Level 6

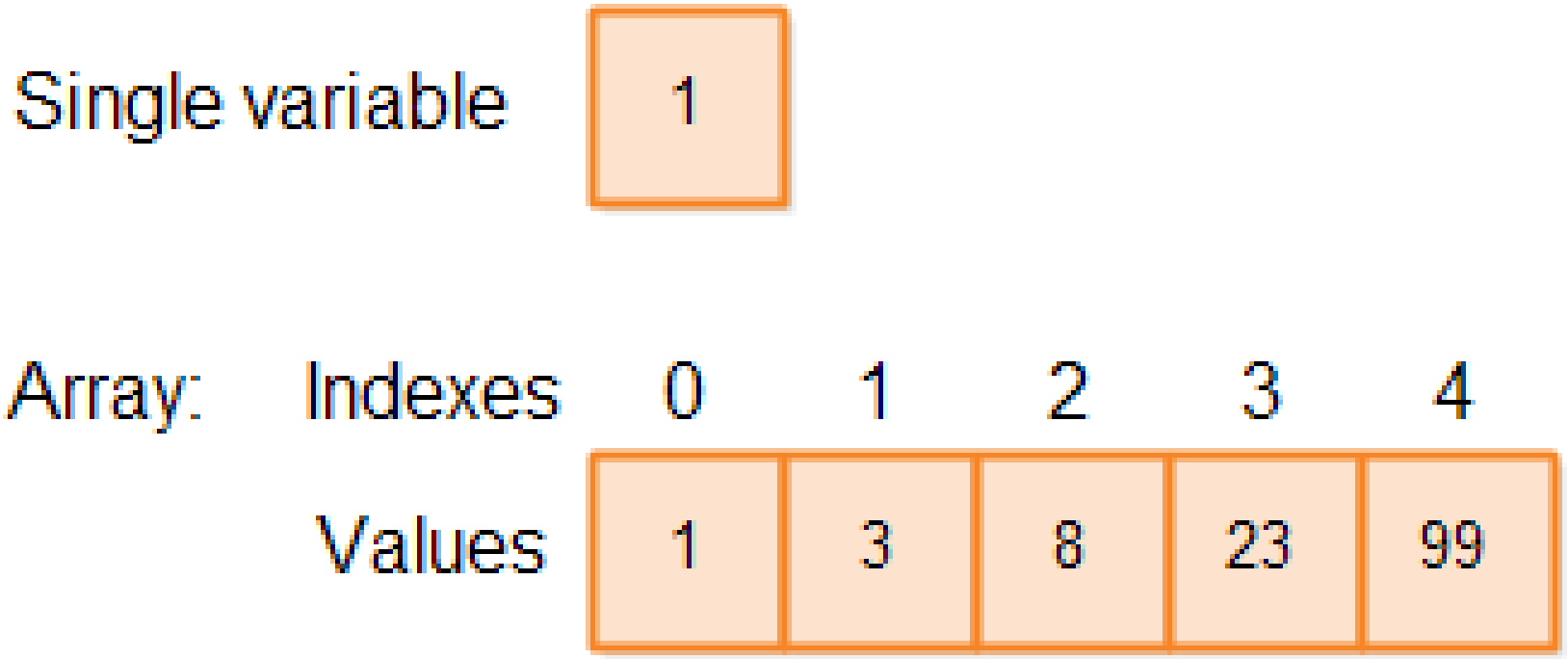
Java for Beginners

Mr. Teasdale

Levels of Java coding

- 1: Syntax, laws, variables, output
- 2: Input, calculations, String manipulation
- 3: Selection (IF-ELSE)
- 4: Iteration/Loops (FOR/WHILE)
- 5: Complex algorithms
- **6: Arrays/Linked Lists**
- 7: File management
- 8: Methods
- 9: Objects and classes
- 10: Graphical user interface elements

