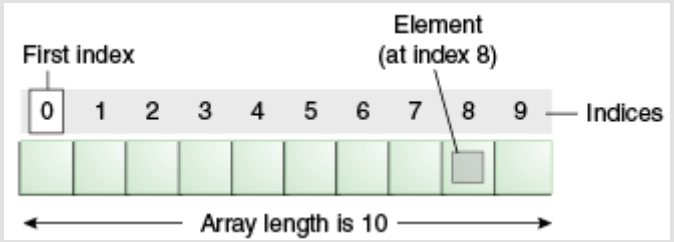
CS620c Structured Programming in Java

Day 6 lecture 1

Arrays and command line input

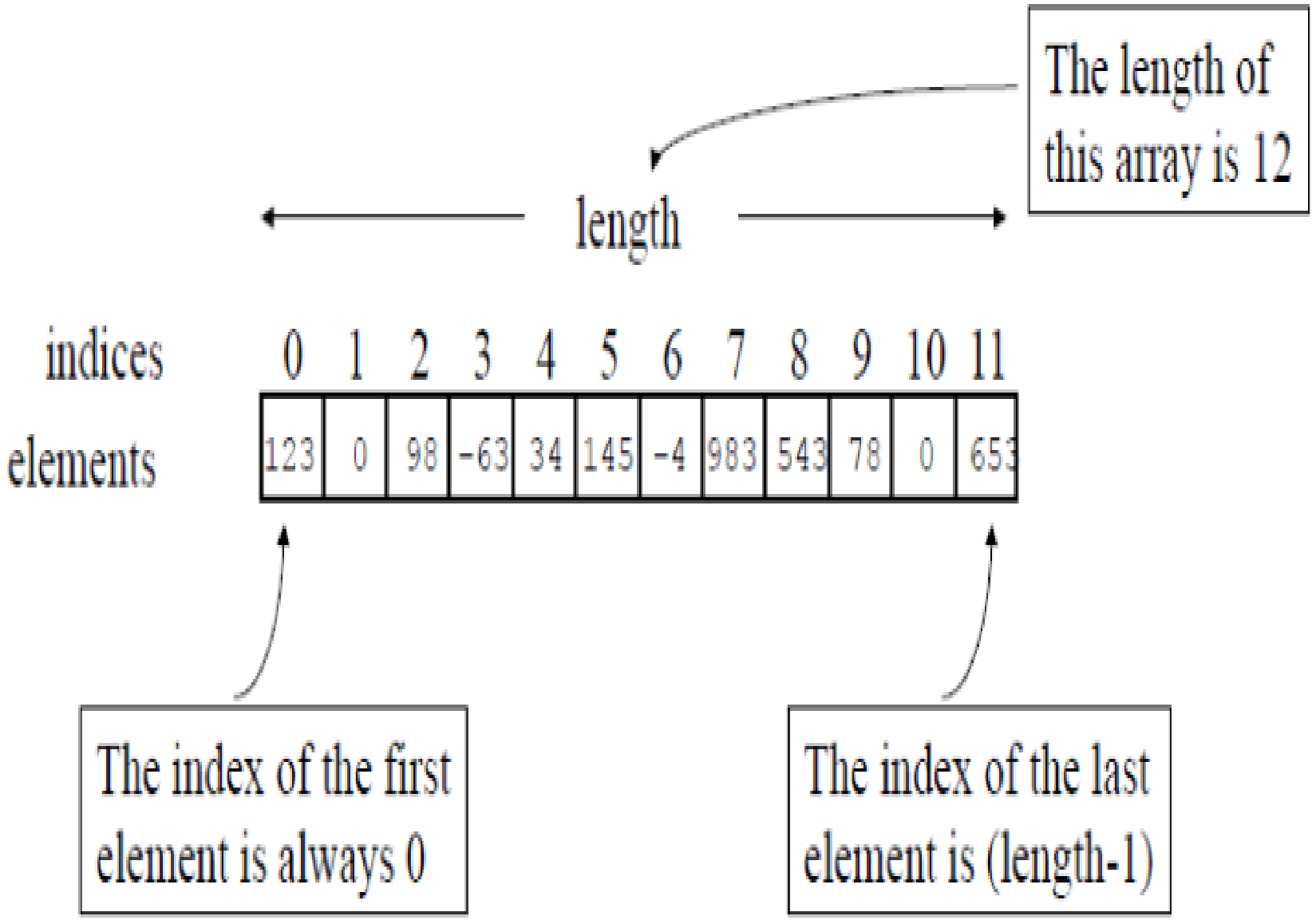
Arrays

- What is an array?



