CS 225

Data Structures

September 9 – Array List Take 2

G Carl Evans

reflections | projections

Where Industry Meets Academia

What is reflections | projections (RP)?

- **Largest** student-run tech conference in the Midwest
- Weeklong conference from **September 26-30th**
- Distinguished speakers from industry and academia
- Corporate tech talks and opportunities from top companies
- Open to ANYONE interested in tech, **all majors and years** welcome

CATERPILLAR®

Qualcomm

Uber

Verkada

VISA

CLOUDFLARE

Sandia National Laboratories

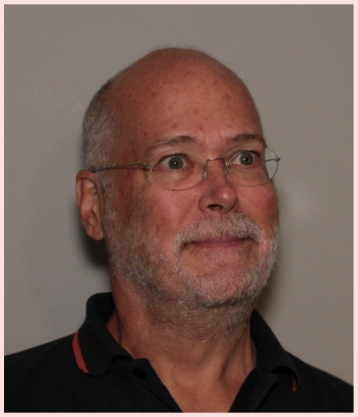
AECHELON TECHNOLOGY

databricks

CIO of Hewlett Packard,
Rashmi Kumar

Googler / Founder of Illuminate AI,
Aishwarya Srinivasan

Chief Architect at Oracle,
Todd Little



\+ More!

Why Should I Attend R|P?

Networking opportunities with top companies like Uber and Databricks

Speaker talks from industry leaders

FREE food and swag!

Participate in a beginner-friendly AI hackathon!

register now at the
link above

learn more at reflections projections.org
We hope to see you there!

CS 225

Data Structures

September 9 – Array List Take 2

G Carl Evans

Array Implementation

	Singly Linked List	Array
Insert/Remove at **front**		
Insert at **given** element		
Remove at **given** element		
Insert at **arbitrary** location		
Remove at **arbitrary** location		

Queue ADT

- [Order]:
- [Implementation]:
- [Runtime]:

Stack ADT

- [Order]:
- [Implementation]:
- [Runtime]:

Queue.h

```
 1  #pragma once
 2
 3  template <typename T>
 4  class Queue {
 5    public:
 6      void enqueue(T e);
 7      T dequeue();
 8      bool isEmpty();
 9
10    private:
11      T *items_;
12      unsigned capacity_;
13      unsigned size_;
14  };
15
16
17
18
19
20
21
22
```

What type of implementation is this Queue?

How is the data stored on this Queue?

Queue.h

```
1  #pragma once
2
3  template <typename T>
4  class Queue {
5    public:
6      void enqueue(T e);
7      T dequeue();
8      bool isEmpty();
9
10   private:
11     T *items_;
12     unsigned capacity_;
13     unsigned size_;
14 };
15
16
17
18
19
20
21
22
```

What type of implementation is this Queue?

How is the data stored on this Queue?

```
Queue<int> q;
q.enqueue(3);
q.enqueue(8);
q.enqueue(4);
q.dequeue();
q.enqueue(7);
q.dequeue();
q.dequeue();
q.enqueue(2);
q.enqueue(1);
q.enqueue(3);
q.enqueue(5);
q.dequeue();
q.enqueue(9);
```

Queue.h

```
1  #pragma once
2
3  template <typename T>
4  class Queue {
5    public:
6      void enqueue(T e);
7      T dequeue();
8      bool isEmpty();
9
10   private:
11     T *items_;
12     unsigned capacity_;
13     unsigned size_;
14 };
15
16
17
18
19
20
21
22
```


```
Queue<char> q;
…
q.enqueue(m);
q.enqueue(o);
q.enqueue(n);
…
q.enqueue(d);
q.enqueue(a);
q.enqueue(y);
q.enqueue(i);
q.enqueue(s);
q.dequeue();
q.enqueue(h);
q.enqueue(a);
```