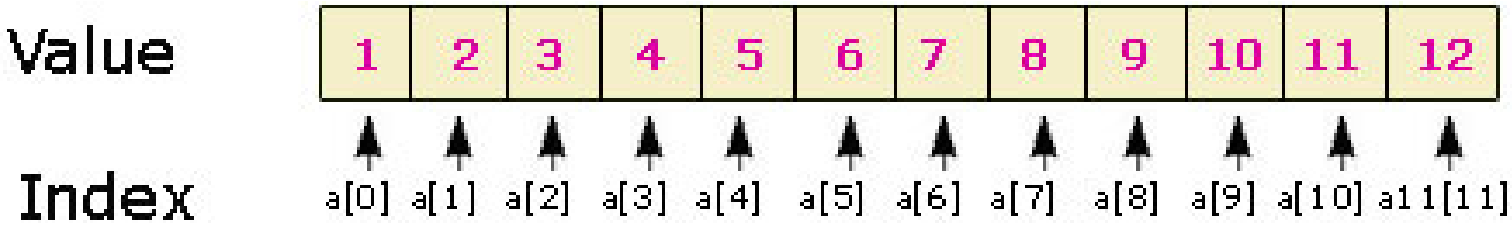
Arrays vs Variables



Initialising an array

One Dimensional array

Initialization `int a[] = new int [12];`




```
System.out.print(a[5]);
```

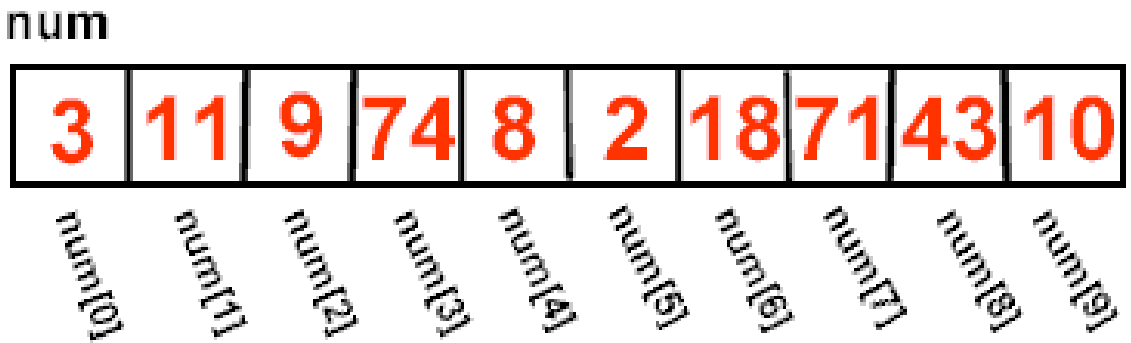
Output: 6

```
int [ ] num = new int [ 10 ];
```

type of each element

name of array

subscript (integer or constant expression for number of elements.)



	What could go wrong?
`num [ 0 ]`	always OK
`num [ 9 ]`	OK (given the above declaration)
`num [ 10 ]`	illegal (no such cell from this declaration)
`num [ -1 ]`	always NO! (illegal)
`num [ 3.5 ]`	always NO! (illegal)

