Linear collections

Arrays

CS111 Computer Programming

Department of Computer Science
Wellesley College

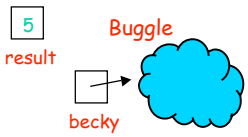
A bedtime story

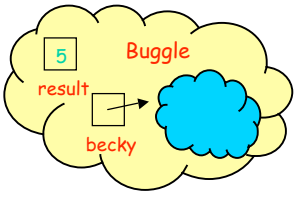
Once upon a time our variables referred to single objects (Buggles, ints, and things …).

But we were sad because we wanted to work with large collections of things.

We tried bundling all our things into one big class.

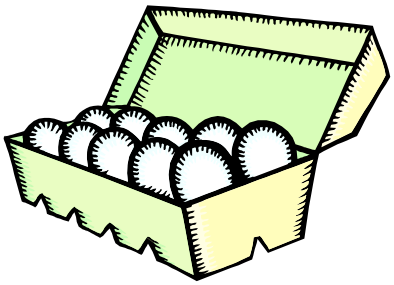
This brought a dark cloud over our heads, until lists came along and the cloud went away for a while.





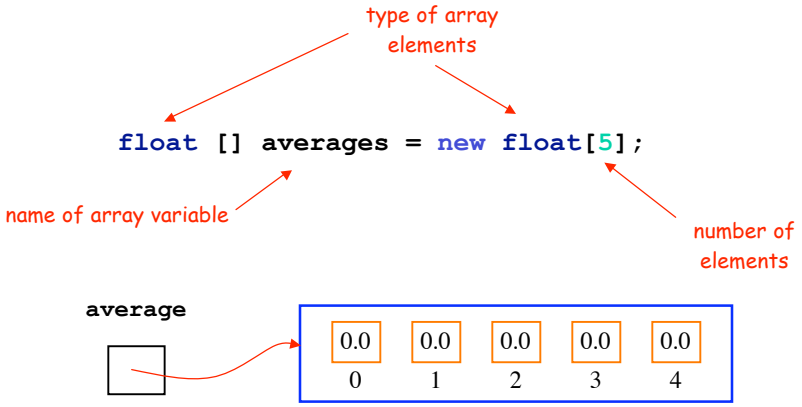
Arrays

- An array is an ordered collection of elements all of the same type or class.*
- In Java arrays are objects of a fixed length.
- Arrays are very good if you want to get to something quickly.



*No casting.

Array notation

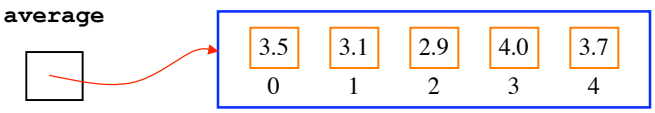


Assigning values to an array

```
float [] averages = float[5];
averages[0] = 3.5;
averages[1] = 3.1;
averages[2] = 2.9;
averages[3] = 4.0;
averages[4] = 3.7;
```

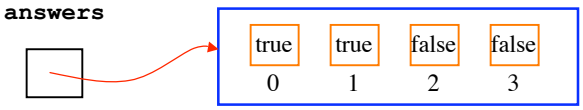
or, more compactly

```
float averages[] =
 {3.5, 3.1, 2.9, 4.0, 3.7}
```

