PROGRAMMINGS SKILLS II

~ ANURUDDHA ABEYSINGHE

Lecture 09

Foundation Certificate in IT – Curtin Batch

ARRAYS

LECTURE 09

WHY WE NEED OF AN ARRAY?

Age → 10 , 20 , 25 , 35 , 50

Initialize the data

```
int age1 = 10;
int age2 = 20;
int age3 = 25;
int age4 = 35;
int age5 = 50;
```

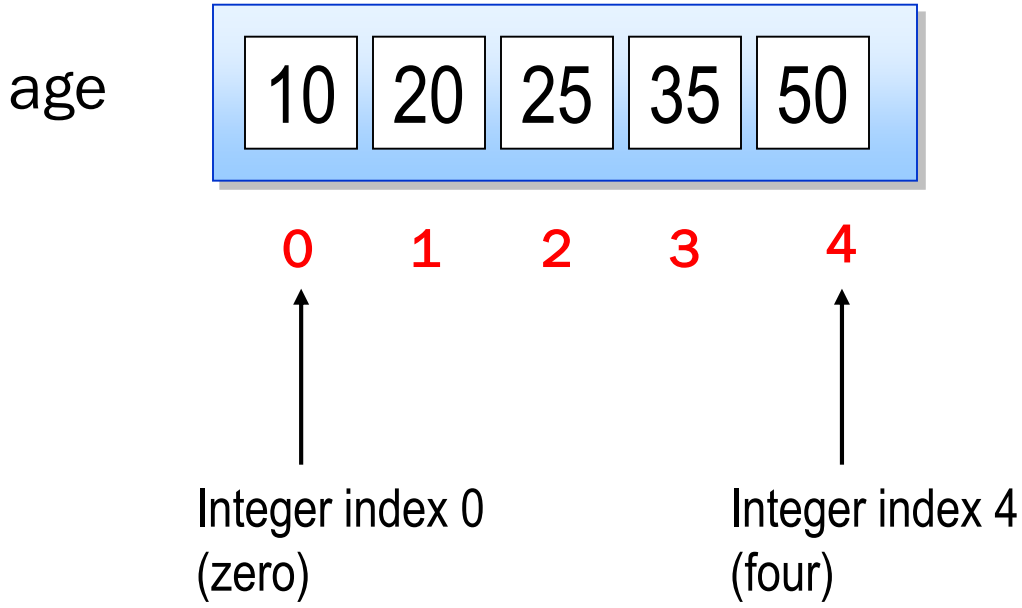
Print the data

```
Console.WriteLine(age1);
Console.WriteLine(age2);
Console.WriteLine(age3);
Console.WriteLine(age4);
Console.WriteLine(age5);
```

This method may fine with less number of data. But think about 100 data?? 1000 data??

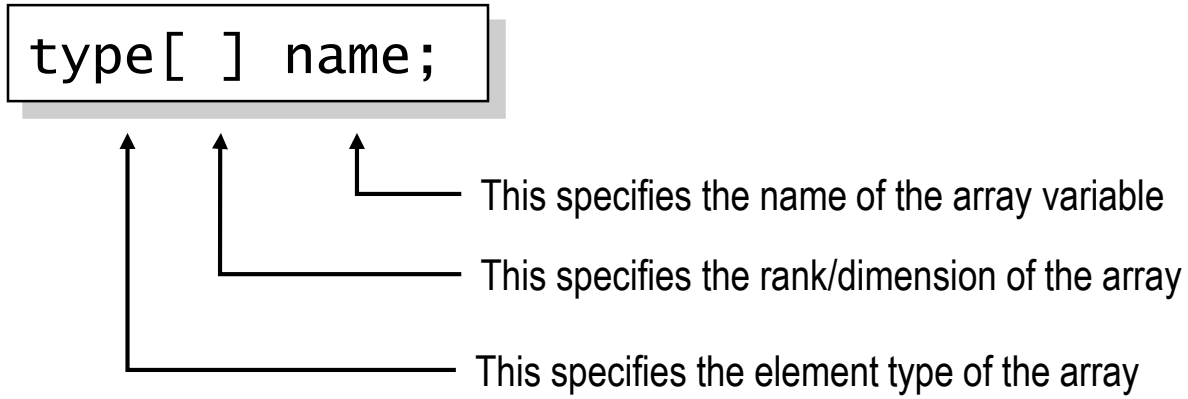
WHAT IS AN ARRAY?