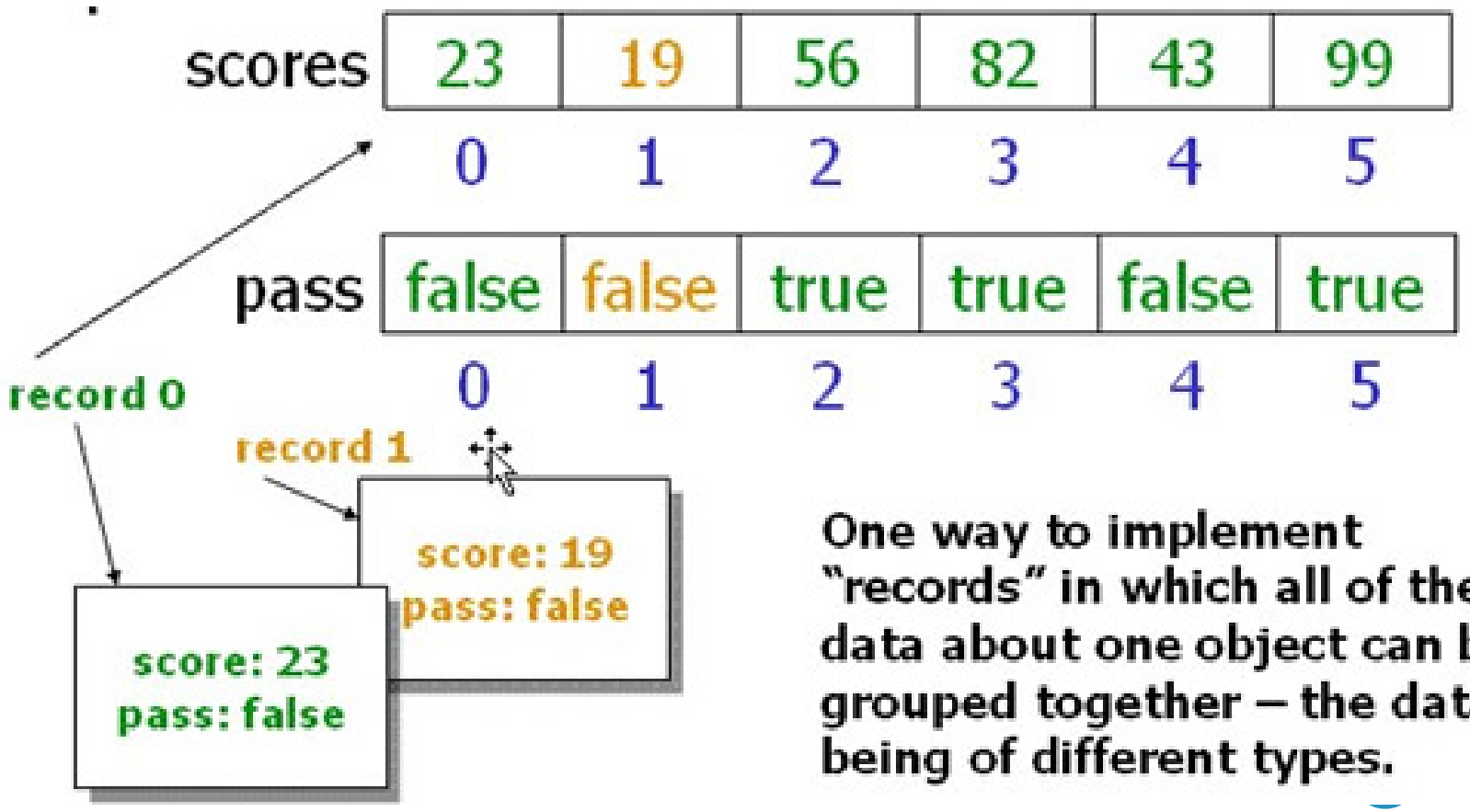
Array length

- When dealing with arrays, it is advantageous to know the number of elements contained within the array, or the array's "**length**". This length can be obtained by using the array name followed by **.length**. If an array named numbers contains 10 values, the code numbers.length will be 10.