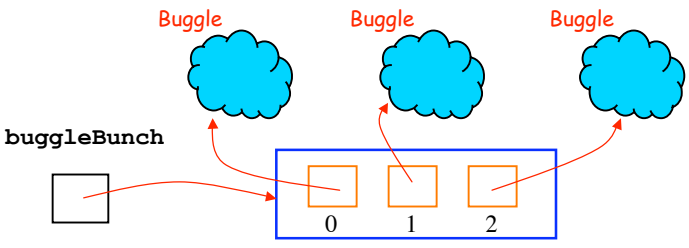


Arrays can be any type

```
boolean [] answers = new boolean[4];
```

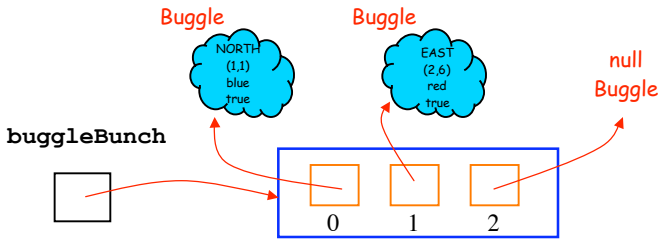



```
Buggle [] buggleBunch = new Buggle[3];
```



The buggleBunch

```
Buggle [] buggleBunch = new Buggle[3];
buggleBunch[0] = new Buggle();
buggleBunch[0].setColor(Color.blue);
buggleBunch[0].setHeading(Direction.NORTH);
buggleBunch[1] = new Buggle();
buggleBunch[1].setPosition(new Point(2,6));
```

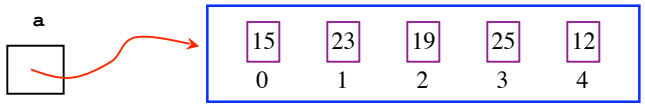


Swapping array elements

```
public static void swap (int[] a, int i, int j)
{
     int temp = a[i];
     a[i] = a[j];
     a[j] = temp;
}
```

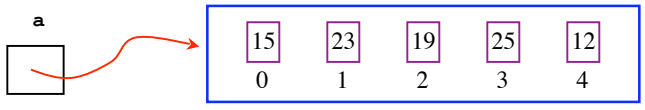
Sum up the elements of an int array

```
public static ___ sumIntArray(___ __ a)
{
      int sum = 0;




      return sum;
}
```



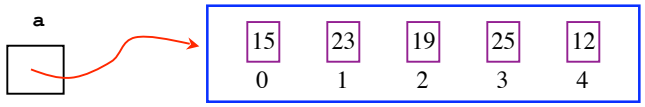
Iterating over an array

```
public static int sumIntArray(int [] a)
{
      int sum = 0;
      for (int i=0; i<a.length; i++) {
            sum = sum + a[i];
      }
      return sum;
}
```



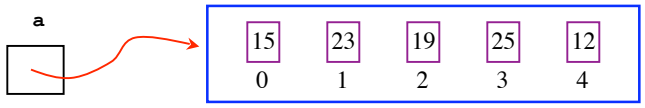
Find the minimum element in an int array

```
public static ___ minIntArray(___ __ a)
{
     int minVal = Integer.MAX_VALUE;









     return minVal;
}
```



minIntArray(): The method

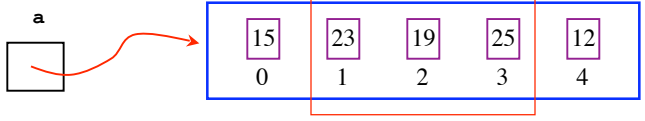
```
public static int minIntArray(int [] a)
{
     int minVal = Integer.MAX_VALUE;
     for (int i = 0; i < a.length; i++) {
          if (a[i] < minVal) {
               minVal = a[i];
          }
     }
     return minVal;
}
```



minBetween()

```
public static int minBetween(int [] a, int lo, int hi)
{
      int minVal = Integer.MAX_VALUE;



      // return min integer in array between lo and hi



      return minVal;
}
```

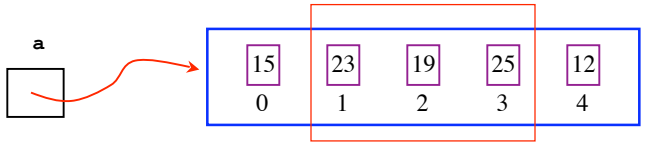
minBetween(a, 1, 3) only looks for minVal here



minBetween()

```
public static int minBetween(int [] a, int lo, int hi)
{
      int minVal = Integer.MAX_VALUE;
      for (int i = lo; i <= hi; i++) {
            if (a[i] < minVal) {
                  minVal = a[i];
            }
      }
      return minVal;
}
```