- An *array* is a container object that holds a fixed number of values of a single type.
- Think of an array as being like a numbered list
- The first item on the list is marked as being element ‘0’
  - In Java (and some other languages such as C), we access elements of an array by their number or **index**

```
0.  Milk
1.  Eggs
2.  Bread
3.  Toothpaste
4.  Bananas
```


The length of this array is 12
length
indices
0 1 2 3 4 5 6 7 8 9 10 11
elements
123 0 98 -63 34 145 -4 983 543 78 0 653
The index of the first element is always 0
The index of the last element is (length-1)

Arrays

- Arrays are useful when you know you're going to have a lot of variables of the same type being used for a similar purpose.

- For instance, if you wanted to store a list of names; or a bunch of results.

- A given array can only store one type of variable at a time.
  - You can't mix ints and Strings or floats and ints in a single array.

Declaring an Array

- Similar to declaring a single variable:

- `dataType[] identifier;`

- Examples:
  - int [ ] arrayOfInts;
  - boolean [ ] arrayOfBooleans;
  - String [ ] arrayOfStrings;

- This only creates a (‘reference’) to manipulate the array
  - **It doesn’t set aside memory for the array, we have to do that manually.**

Constructing an Array

- This reserves memory for the array and sets up its internal variables
  - Each of the elements of the array needs to be given a space in memory, the same as if they were individually-declared variables.
- Similar to creating an object from a class:
  - `identifier = new dataType[ arraySize ];`
- `arraySize` is an integer value representing the number of elements that the array can hold.

Constructing an Array

- We can combine the declaration and creation into one command as follows:
  - `dataType identifier = new dataType [ size ];`
- The syntax is almost identical to that of class/object instantiation/construction!
  - `Bicycle bike = new Bicycle(); // Object`
  - `int arrayOfStrings = new int[ 50 ]; // Array`

Getting the Array length

- The length of an array is established when the array is instantiated.
- After creation, its length is fixed.
  - We can't (strictly-speaking) make an array bigger to accommodate more elements if we change our minds later.
- This won't matter at first, but its importance will become apparent eventually.
- This means that you need to set aside 'enough' space in the array to hold all of the elements you think you'll need.
  - This can be wasteful

Why use arrays?

```
4   An array lets us replace this:
5
6       x0 = 0;
7       x1 = 1;
8       x2 = 2;
9       x3 = 3;
10      x4 = 4;
11      x5 = 5;
12
13  With this:
14
15      x[0] = 0;
16      x[1] = 1;
17      x[2] = 2;
18      x[3] = 3;
19      x[4] = 4;
20      x[5] = 5;
```

What's the advantage?

- To access each of x0, x1, x2, x3, etc. you need to know the identifiers of those variables and you need to explicitly refer to them in your code:

```
23      System.out.println( x0 );
24      System.out.println( x1 );
25      System.out.println( x2 );
26      System.out.println( x3 );
27      System.out.println( x4 );
28      System.out.println( x5 );
```

Array access by index

- Arrays let us do this instead:

```
36      x[0] = 0;
37      x[1] = 1;
38      x[2] = 2;
39      x[3] = 3;
40      x[4] = 4;
41      x[5] = 5;
42
43      for(count=0; count<5; count++)
44      {
45          System.out.println( x [count] );
46      }
```

- Being able to work with sets of data programmatically (such as in loops) like this is a very powerful feature

Three steps to using and Array

- 1 - Declare the Array
  - `int [] someArray;`
- 2 - Construct the Array
  - someArray = **new** int [arraySize];
- 3 - Initialize the Array
  - `someArray[0] = 0;`
  - **`Alternatively:`**

```
49  // Initialize all elements to 0
50  for(int i = 0; i < arraySize; i++)
51  {
52      someArray[i] = 0;
53  }
```

NB: If you try to access a value beyond the size of the array you will generate an out of bounds error.

Multidimensional Arrays

- Arrays can also hold other arrays
- There's a simple syntax for doing this:
  - int someArray[ ][ ] = new int[4][5] ;
- You can now think of each 'dimension' of the array as being like columns and rows in a table
- Each of the 'sub-arrays' has to be constructed and initialized

Array Examples

Access the Array folder in the code folder in Day 6 for examples of array programs.

Handling command line arguments stored in String [] args

- Echo.java
- Echo2.java
- CmdLine.java
- CmdLine_Safe.java
- CmdLine_Safe_Methods.java