Module 26: Partially Filled Arrays

Intro to Computer Science 1 - C++

Professor Scott Frees

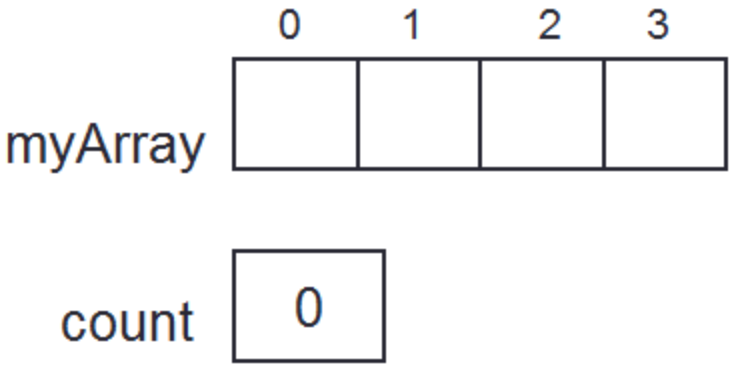
Size versus "capacity"

Often, we do not know in advance *exactly* how many items the user will enter...

We **cannot** change the size of the array - we must declare a size that is *big enough* for most situations

The size we declare the array with is really its capacity - it doesn't mean we must put data at each element

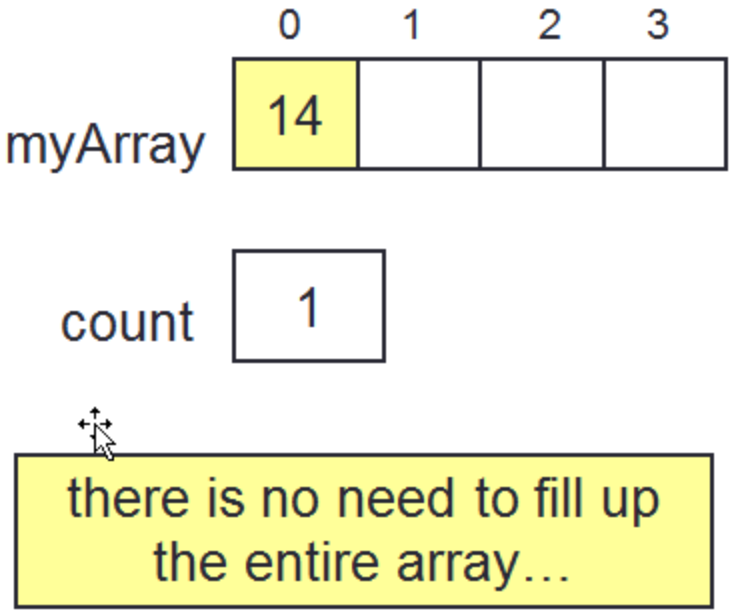
Filling Arrays




```
int myArray[4];
int count = 0;
```

there is no need to fill up the entire array…

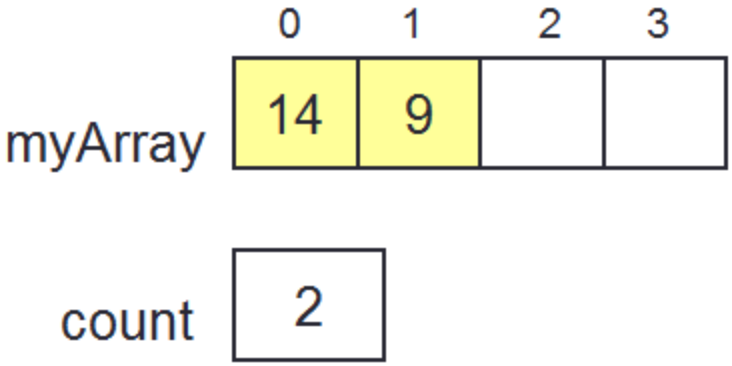
Filling Arrays




```
int myArray[4];
int count = 0;

myArray[count] = 14;
count++;
```

Filling Arrays



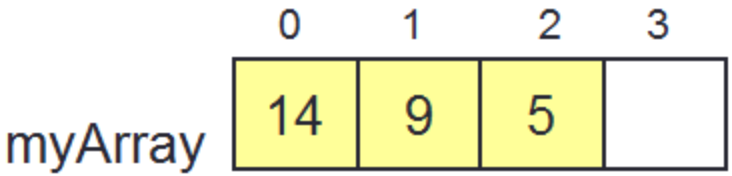

```
int myArray[4];
int count = 0;

myArray[count] = 14;
count++;

myArray[count++] = 9;
```

there is no need to fill up the entire array…

Filling Arrays



there is no need to fill up the entire array…

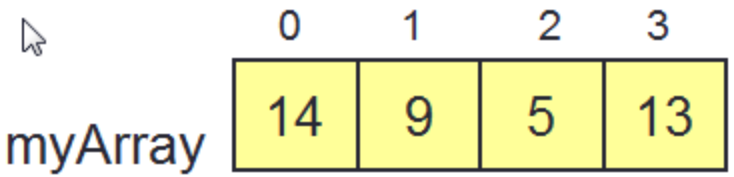
```
int myArray[4];
int count = 0;

myArray[count] = 14;
count++;

myArray[count++] = 9;

myArray[count++] = 5;
```

Filling Arrays



count: 4

there is no need to fill up the entire array…

```
int myArray[4];
int count = 0;

myArray[count] = 14;
count++;

myArray[count++] = 9;

myArray[count++] = 5;

myArray[count++] = 13;
```

Programming Example 30

- Write a program that reads in **up to 15** numbers from user:
- Next let the user ask if a particular number is present in the array
  - Yes/No
  - Count?