

CS 200 Sections 02 & 04 Spring 2013

Week #13: Top 3 Lessons Learned

1. It is important to remember that when declaring an array, It must have the same data type, and the size of the array must be a positive integer value, as you cannot declare a fractional portion of memory for data storage. See examples:

`int [] number = new int [5]` is correct.

`int[] number = new double [5]` is not correct.

`int[] number = new int [5.5]` is not correct.

2. Remember that array subscripts start with the number 0 (not 1).

Example:

num =	3	8	11	10	7
	num[0]	num[1]	num[2]	num[3]	num[4]



values stored in the array



use [subscript] to access specific array values

You must remember that the length of an array is the number of elements in the array, which is one more than the largest subscript.

3. Remember that the “Bubble Sort” repeatedly compares adjacent elements of an array. The first and second elements are compared and swapped if out of order. Then the second and third elements are compared and swapped if out of order. This sorting process continues until the last two elements of the array are compared and swapped if out of order. This swapping process is then repeated n-1 passes through the entire array, where n= the size of the array.

T. Blanchard

1. **Swapping** is very easy with Arrays. To perform simple swap we have to use this algorithm:

```
temp = a[n];  
a[n] = a [n+1];  
a[n+1] = temp;
```

Swapping values is commonly used to sort an array of values to place them in order.

2. Also, we cannot forget about for loops which are very helpful:

```
for (int i=0; i < anArray.length; i++) // check one by one beginning from 0 which is the first element of an array
{
    // we have to always begin from 0
    System.out.println( anArray[i] ); // display all elements of array and goes back to new line
    if(anArray[i] % 2 == 0); // for example, check one by one if even
    noReminder++; // increment value if true to count number of even values
} // here i++ happens as last command in for loop!
//and again go back till i < anArray.length is satisfied
```

3. Creating an Array:

```
int[] anArray; // declares an array of integer type – address assigned to start of storage space
```

```
anArray = new int[10]; // allocates memory storage for 10 integers
```

```
anArray[0] = 12; // initialize first element
```

```
anArray[1] = 2; // initialize second element etc.
```

3. We can also declare an Array with elements already in:

```
int[] anArray = { 5, 2, 30 };
```

or

```
char [] aCharArray = { a, b, c };
```

or even with elements of String:

```
String[] anArrayWithSomeWords = new String[]{ "one", "two", "three" };
```

M. Mardosz

Three things that I learned are that the data stored in an array must be of the same data type because of the way the certain data types are stored in memory.

Second, it is important to remember that the indices of an array start at 0. If you want to figure out the last valid index of an array, subtract the length of the array by one.

Third, a partially filled array can be used when you do not yet know the possible values of all the indexed variables, but you will need a “counter variable” to keep track of how many values you place/assign to the array.

Jose Gomez

My lessons are:

- 1. Arrays are collections of items of the same data type. The index of the array inside [] is called a subscript. It is used for allocation and access. The allowed subscript values are integers. The range of the subscripts is 0 to (size of array - 1) .**
2. Arrays can be declared and initialized by the single line of code: `int [] myNums = new int [4];` Arrays can be also written: `int [] x = {17,-77,23,24}`
- 3. When you don't know the exact size needed for an array, just make a large enough estimate, therefore making a partially filled array. Algorithm: Declare large array, make counter for number of items, last index is (counter -1).**

D. Starostka

1. An array is a collection of items of the same type.
2. In Java, the index of the first value in the array is always at zero [0].

D. Mirdadi

- 1.) It is proper coding to have your array size set up as a constant, but that does not mean you have to fill the whole array to make it work.

If you need the user to set up the size of the array; there is only ONE way to set it up:

```
ex: int [] x; //declare array object
    int size; //size of array must be of integer type and must be a positive integer
    System.out.print("prompt for size: ");
    size = kbd.nextInt; //get size of array from user
    x = new int [size]; //allocate storage space for "size" number of values
```

- 2.) you could declare your array: `int [] x;`
and then initialize to an array storing 3 values `x = new int[3];`
or you could do both in the same time by: `int []x = new int[3];`

- 3.) Remember, an array is just like a string. It starts (first value) at 0 and ends at length - 1, so when you're looking for an element or need to call it up make sure you keep it in bounds.

E. Herring

Time yourself when doing tracing problems and do a lot of them to get faster

When you create a variable where you will be storing a value temporarily, name it "temp" so you're not confused about what is in that variable.

In an array the first value will be stored at address 0

S. Malik