ARRAYS

BME 121. FALL 2016

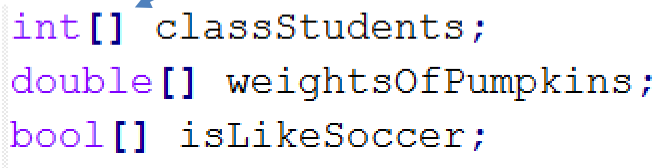
Outline:

- Quick review 1D array
  - Definition
  - Initialization
  - Assign value
  - Access elements
- 2D Arrays
  - Definition
  - Initialization
  - Assign Value
  - Access elements
- Examples
  - Printing contents of a 2D array.
  - Reinforce arrays, for loops, and while loops.

1D array

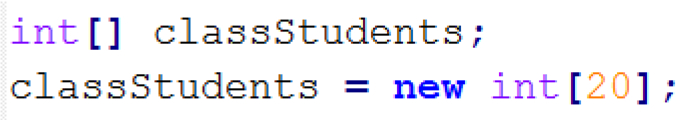
- Arrays
  - Collection of variables
  - organized
  - Same types

Symbol for definition of an array



```
int[] classStudents;
double[] weightsOfPumpkins;
bool[] isLikeSoccer;
```

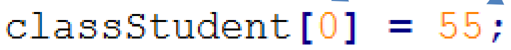
  - Allocate memory for array (make array)



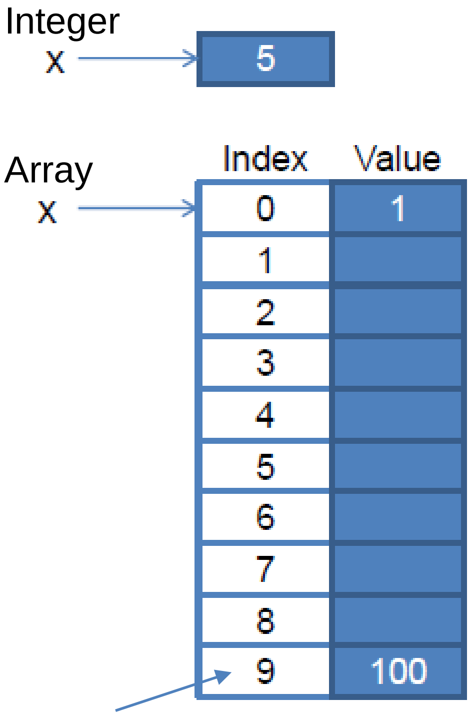
```
int[] classStudents;
classStudents = new int[20];
```

