Introduction

- Many times in engineering we will encounter problems where large amounts of the same data type must be manipulated, such as temperature measurements, earthquake data, etc.
- It is inconvenient to define so many individual variables (i.e. int data1, int data2, int data3….).
- An array is a data structure consisting of variables of the same data type such as integers, floating point numbers or characters.
- You cannot mix the data types in an array.

Declaring a *Primitive* 1-D Array

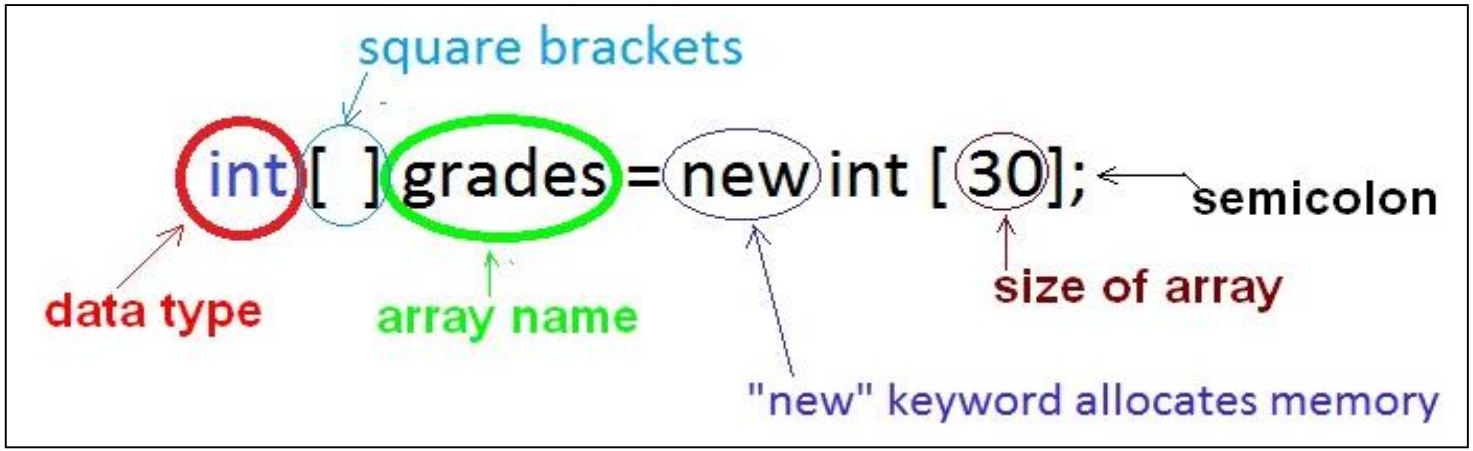
- The syntax for declaring an array is:

  datatype [ ] arrayname = new datatype[numberOfElements];

  Example:

  int [ ] grades = new int [30];

  Alternative syntax: int grades[] = new int [30];



The Index (subscript)

- The number inside the [ ] is referred to as the index or subscript.
- The index specifies the position in the array of a particular element.
- Must be a **positive integer** or an expression that evaluates to a positive integer.

Warning: the first element of the array is indexed with a 0. The last element of the array is indexed with the size of the array - 1

Referring to Array Elements

- In order to refer to individual array elements, you must state:

1. The array name
2. The subscript of the element in the array

- The first element of an array is referred to as the "*zeroth" element* because it is indexed with 0.

```
int myArray[  ] = {0, 2, 4, 6, 8};
```

myArray[0] = 0

myArray[1] = 2

myArray[2] = 4

myArray[3] = 6

myArray[4] = 8

Make your Arrays *Scalable*

- Make use of final int in order to declare the size of an array.

- Ex. final int size = 24;

  myArray = new int [size];

- This makes your program more *scalable*.

Initializing Arrays

1. In a declaration list.

```
int myArray[ ] = {0, 2, 4, 6, 8};
```

2. In a loop.

```
for( i = 0; i < 5; i++)
        myArray[ i ] = 2*i;
```

Exploiting the Array Index

1. To refer to a specific array element
2. To accomplish mathematical operations
3. Initializing arrays
4. Printing arrays

Example

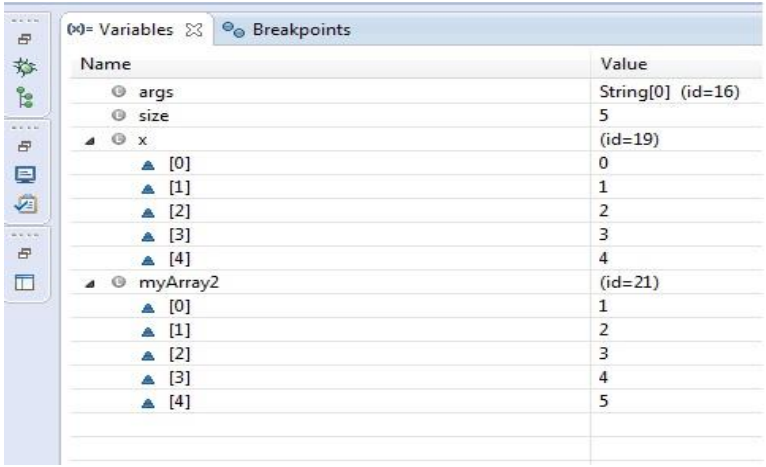
```
final int size = 3;

//declare and initialize an array
//WRONG SYNTAX: double temp[size] = {33.0, 32.5, 31.1};
float temp[] = {33.0, 32.5, 31.1};
//variables to hold some real numbers
float sum = 0.0, average = 0.0;

//add all of the elements
sum = temp[0] +temp[1] +temp[2]; //in class we will
                                 //sum in a loop

//take the average
average = sum/size;
```

Initialize/Printing Arrays

```
//use a constant to declare the size of the array
final int size = 3;

//an array
int x[ ] = new int [size];

//initialize the array in a loop
for(int i = 0; i < size; i++)
        x[i] = i;

//print out the array
for(int j = 0; j < size; j++)
        System.out.println(x[i]);
```

Final Points

- Arrays elements can be either primitive variables or refererence variables.
- An array of reference variables is also known as “an array of objects”.



Arrays of Objects

- Arrays of objects require two levels of instantiation:
  - One for the entire array
  - A second one for each object in the array – needs to be done individually. (see class example).