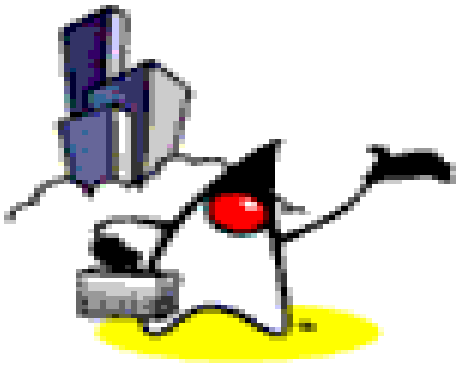
Java Array

Sang Shin
JPassion.com
"Code with Passion!"

Agenda

- What is an array?
- Declaration of an array type variable
- Instantiation of an array object
- Accessing array element within an array object
- Array length
- Multi-dimensional array



What is an Array?

Why you want to use an Array?

- Suppose we have three variables of same type *int*

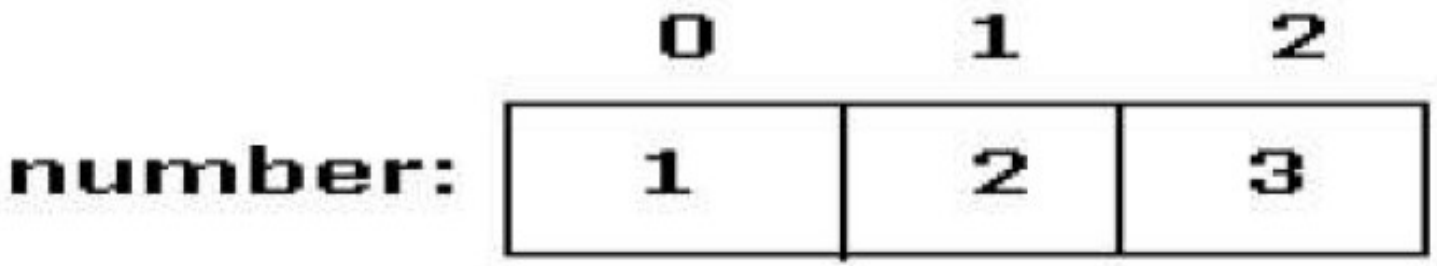
```
int number1;
int number2;
int number3;

number1 = 1;
number2 = 2;
number3 = 3;
```

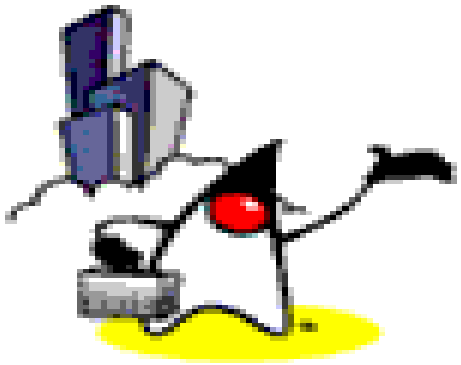
As you can see, initializing and using them individually is a tedious task. And it will get worse as the number of variables is getting larger.

Introduction to Arrays

- In Java (and other programming languages), there is a feature wherein we use one variable to store multiple items of same type and manipulate them. This type of variable is called an array.

- An array stores multiple data items of the same type, in a contiguous block of memory, divided into a number of slots.





Declaration of an Array

Declaring an Array Type variable

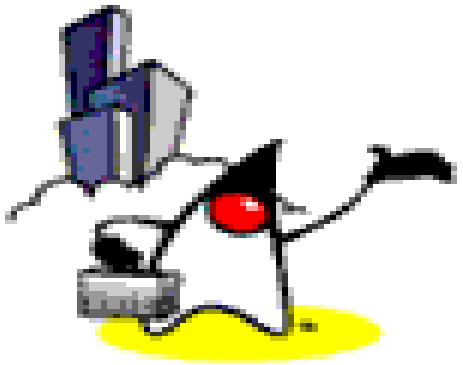
- To declare an array, write the data type, followed by square brackets [ ], followed by the variable name.