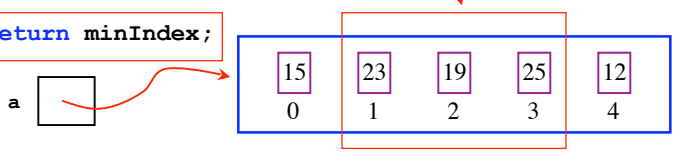
change limits of search so that minBetween(a, 1, 3) looks between lo & hi



What changes do we have to make …

```
public static int minIndexBetween(int [] a,int lo,int hi)
{
      int minVal = Integer.MAX_VALUE;

      for (int i = lo; i <= hi; i++) {
            if (a[i] < minVal) {
                  minVal = a[i];

            }
      }
      return minIndex;
}
```

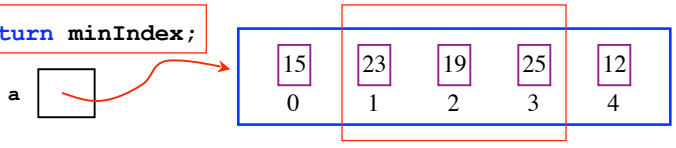
return the index of minVal between lo & hi



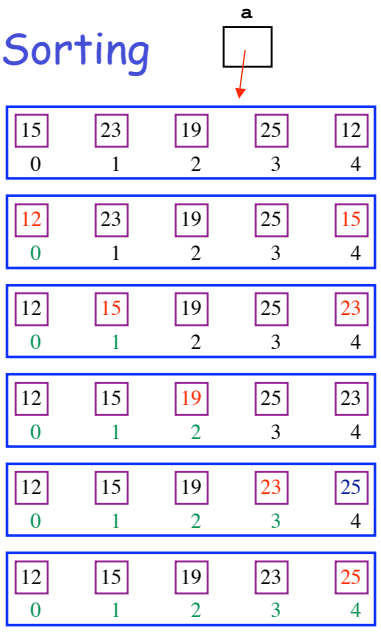
minIndexBetween()

```
public static int minIndex(int [] a, int lo, int hi)
{
      int minVal = Integer.MAX_VALUE;
      int minIndex = -1;
      for (int i = lo; i <= hi; i++) {
            if (a[i] < minVal) {
                  minVal = a[i];
                  minIndex = i;
            }
      }
      return minIndex;
}
```

return the index of minVal between lo & hi



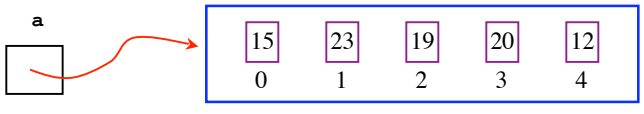
Sorting



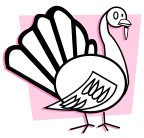

```
int lastIndex = a.length - 1;

swap(a,
     0,
     minIndexBetween(a,0,lastIndex));
swap(a,
     1,
     minIndexBetween(a,1,lastIndex));
swap(a,
     2,
     minIndexBetween(a,2,lastIndex));
swap(a,
     3,
     minIndexBetween(a,3,lastIndex));
swap(a,
     4,
     minIndexBetween(a,4,lastIndex));
```


selectionSort()

```
public static void selectionSort(int [] a, int lo, int hi)
{
     for (int i = lo; i < hi; i++) {
          swap(a, i, minIndexBetween(a, i, hi));
     }
}
```



Good news & bad news



	Lists	Arrays
Find first element		
Find last element		
Add element to front		
Add element to end		
Find length		
Mutable		
Work for diff types