- An array is a sequence of elements
  - All elements in an array have the same data type
  - Individual elements are accessed using integer indexes



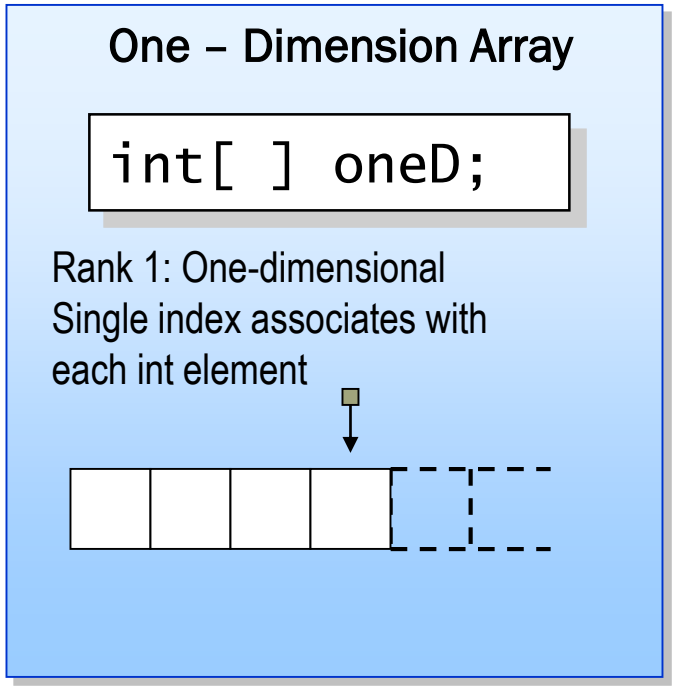
ARRAY NOTATION IN C#

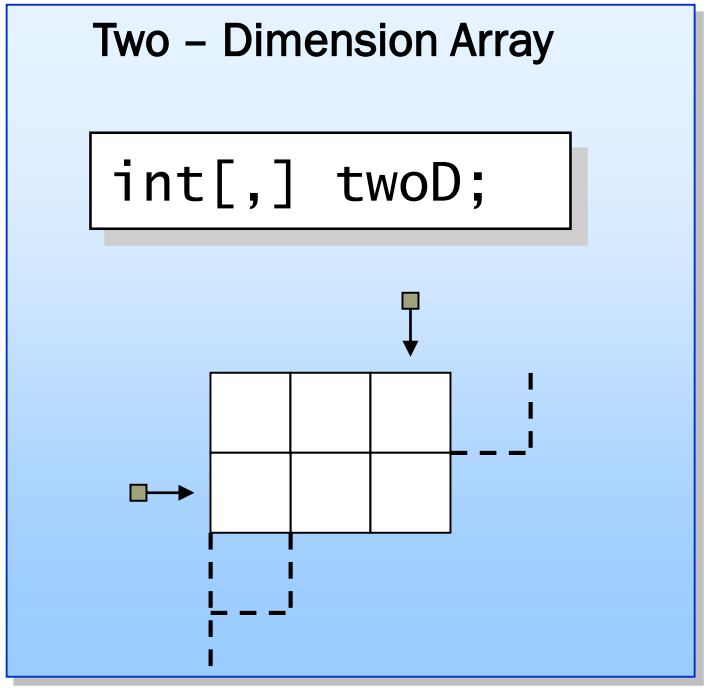
- You declare an array variable by specifying:
  - The element **type of the array**
  - The **rank/dimension** of the array
  - The **name** of the variable



