


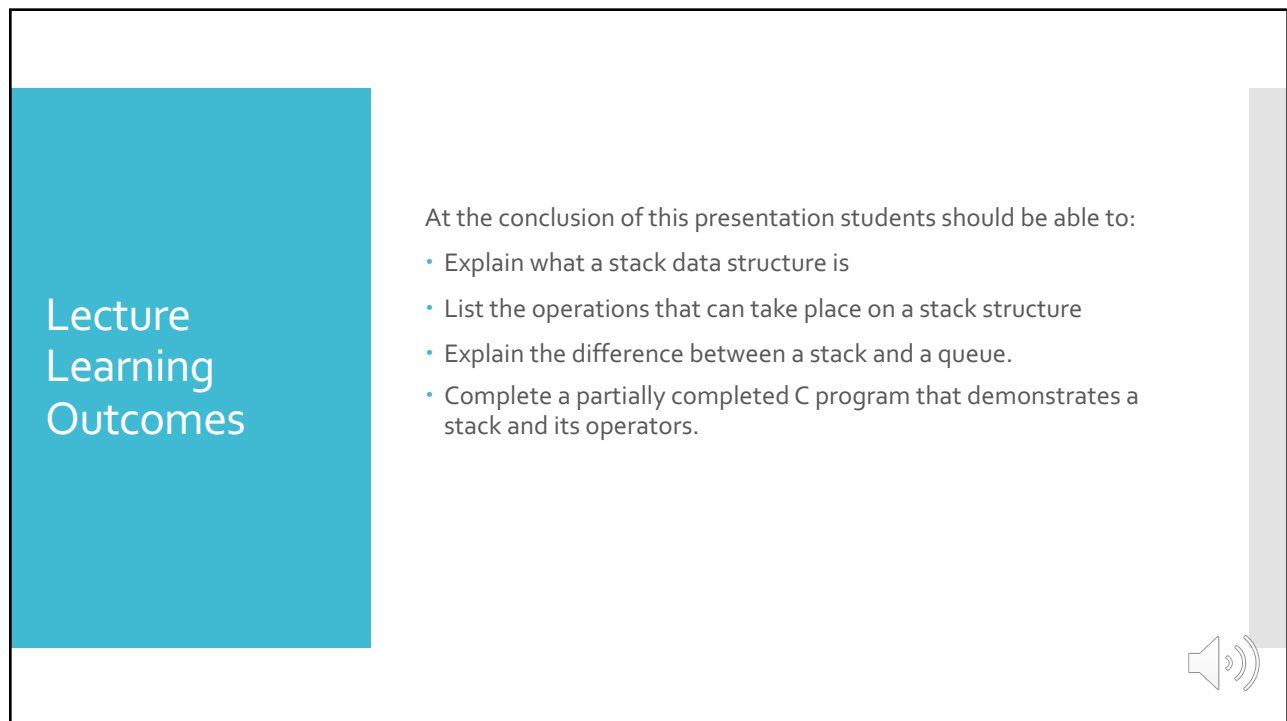

The slide features a large blue rectangular area on the right side containing the word "Stacks" in white. To the left of this area is a vertical grey bar. Below the blue area is a dark grey horizontal bar containing the text "CS2263 – Systems Software Development" in white. A small speaker icon is located in the bottom right corner of the slide.

Stacks

CS2263 – Systems Software Development



1




The slide has a blue vertical bar on the left side with the text "Lecture Learning Outcomes" in white. To the right of this bar, the text "At the conclusion of this presentation students should be able to:" is followed by a bulleted list of four items. A small speaker icon is located in the bottom right corner of the slide.

Lecture Learning Outcomes

At the conclusion of this presentation students should be able to:

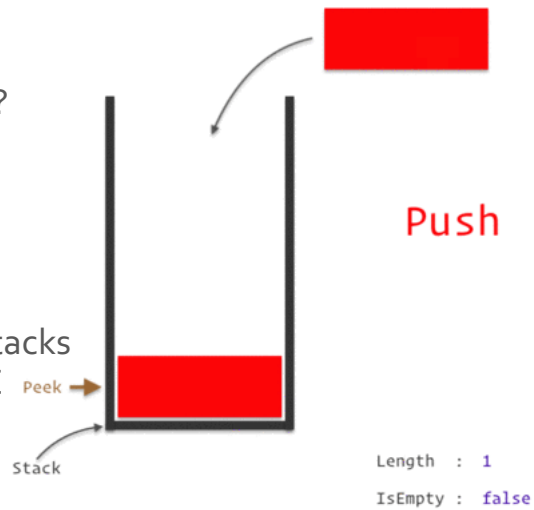
- Explain what a stack data structure is
- List the operations that can take place on a stack structure
- Explain the difference between a stack and a queue.
- Complete a partially completed C program that demonstrates a stack and its operators.



2

Stack

- What is a stack?
- How do we manipulate it?
 - Push/add
 - Pop/remove
- Manipulating stacks and queues in C



<https://medium.com/@1991dharapatel/javascript-stacks-and-queues-130-ba28759>

3

Stack in Code

Easily implemented using array, e.g.

```
int stack[MAX], size = 0;
```

Push value

```
if(size < MAX){
    stack[size] = value;
    size++;
}
```

Pop value

```
if(size > 0){
    value = stack[size];
    size--;
}
```

Note that the audio is incorrect: In both push and pop we modify the array index after using it. The code here has been corrected.

4

FIFO Queue

- A bit more tricky: add and remove from opposite ends.
- Use circular list, and keep track of front and size of queue.

IN



OUT

<https://qifer.com/en/gifs/queue>



5

Queue in Code

Easily implemented using array, e.g.

```
int queue[MAX];
int front=0, size=0;
```

Enqueue value

```
if(size < MAX){
    queue[(front+size)%MAX] = value;
    size++;
}
```

Dequeue value

```
if(size > 0){
    value = queue[front];
    front = (front+1)%MAX;
    size--;
}
```



6



forNextDay()

- `playStack.c` (in `L2src.zip` on the lms) takes commands and values to manipulate a stack. Make the necessary additions so that it works.
- Although you don't need to submit anything for it, you should begin reading Chapter 2 in the text.