- For example,

```
int[] ages;  // “ages” is int array type
```

or

```
int ages[];  // This is allowed as well
```



Instantiation of an Array Object

Array Object Instantiation

- After declaring array type variable, we can create an array object instance and specify its length with a constructor statement.

- What does “Object Instantiation” mean?
  - In Java, this means creation of an object
- What is a “Constructor”?
  - In order to instantiate an object, Java calls a constructor method (We will cover more about instantiating objects and constructors in detail later.)

Array Object Instantiation

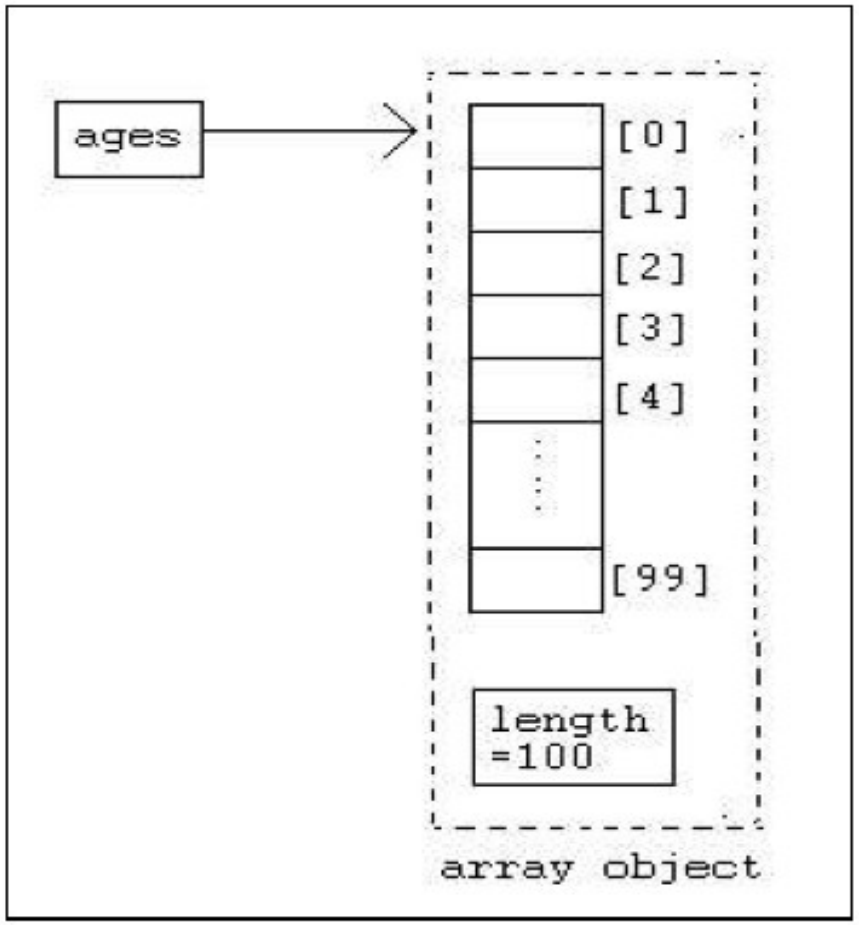
- To instantiate (or create) an array object, use the new keyword, followed by the square brackets containing the number of items you want the array to have
  - Array cannot be resized
- For example,

```
//declaration
int[] ages;

//instantiate int array object with length of 100
ages = new int[100];
```

or, can also be written as,

```
//declare and instantiate object
int[] ages = new int[100];
```

How does an Array Object look like in Memory?



Array Object Instantiation with data

- You can also instantiate an array object by directly initializing it with data.