ARRAY RANK

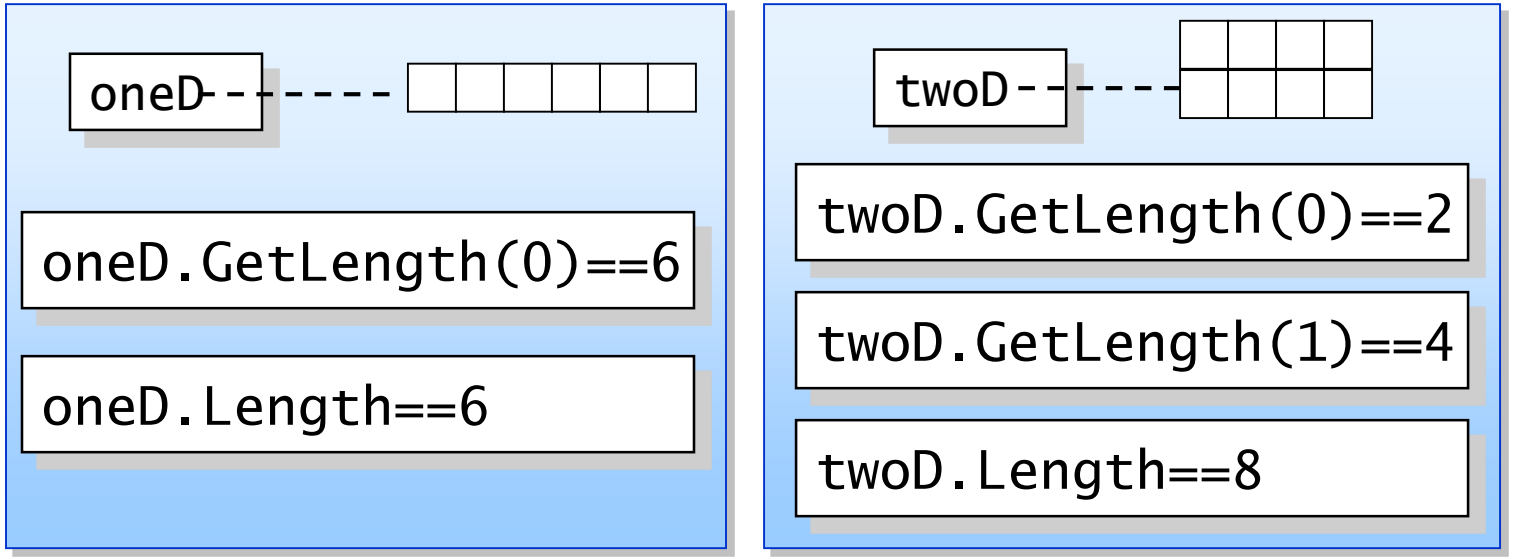
- Rank is also known as the array dimension
- The number of indexes associated with each element