  - Assign a value to an element of array
  - Index start from zero

Index Value



```
classStudent[0] = 55;
```

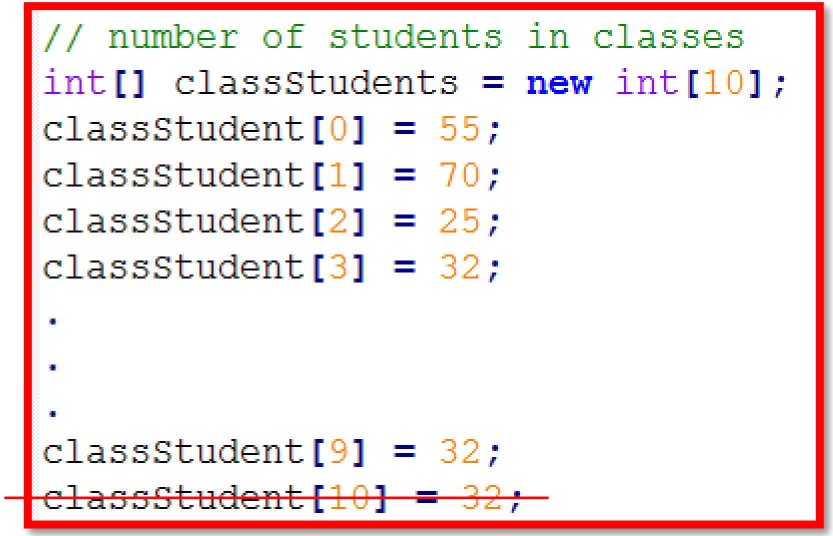


Index	Value
0	1
1	
2	
3	
4	
5	
6	
7	
8	
9	100

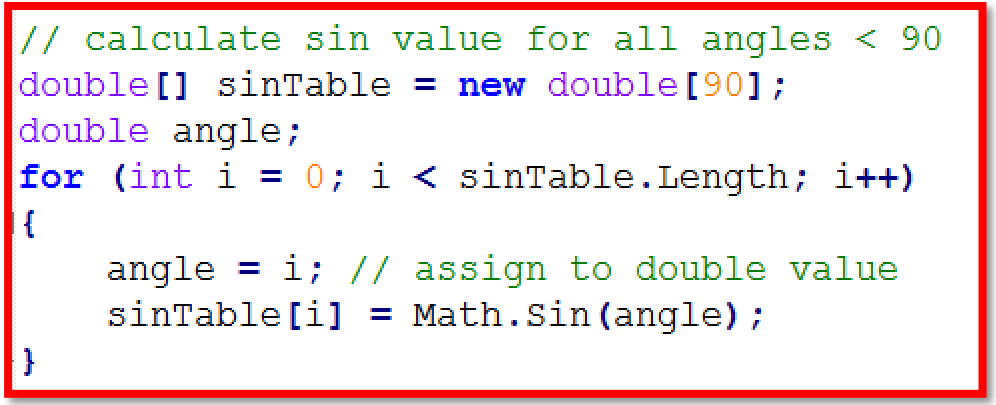
last index = length-1

1D array initialization

- Initialize all elements



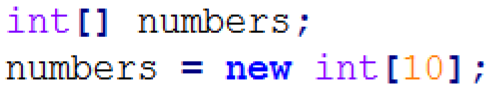
```
// number of students in classes
int[] classStudents = new int[10];
classStudent[0] = 55;
classStudent[1] = 70;
classStudent[2] = 25;
classStudent[3] = 32;
.
.
.
classStudent[9] = 32;
~~classStudent[10] = 32;~~
```



```
// calculate sin value for all angles < 90
double[] sinTable = new double[90];
double angle;