  ** **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**.

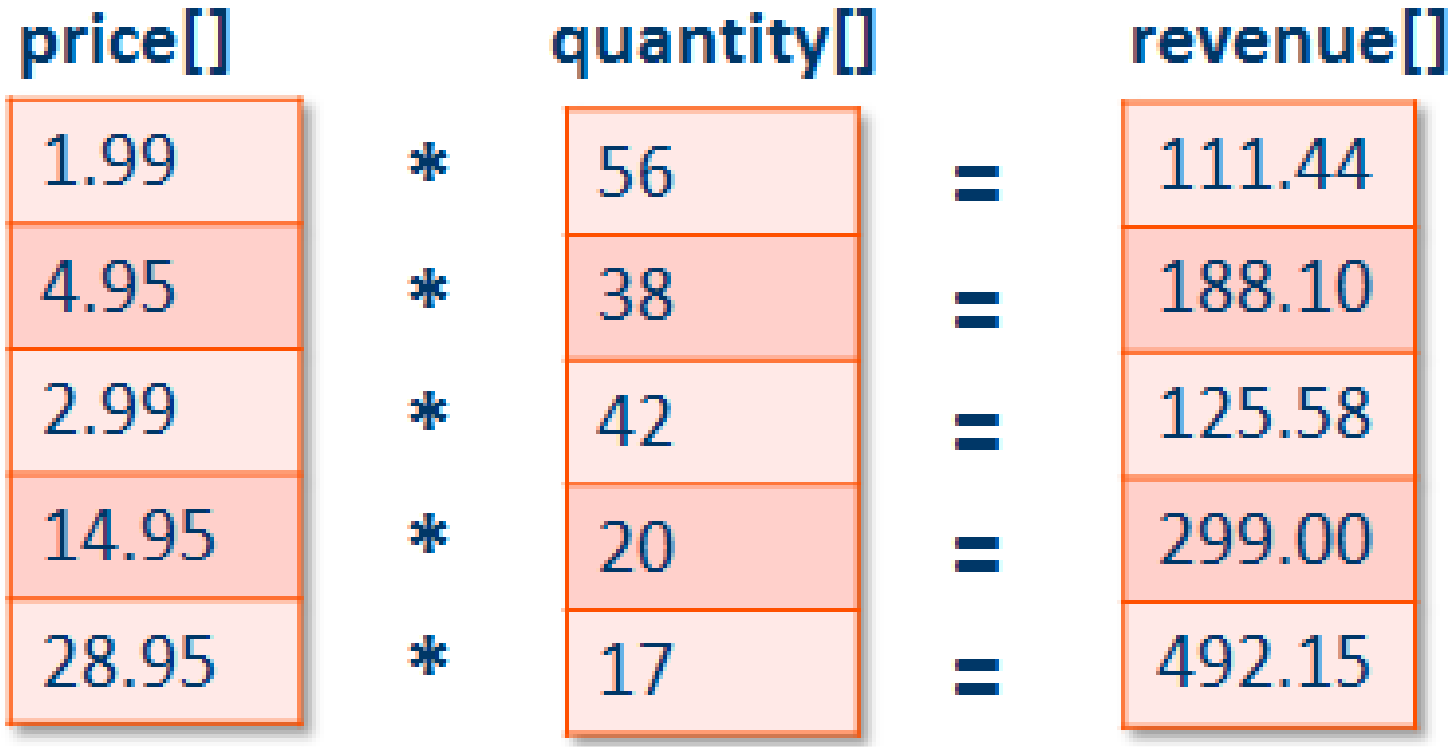
http://mathbits.com/MathBits/Java/arrays/Declare.htm

Parallel arrays



One way to implement "records" in which all of the data about one object can be grouped together – the data being of different types.

Parallel array applications



Parallel arrays in Java

```
for(int index = 0; index < dogname.length; index++)
{
     System.out.println(dogname[index]);
     System.out.println(round1[index]);
     System.out.println(round2[index]);
}
```

dogname

Wally	Skeeter	Corky	Jessie	Sadie

round1

18	22	12	17	15

round2

20	25	16	18	17

****The true beauty of parallel arrays, is that each array may be of a different data type.**

Searching an array using a *flag*

```
public class BreakBooleanDemo{
      public static void main(String[ ] args){
            int[ ] numbers = { 12, 13, 2, 33, 23, 31, 22, 6, 87, 16 };
            int key = 31;
            int i = 0;
            // set the boolean value to false until the key is found
            boolean found = false;
            for
                  ??
             }
            //When found is true, the index of the location of key will be printed.
            if (found){
                  System.out.println("Found " + key + " at index " + i + ".");
            }
            else{
                  System.out.println(key + "is not in this array.");
            }
       }
 }
```

