often used for buffering information
- Push, pop and top are often the accessor functions, however it may be enqueue, dequeue and top.
- Supplemental functions
 - clear
 - isEmpty
 - size



1.12 Queue, How It Works

Queue, How It Works



- Enqueue adds to the end
- Dequeue removes from the front
- Could also be vice-versa
- Supplemental functions need to be provided as well as Big 3, if applicable



1.13 Queue Storage

Queue Storage




	List	Array
push	$O(1)$	$O(1)/O(N)$
pop	$O(1)$	$O(N)$
top	$O(1)$	$O(1)$
clear	$O(N)$	$O(1)$
isEmpty	$O(1)$	$O(1)$
size	$O(1)$	$O(1)$



1.14 Knowledge Check

(Multiple Choice, 10 points, 1 attempt permitted)

Knowledge Check



When comparing the running times of Stack and Queue storage with respect to arrays, all of the functions have the same running times, except for which function?

☐ top

☒ pop

☐ push

☐ clear

Correct	Choice
	top
X	pop
	push
	clear

Feedback when correct:

That's right! You selected the correct response.

Feedback when incorrect:

You did not select the correct response.

Correct (Slide Layer)

Knowledge Check

When comparing the running times of Stack and Queue storage with respect to arrays, all of the functions have the same running times, except for which function?

☐ top

☒ pop

☐ push

☐ clear

Correct

That's right! You selected the correct response.

Continue

Incorrect (Slide Layer)

Knowledge Check

When comparing the running times of Stack and Queue storage with respect to arrays, all of the functions have the same running times, except for which function?

☐ top

☒ pop

☐ push

☐ clear

Incorrect


You did not select the correct response.

Continue

1.15 Queue - Continued

Queue - Continued

- Much more obvious that we will use a List
- Similar implementation to the stack with minor changes
 - Enqueue will add to the end of the queue
 - Dequeue will remove from the front
 - Necessary to make sure the list has both head and tail pointers.

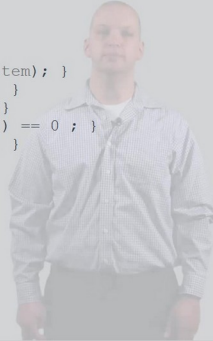


1.16 Queue Code

Queue Code



```
template <class T>
class Queue{
    list<T> data;
public:
    void enqueue(T newItem){ data.push_back(newItem); }
    T dequeue() { return data.pop_front(); }
    T top() const { return *data.begin(); }
    bool isEmpty() const { return data.size() == 0; }
    int size() const { return data.size(); }
    void clear() { data.clear(); }
};
```



1.17 Queue, What Is It Used For

Queue, What Is It Used For

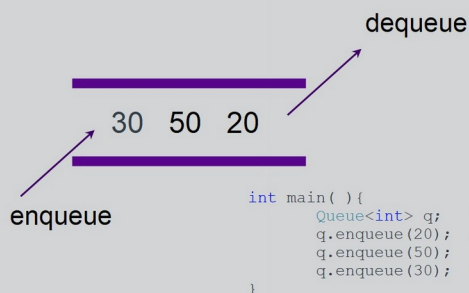


- Storage Buffers
- Anywhere that we need an ordered list with First in, First out properties




1.18 Queue Image

Queue Image




1.19 Stacks and Queues - Conclusion

Stacks and Queues - Conclusion




- What we learned
 - Stacks
 - How they work
 - How they are implemented
 - What they are used for
 - Queues
 - How they work
 - How they are implemented
 - What they are used for



1.20 Knowledge Check

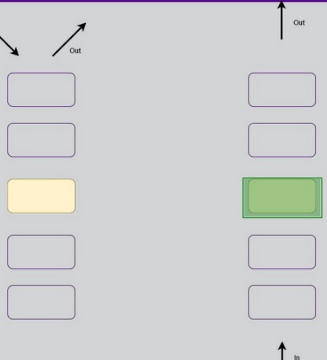
(Hotspot, 10 points, 4 attempts permitted)

Knowledge Check



In the image, assume the following: Every second, 1 item is removed from the stack and queue. At 2 seconds, an item is added to both the stack and queue. No items are entered before and after the 2 second mark.

Click which yellow block will be the first to leave.



Feedback when correct:

That's right! You selected the correct response.

Feedback when incorrect:

You did not select the correct response.