for (int i = 0; i < sinTable.Length; i++)
{
    angle = i; // assign to double value
    sinTable[i] = Math.Sin(angle);
}
```

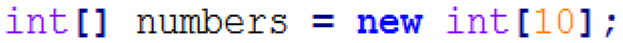
1D array: different initializations

- Creation



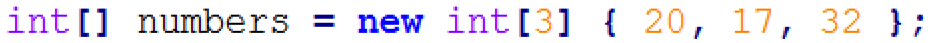
```
int[] numbers;
numbers = new int[10];
```

- Declaration and creation



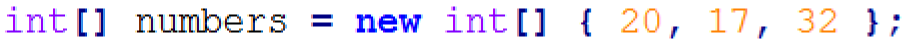
```
int[] numbers = new int[10];
```

- Filling in creation



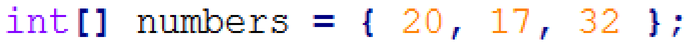
```
int[] numbers = new int[3] { 20, 17, 32 };
```

- Implicit size and filling



```
int[] numbers = new int[] { 20, 17, 32 };
```

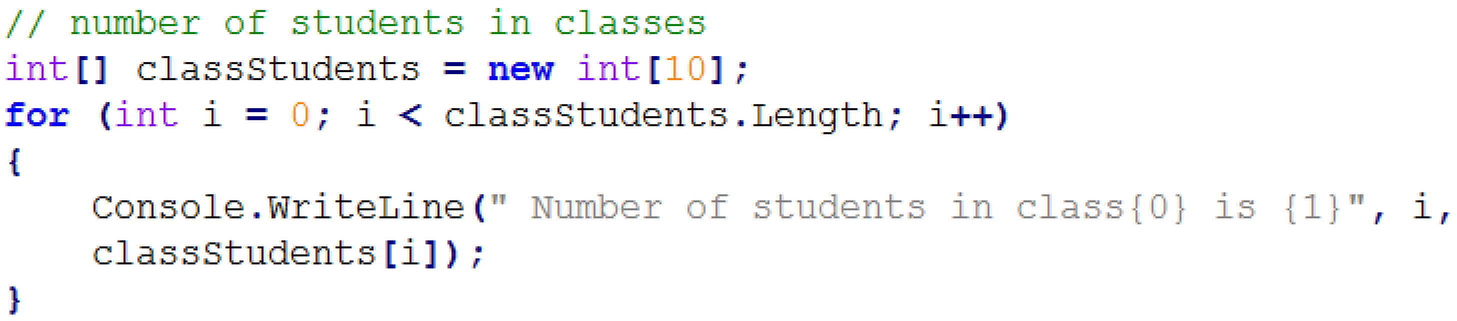
- Implicit creation



```
int[] numbers = { 20, 17, 32 };
```

