

# Using Arrays and Lists



**Gill Cleeren**

CTO Xebia Microsoft Services Belgium

@gillcleeren

# Overview



**Understanding arrays**

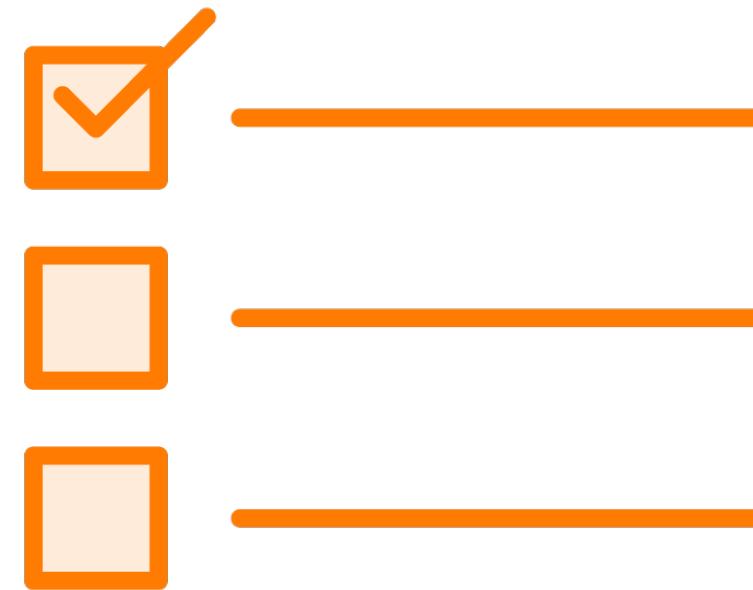
**Working with arrays**

**Working with collections**



# Understanding Arrays





### List of

- Employees
- Songs
- Records in a database
- ...

### Impossible to just use variables

- Amount of items coming in is unknown



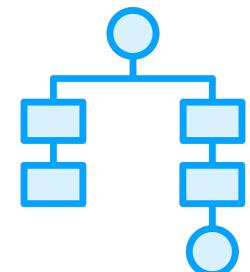
# Working with Lists of Data

Arrays

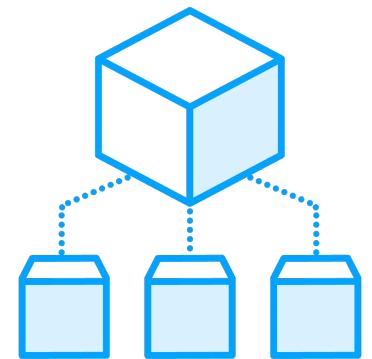
Collections



# Arrays in C#



**Data structure to store multiple variables**



**All variables must have the same type (can be object)**

`[1,2,3]`

**Accessed through use of index**



# Creating an Array Variable

Type of array      Variable name

`int[]`      `allEmployeeIds;`

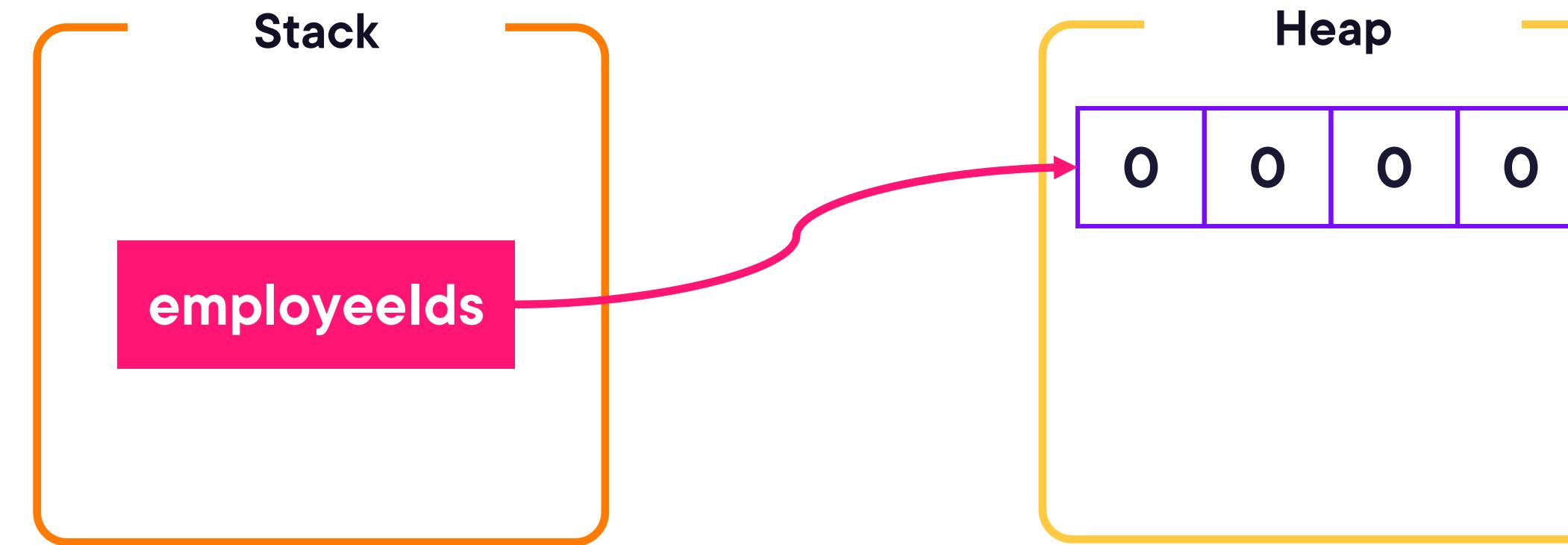
`DateTime[] startDates;`



# Instantiating the Array

int[] allEmployeeIds = new int[4];