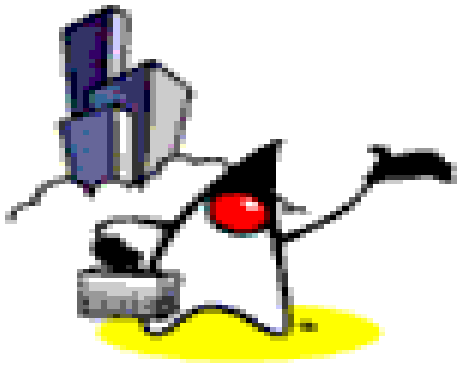
- For example,

```
int[] arr = {1, 2, 3, 4, 5};
```

This statement declares and instantiates an array of “int” type with five items (initialized to the values 1, 2, 3, 4, and 5).

Sample Program

```
1     // Creates an array of boolean type
2
3     boolean[] results = { true, false, true, false };
4
5     // Create an array of 4 double type initialized
6     // to the values of  {100, 90, 80, 75};
7
8     double[] grades = {100, 90, 80, 75};
9
10    // Create an array of Strings
11
12    String[] days = { “Mon”, “Tue”, “Wed”, “Thu”, “Fri”, “Sat”, “Sun”};
```



Accessing Array Element

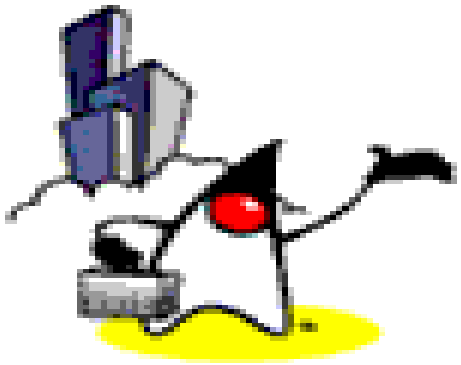
Accessing an Array Element

- To access an array element, you provide an index
- Index starts from 0

```
//assigns 10 to the third element of the array
ages[2] = 10;

//prints the last element in the array
System.out.println(ages[ages.length - 1]);
```

Initialization of the Array Elements

- Once an array object is created, the stored value of each member of the array will be initialized to zero for number type
  - int[ ] ages = new int[10]; // Each item is initialized to 0
- For reference data types such as Strings, they are initialized null
  - String[ ] ages = new String[10]; // Each item is initialized to null



Array Length

Array Length

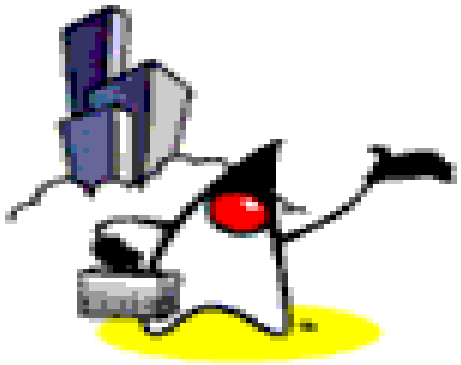
- In order to get the number of elements in an array, you can use the length field of an array.

- The length field of an array returns the size of the array.

```
arrayName.length
```

Array Length

```
1    public class ArraySample {
2        public static void main( String[] args ){
3            int[] ages = new int[100];
4
5            for( int i=0; i<ages.length; i++ ){
6                System.out.print( ages[i] );
7            }
8        }
9    }
```



Multi-Dimensional Array

Multidimensional Arrays

- Multidimensional arrays are implemented as arrays of arrays.
- Multidimensional arrays are declared by appending the appropriate number of bracket pairs after the array name.

Multidimensional Arrays

```
// integer array 512 x 128 elements
int[][] twoD = new int[512][128];

// character array 8 x 16 x 24
char[][][] threeD = new char[8][16][24];

// String array 4 rows x 2 columns
String[][] dogs = {{ "terry", "brown" },
                   { "Kristin", "white" },
                   { "toby", "gray"},
                   { "fido", "black"}
                  };
```

Multidimensional Arrays

- To access an element in a multidimensional array is just the same as accessing the elements in a one dimensional array.