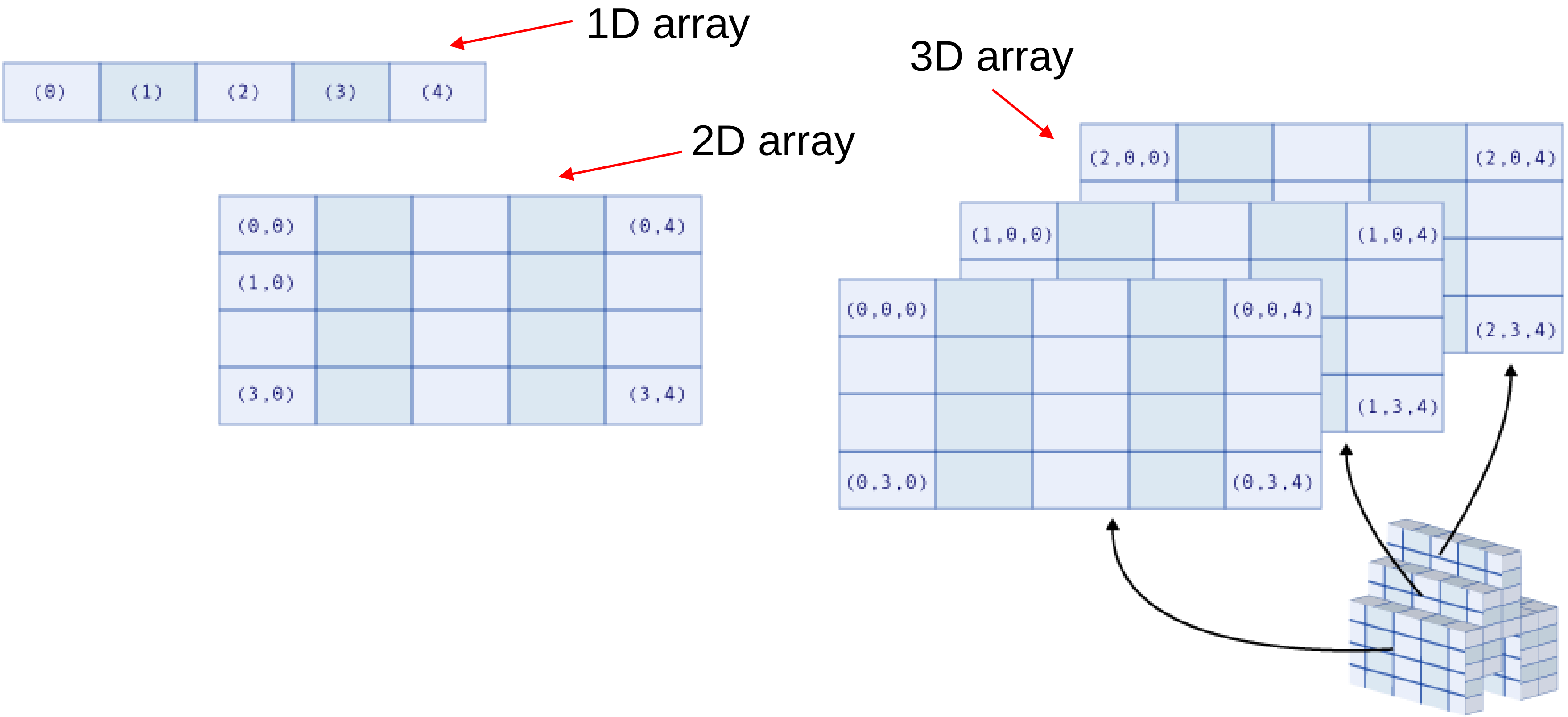
1D array: access the elements

- Index start from 0
- Last index Length-1



```
// number of students in classes
int[] classStudents = new int[10];
for (int i = 0; i < classStudents.Length; i++)
{
    Console.WriteLine(" Number of students in class{0} is {1}", i,
    classStudents[i]);
}
```

Multi-dimensional rectangular array

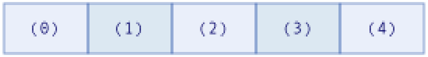


1D array

2D array

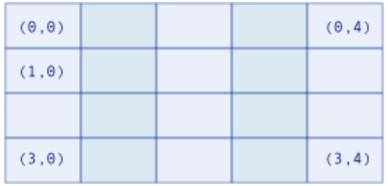
3D array

2D rectangular array



- Definition and creation

```
int[,] array = new int[4, 2];
```



- Implicit Initialization

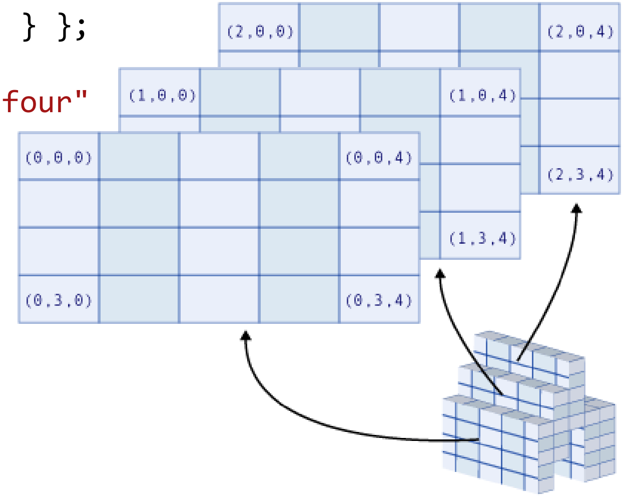
First row

Second Row

```
// Two-dimensional array.
int[,] array2D = new int[,] { { 1, 2 }, { 3, 4 }, { 5, 6 }, { 7, 8 } };
// The same array with dimensions specified.
int[,] array2Da = new int[4, 2] { { 1, 2 }, { 3, 4 }, { 5, 6 }, { 7, 8 } };
// A similar array with string elements.
string[,] array2Db = new string[3, 2] { { "one", "two" }, { "three", "four"
},
                                        { "five", "six" } };
```