Correct (Slide Layer)

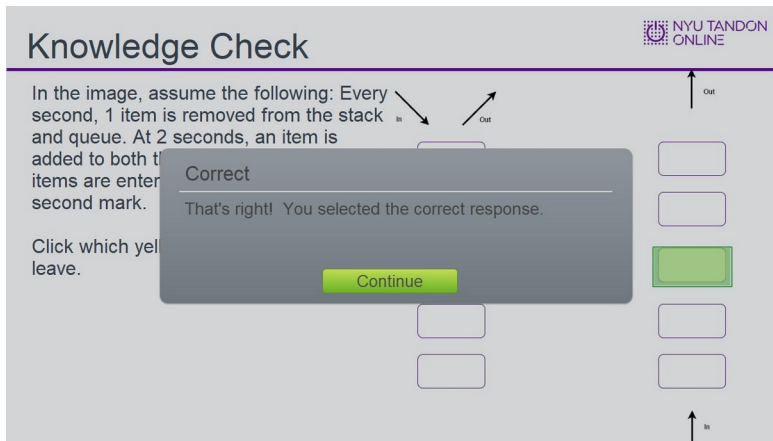
Knowledge Check NYU TANDON ONLINE

In the image, assume the following: Every second, 1 item is removed from the stack and queue. At 2 seconds, an item is added to both the stack and queue. At 4 seconds, 2 items are entered into the queue. At 6 seconds, 1 item is removed from the queue.

Click which yellow box is correct.

Correct
That's right! You selected the correct response.

Continue



Incorrect (Slide Layer)

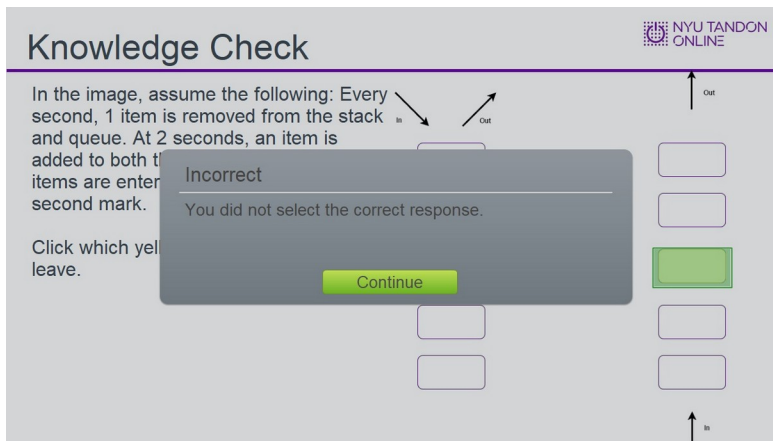
Knowledge Check NYU TANDON ONLINE

In the image, assume the following: Every second, 1 item is removed from the stack and queue. At 2 seconds, an item is added to both the stack and queue. At 4 seconds, 2 items are entered into the queue. At 6 seconds, 1 item is removed from the queue.

Click which yellow box is correct.

Incorrect
You did not select the correct response.

Continue



Try Again (Slide Layer)

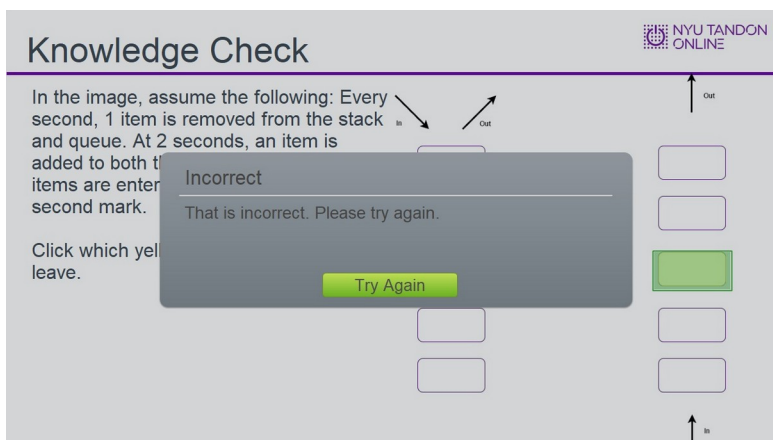
Knowledge Check NYU TANDON ONLINE

In the image, assume the following: Every second, 1 item is removed from the stack and queue. At 2 seconds, an item is added to both the stack and queue. At 4 seconds, 2 items are entered into the queue. At 6 seconds, 1 item is removed from the queue.


Click which yellow box is correct.

Incorrect
That is incorrect. Please try again.

Try Again



1.21 End of Module



NYU TANDON
ONLINE

End of Module

Exit