Sukkur IBA

Dr. Faheem Akhtar Rajput

Object Oriented Programming (JAVA)

Lecture 9

Scope and lifetime of Variables

- Java allows to use variables within any block
- A block defines the scope, each time new block is a new scope
- Declaring variable within the scope is preventing variable from localizing.
- Scope rules provide for encapsulation
- Scopes can be nested. For example, each time you create a block of code, you are creating a new, nested scope.

Example - 4

```
// Demonstrate block scope.
class Scope {
        public static void main(String args[]) {
                int x;    // known to all code within main
                x = 10;
                if(x == 10) {     // start new scope
                        int y = 20; // known only to this block
                        // x and y both known here.
                        System.out.println("x and y: " + x + " " + y);
                        x = y * 2;
                }
        // y = 100; // Error! y not known here
        // x is still known here.
        System.out.println("x is " + x);
        }
}
```

Type Conversion and Casting [1/2]

- It is fairly common to assign a value of one type to a variable of another type.
- If both types are compatible (Size) then java perform conversion automatically (int to long)
- But conversion from incompatible requires casting for eg. Double to byte (Explicit Conversion)
- **Automatic Conversion:** *widening conversion* takes place if
  - The two types are compatible
  - Destination type is large enough to hold source type

Type Conversion and Casting [2/2]

- **Casting Incompatible:** what if you want to assign an int value to a byte variable? This kind of conversion is sometimes called a *narrowing conversion*

(target-type) value

```
int a;

byte b;

 // ...

 b = (byte) a;
```

- *Truncation*: While assinging floating point values to integer type. Thus if the value 1.23 is assigned to an integer, the resulting value will simply be 1.

Automatic Type Promotion in Expressions

- When conversion occurs in expressions

```
byte a = 40;
byte b = 50;
byte c = 100;
int d = a * b / c;
```

- Result of a * b easily exceeds the range of either of its byte operands. But java performs automatic promotion to each byte short and char to int.

```
byte b = 50;
b = b * 2; // Error! Cannot assign an int to a byte!
```

- Result is integer value, so need casting

```
byte b = 50;
b = (byte)(b * 2);
```

Arrays

- Group of like-typed variables that are referred to by a common name
- May have one or more dimensions
- Element in an array accessed by index number
- ONE DIMENSIONAL
- MULTI DIMENSIONAL

One dimensional Arrays[1/4]

- A list of like-typed variables. The syntax for One dimensional Array is:

  type var-name[ ];

  int month_days[];

- Array here really not exist actually. month_days set to *null*. To allocate a memory location you must use *new* keyword

  array-var = **new** type[size];

  month_days = new int[12];

- Array declaration is 2 step process 1) declare variable of desired type 2) assign physical existence using new keyword

One dimensional Arrays[2/4]

- In Java all arrays are dynamically allocated
- To access an element of array by specifying the index number of that element.
- All array indexes start at zero

```
month_days[1] = 28;
```

OneD Arrays - Example. [3/4]

```
// Demonstrate a one-dimensional array.
class Array {
      public static void main(String args[])  {
            int month_days[]; month_days = new int[12];
            month_days[0] = 31;
            month_days[1] = 28;
            month_days[2] = 31;
            //.....
            //.....
            month_days[10] = 30;
            month_days[11] = 31;
            System.out.println("April has " + month_days[3] + " days.");
            }
}
```

One dimensional Arrays[4/4]