Searching an array using a *flag*

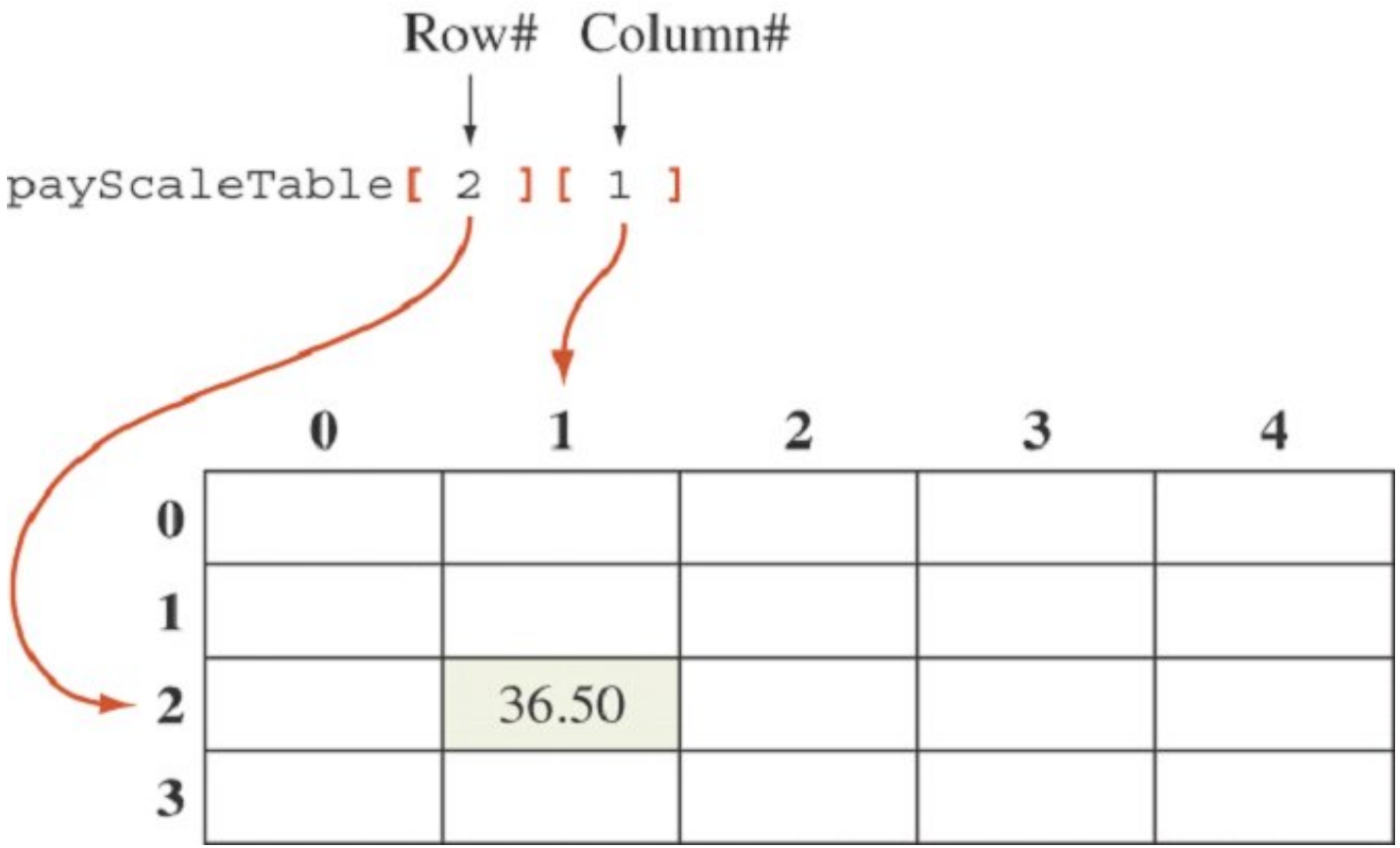
```
public class BreakBooleanDemo{
      public static void main(String[ ] args){
            int[ ] numbers = { 12, 13, 2, 33, 23, 31, 22, 6, 87, 16 };
            int key = 31;
            int i = 0;
            // set the boolean value to false until the key is found
            boolean found = false;
            for ( i = 0; i < numbers.length; i++){
                  if (numbers[ i ]  == key){
                        found = true;
                     }
             }
            //When found is true, the index of the location of key will be printed.
            if (found){
                  System.out.println("Found " + key + " at index " + i + ".");
            }
            else{
                  System.out.println(key + "is not in this array.");
            }
       }
 }
```