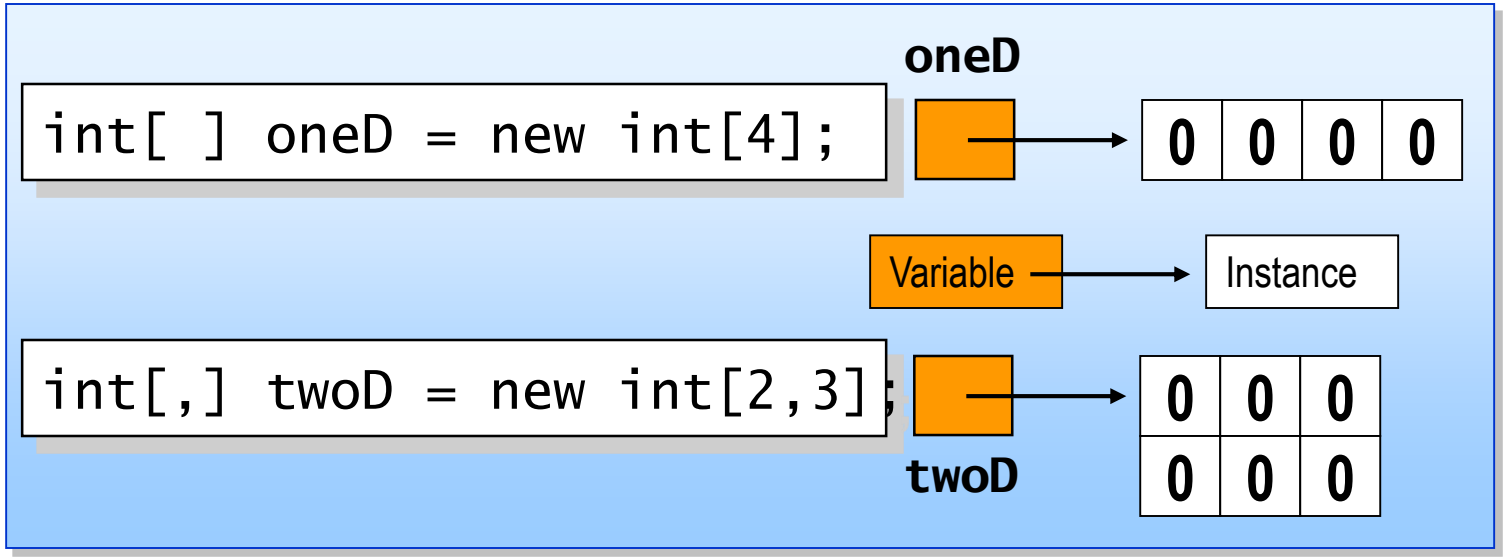
CHECKING ARRAY BOUNDS

- All array access attempts are bounds checked
  - A bad index throws an IndexOutOfRangeException
  - Use the **Length** property and the **GetLength** method



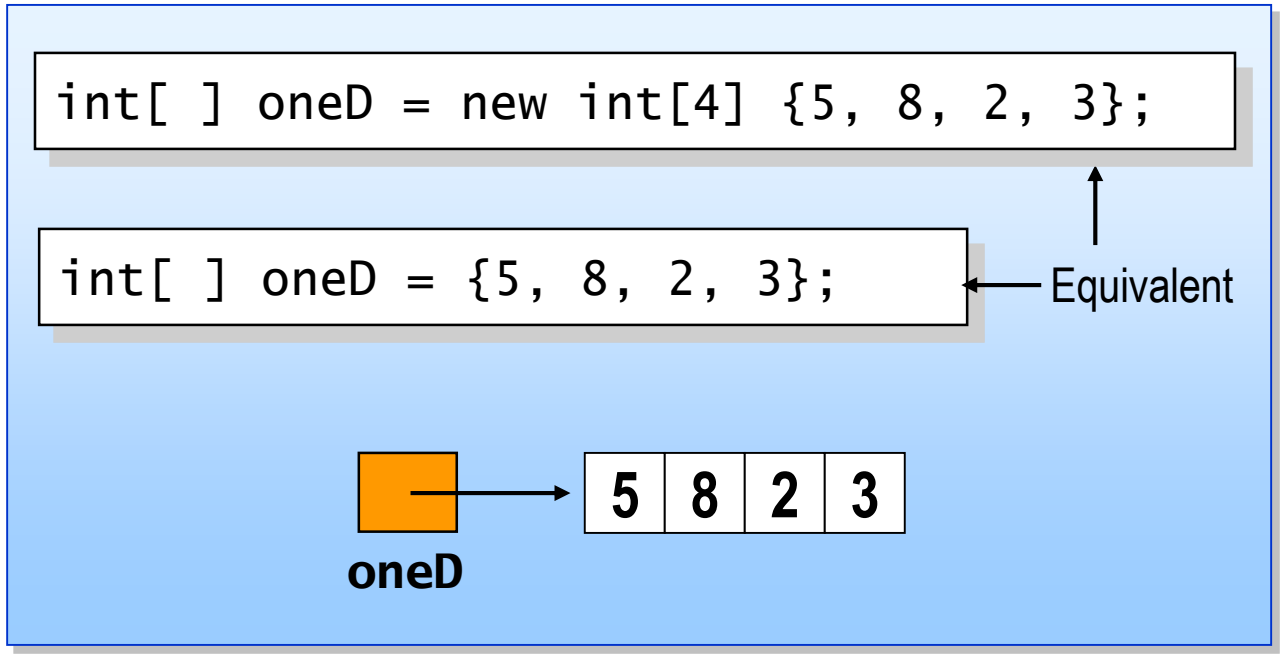
CREATING ARRAY INSTANCES

- Declaring an array variable does not create an array!
  - You must use **new** to explicitly create the array instance
  - Array elements have an implicit default value of zero