- Alternative declaration:

```
int month_days[] = new int[12];
```

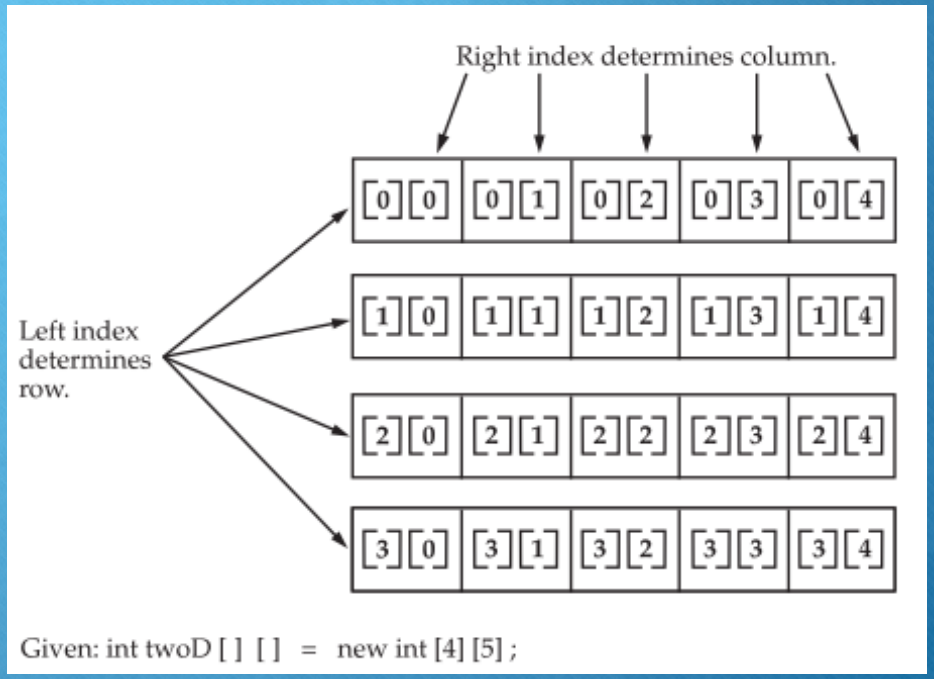
- Alternative declaration & initializations:

```
class AutoArray {
    public static void main(String args[])
    {
      int month_days[] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };
      System.out.println("April has " + month_days[3] + " days.");
    }
}
```

Multidimensional Arrays

- This is Arrays of Arrays
- Act like regular multidimensional arrays.

```
int twoD[][] = new int[4][5];
```



Multidimensional Arrays Example

```
public class TwoDArray {
	public static void main(String args[])
	{
		int twoD[][]= new int[4][5];
		int i, j, k = 0;
		for(i=0; i<4; i++)
			for(j=0; j<5; j++) {
				twoD[i][j] = k;	k++;
			}
		for(i=0; i<4; i++) {
			for(j=0; j<5; j++)
				System.out.print(twoD[i][j] + " ");	System.out.println();
		}
	}
}
```

Output:

```
0 1 2 34
5 6 7 8 9
10 11 12 13 14
15 16 17 18 19
```

Uneven Multidimensional Arrays

- The length of each array is under your control

```
int twoD[][] = new int[4][];

 twoD[0] = new int[1];

twoD[1] = new int[2];

twoD[2] = new int[3];

twoD[3] = new int[4];
```

```
class TwoDAgain{
 public static void main(String args[]) {
        int twoD[][] = new int[4][];
        twoD[0] = new int[1];
        twoD[1] = new int[2];
        twoD[2] = new int[3];
        twoD[3] = new int[4];
        int i, j, k = 0;
        for(i=0; i<4; i++)
          for(j=0; j<i+1; j++) {
            twoD[i][j] = k; k++;
          }
        for(i=0; i<4; i++) { for(j=0; j<i+1; j++) {
           System.out.print(twoD[i][j] + " ");
           System.out.println();
        }
   }
}
```

Output:

```
0
1 2
3 4 5
6 7 8 9
```

Alternative Syntax Array declaration

Syntax*:* *type[ ] var-name;*

```
int al[] = new int[3];
```

OR

```
int[] a2 = new int[3];
```

The following declarations are also equivalent:

```
char twod1[][] = new char[3][4];
```

```
char[][] twod2 = new char[3][4];
```

Multiple Arrays Delcaration

```
int[] nums, nums2, nums3; // create three arrays
```

```
int nums[], nums2[], nums3[]; // create three arrays
```

Week Review

- Primitive Data Types: (Integer, floating point, char and boolean)
- Literals: Integer, floating-point, and boolean and string literals
- Escape sequence, identifier
- Scope and life time of variables
- Conversion and casting, automatic promotion
- Arrays (One dimensional, Multidimensional), alternative ways to declare arrays

Questions?

Sukkur IBA

Thanks…