Sorting an array (*Bubble Sort*)

```
public static void BubbleSort( int [ ] num )
{
    int j;
    boolean flag = true;   // set flag to true to begin first pass
    int temp;    //holding variable

    while ( flag )
    {
        flag= false;    //set flag to false awaiting a possible swap
        for( j=0;  j < num.length -1;  j++ )
        {
            if ( num[ j ] < num[j+1] )   // change to > for ascending sort
            {
                temp = num[ j ];     //swap elements
                num[ j ] = num[ j+1 ];
                num[ j+1 ] = temp;
                flag = true;          //shows a swap occurred
            }
        }
    }
}
```

2D arrays



Declaring a 2D array in Java

```
int[ ][ ] arrNumbers = new int[6][5];
```

6 = number of rows (DOWN)

5 = number of columns (ACROSS)

	A	B	C	D	E
0	10	12	43	11	22
1	20	45	56	1	33
2	30	67	32	14	44
3	40	12	87	14	55
4	50	86	66	13	66
5	60	53	44	12	11

Instantiating a 2D array

NumPad.java

```
24
25      String[][] keyCaps =
26      {
27          { "7", "8", "9" },
28          { "4", "5", "6" },
29          { "1", "2", "3" },
30          { "0", "C", "." }
31      };
32
```

Example: Fill a 2D array with “X”

```
public class Example {

    public static void main(String[] args) {

        //making the 2D array
        String[][] table = new String[5][4];

        //get stuff in
        for (int row = 0; row < 5; row++) {
            for (int col = 0; col < 4; col++) {
                table[row][col] = "X";
            }
        }
```

Output

Nothing!
You only put data in, not printed it!

Common mistake: printing a 2D array

You can't just print the array name,

You have to print every element in the array separately!

```
public class Example {

    public static void main(String[] args) {

        //making the 2D array
        String[][] table = new String[5][4];

        //get stuff in
        for (int row = 0; row < 5; row++) {
            for (int col = 0; col < 4; col++) {
                table[row][col] = "X";
            }
        }

        System.out.println(table);
```

Output

```
<terminated> Example (1) [Java Application] C
[[Ljava.lang.String;@2a139a55
```

Correct way: printing a 2D array

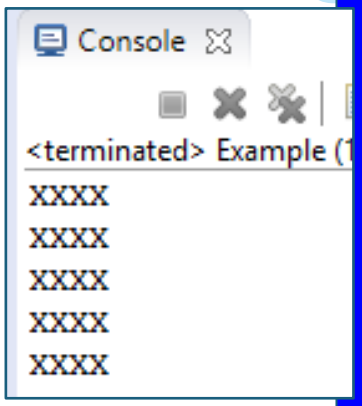
```
public class Example {

    public static void main(String[] args) {

        //making the 2D array
        String[][] table = new String[5][4];

        //get stuff in
        for (int row = 0; row < 5; row++) {
            for (int col = 0; col < 4; col++) {
                table[row][col] = "X";
            }
        }
        //get stuff out
        for (int i = 0; i < 5; i++)  {
            for (int j = 0; j < 4; j++)  {
                System.out.print(table[i][j]);
            }
            System.out.println();
        }
    }
}
```

Output



Common 2D array tasks

Task	Java Syntax	Examples
Declare a 2D array	`type[][] name`	`int[][] matrix` `Pixel[][] pixels`
Create a 2D array	`new type[nRows][nCols]`	`new int[5][8]` `new Pixel[numRows][numCols]`
Access an element	`name[row][col]`	`int value = matrix[3][2];` `Pixel pixel = pixels[r][c];`
Set the value of an element	`name[row][col] = value`	`matrix[3][2] = 8;` `pixels[r][c] = aPixel;`
Get the number of rows	`name.length`	`matrix.length` `pixels.length`
Get the number of columns	`name[0].length`	`matrix[0].length` `pixels[0].length`