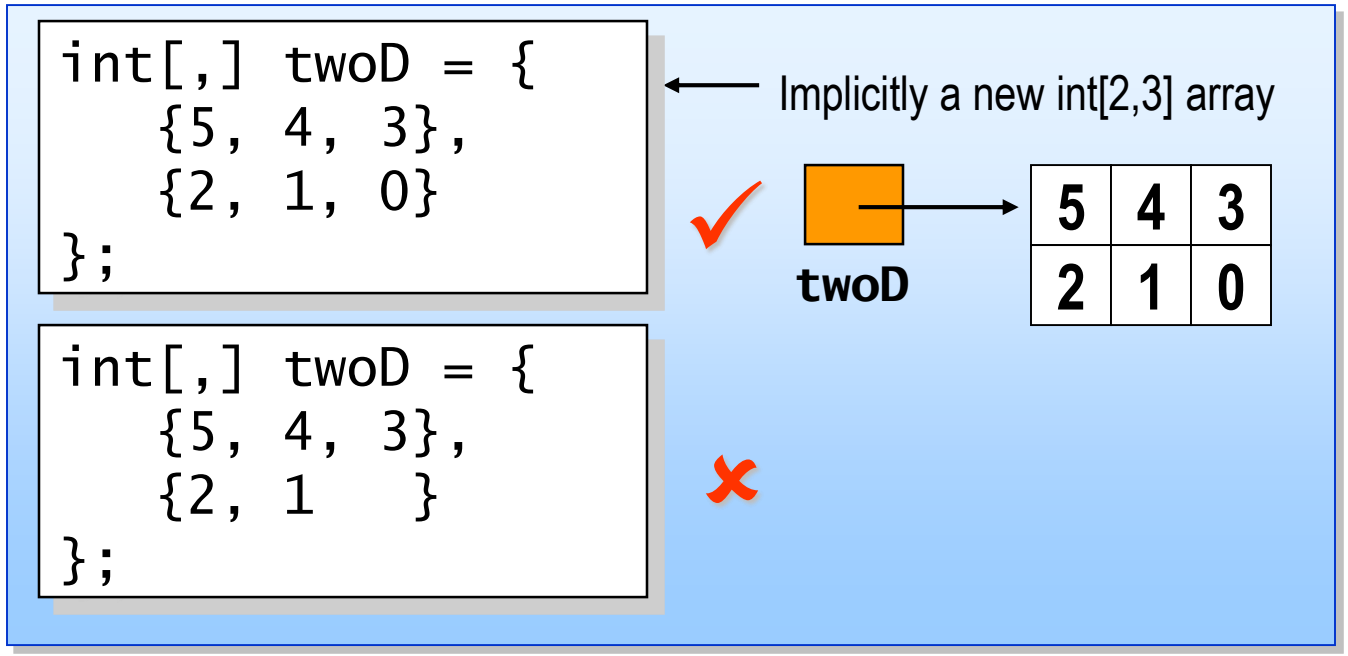
INITIALIZING ARRAY ELEMENTS

- The elements of an array can be explicitly initialized
  - You can use a convenient shorthand