Array to contain 4 ints



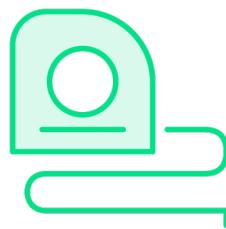
# Instantiating the Array



**Arrays are reference types, even for value types used in the array**



**Creation happens upon using new**



**Size is set upon creation of the array, but can be at runtime**



**Arrays are zero-based**



```
int size = int.Parse(Console.ReadLine());
```

```
int[] employeeIds = new int[size];
```

## Determining the Array Size at Runtime



# Populating the Array

```
int[] allEmployeeIds = new int[4] {11, 44, 179, 161};  
  
int[] managerIds = new int[4] {11, 44, 179};  
  
int[] supportStaffIds = new int[] {11, 44};
```



# Accessing Elements within the Array

```
allEmployeeIds[0] = 123;
```

```
int firstEmployeeId = allEmployeeIds[0];  
int secondEmployeeId = allEmployeeIds[1];
```

```
allEmployeeIds[2] = 33;
```

```
allEmployeeIds[7] = 33; //runtime error
```



# Demo



**Creating an array**

**Accessing elements in an array**

**Looping through an array**



## Demo



**Working with an array of Employee objects**



# Working with Arrays



# The Array Base Class

**CopyTo()**

**Sort()**

**Reverse()**

**Length**



# Demo

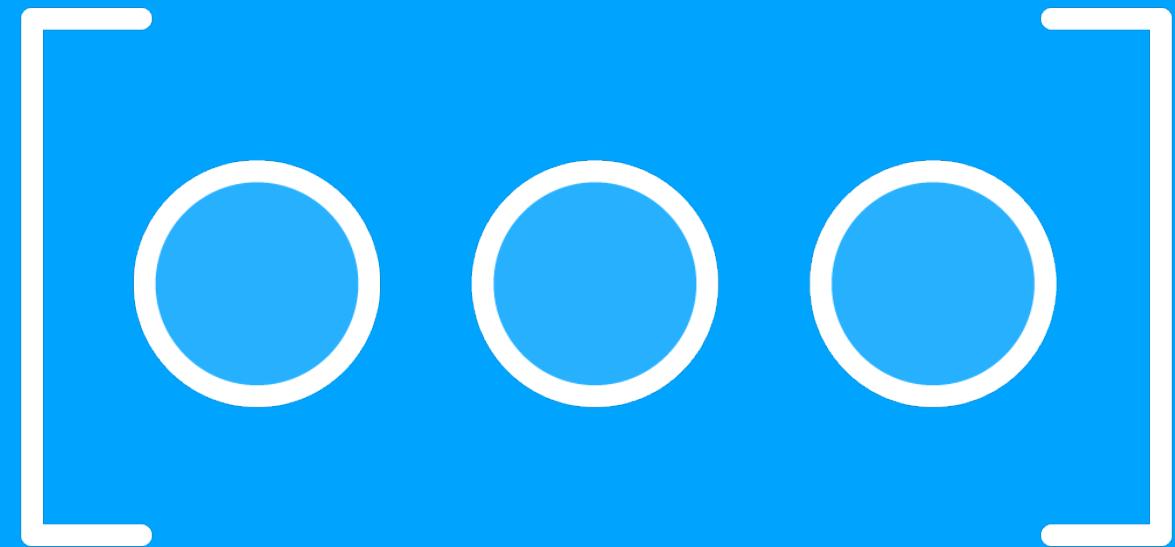


## Working with arrays



# Working with Collections





**Arrays are somewhat limited...**

**Changing the size is hard**

**Accessing the items is limited**

**Collections can be a better solution!**



```
List<int> employeeIds = new List<int>();
```

## Creating a List

Can contain int values



```
List<int> employeeIds = new List<int>();  
  
employeeIds.Add(1);  
employeeIds.Add(99);  
employeeIds.Add(458);  
  
employeeIds.Remove(1);  
  
int selectedId = employeeIds[2];
```

## Working with the List<T>

**Adding items is done using a method**



```
int length = employeeIds.Count;
```

## Lists Know Their Length



# Lists are Type-safe

```
List<int> employeeIds = new List<int>();  
employeeIds.Add(new Employee());
```



# Demo



## Working with the Collection classes



# Summary



**Arrays allow us to work with simple sequences**

**Are reference types**

**Collection classes are more flexible**



**Up Next:**

# **Introducing inheritance and other OO principles**

---