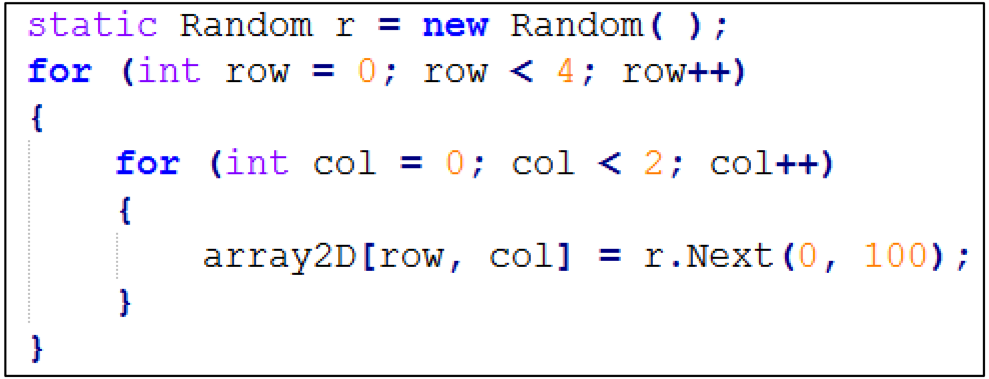
array2D

1	2
3	4
5	6
7	8



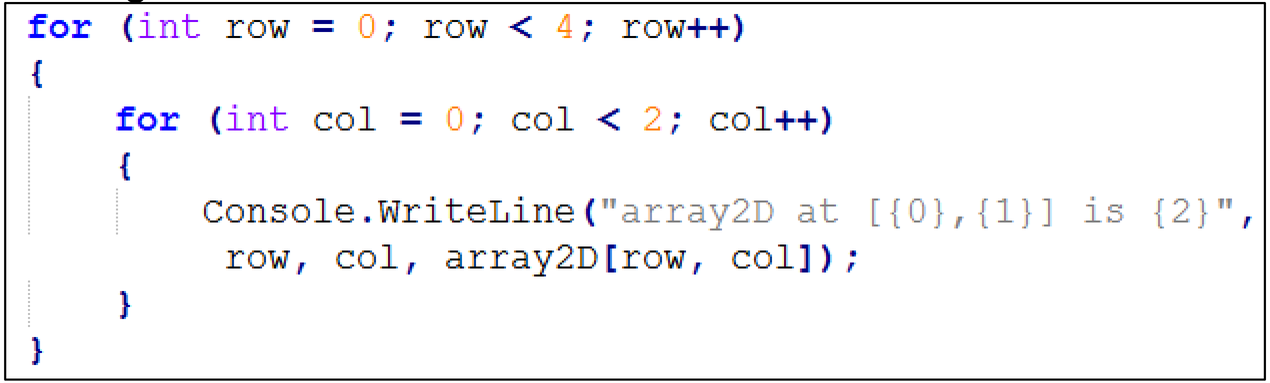
2D rectangular array

- Initialization with for loop



```
static Random r = new Random( );
for (int row = 0; row < 4; row++)
{
    for (int col = 0; col < 2; col++)
    {
        array2D[row, col] = r.Next(0, 100);
    }
}
```

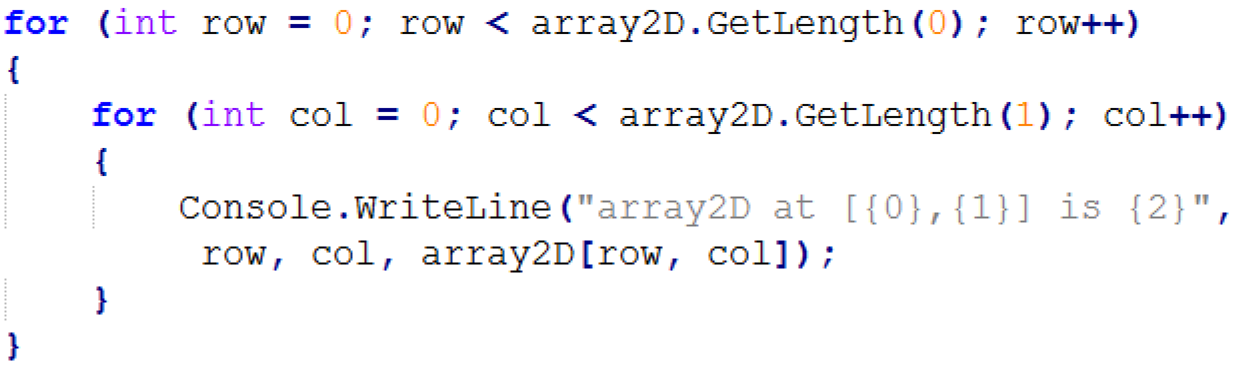
- Accessing elements



```
for (int row = 0; row < 4; row++)
{
    for (int col = 0; col < 2; col++)
    {
        Console.WriteLine("array2D at [{0},{1}] is {2}",
         row, col, array2D[row, col]);
    }
}
```