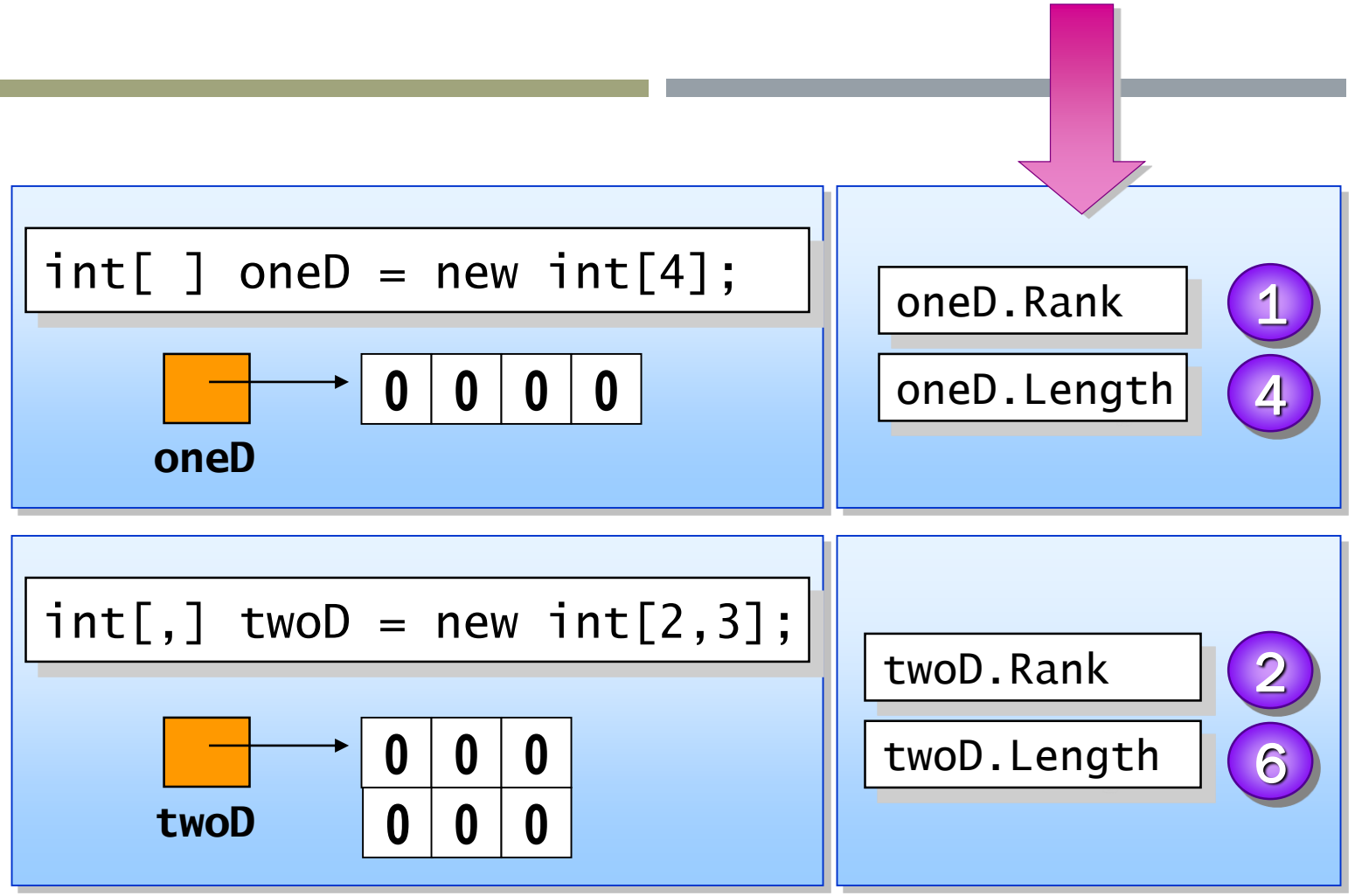


INITIALIZING MULTIDIMENSIONAL ARRAY ELEMENTS

- You can also initialize multidimensional array elements
  - All elements must be specified



ARRAY PROPERTIES



USING ARRAYS WITH FOREACH

- The foreach statement abstracts away many details of array handling

```
int[] arr = new int[4] {2,5,7,8};


foreach (int i in arr)
{
   Console.WriteLine(i);
}
```

Output:

```
2
5
7
8
```

ARRAY METHODS

- Commonly used methods
  - **Sort** – sorts the elements in an array of rank 1
  - **Clear** – sets a range of elements to zero or **null**
  - **GetLength** – returns the length of a given dimension
  - **IndexOf** – returns the index of the first occurrence of a value

THANK YOU

SEE YOU ON
NEXT WEEK