- For example, to access the first element in the first row of the array dogs, we write,

```
System.out.print( dogs[0][0] );
```

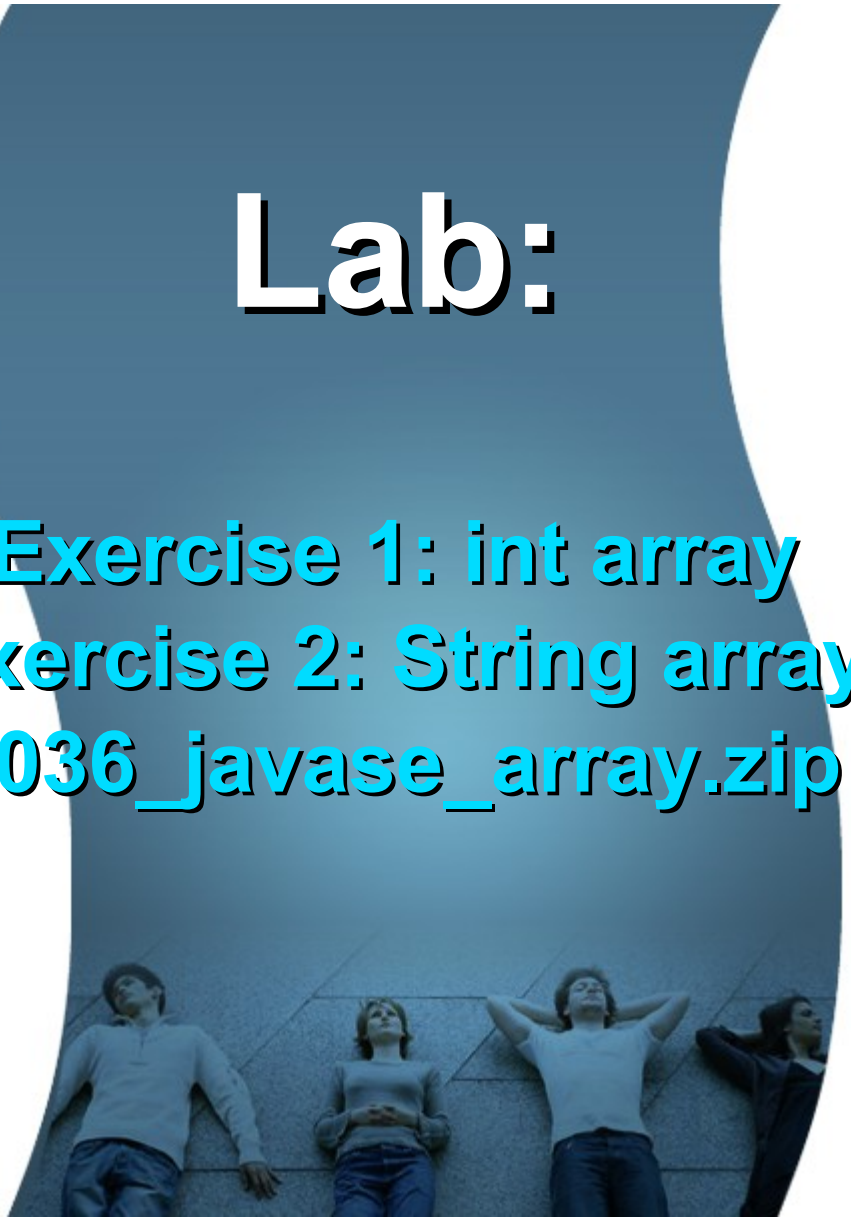
This will print the String "terry" on the screen.

Lab:

Exercise 1: int array

Exercise 2: String array

1036_javase_array.zip



Code with Passion!
JPassion.com