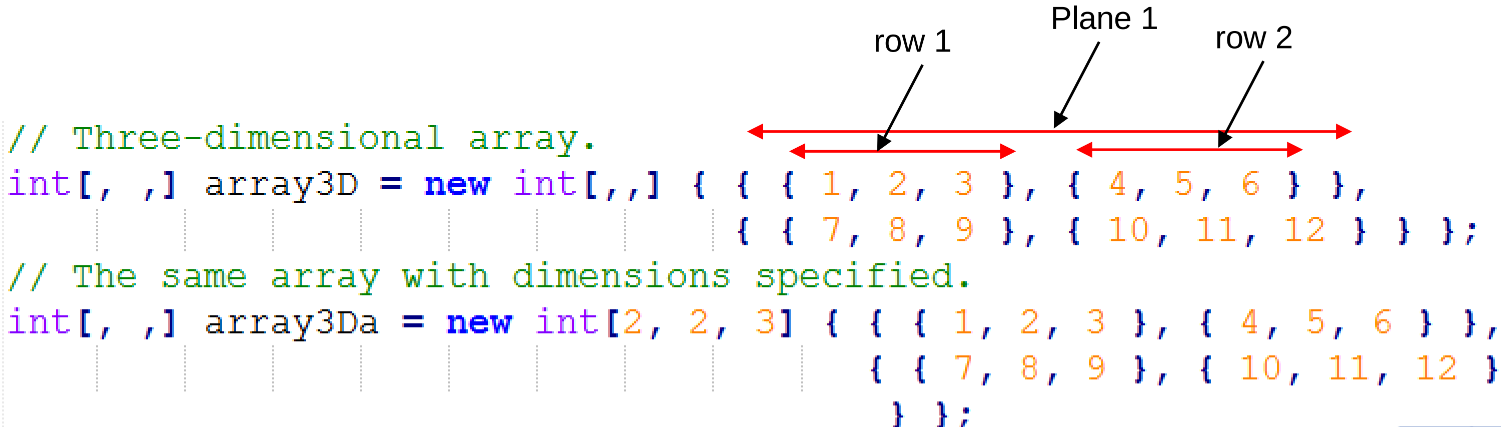
2D rectangular array

- Accessing elements
- GetLength(int dimension) get the length in the specified dimension

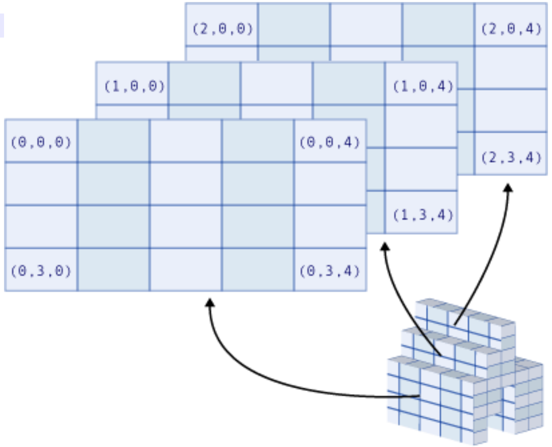


```
for (int row = 0; row < array2D.GetLength(0); row++)
{
    for (int col = 0; col < array2D.GetLength(1); col++)
    {
        Console.WriteLine("array2D at [{0},{1}] is {2}",
         row, col, array2D[row, col]);
    }
}
```

Multi-dimensional arrays



```
// Three-dimensional array.
int[, ,] array3D = new int[,,] { { { 1, 2, 3 }, { 4, 5, 6 } },
                                 { { 7, 8, 9 }, { 10, 11, 12 } } };
// The same array with dimensions specified.
int[, ,] array3Da = new int[2, 2, 3] { { { 1, 2, 3 }, { 4, 5, 6 } },
                                       { { 7, 8, 9 }, { 10, 11, 12 }
                                        } };
```



Exercise

- Write a program, that plots a histogram for a quiz given to a class of students.
  - The quiz is graded on a scale from 0 to 5.
  - Use an array to count the number of 0’s, 1’s, 2’s, 3’s, 4’s, and 5’s.
  - The program should be able to handle an arbitrary number of student grades.
- Tips:
  - You can make an array of size 5, where each element is initialized to zero.
  - Whenever a value in [0,1) is entered, increment the value of the array index 0. Whenever a value in [1,2) is entered, increment the value of the array at index 1… and so on, up to index 4 of the array.

Taken from: Savitch, Walter J. *Absolute C++*. Pearson Education, 2006.

THE END