



## L1-2: Intro

≡ SESSION	Session 1-2
▼ CLASS MODE	LECTURE
📎 SLIDES	<a href="#">Algorithms and Datastructures Session 1-2 (1).pdf</a>
☑ Completed	<input type="checkbox"/>
📅 Date	@October 4, 2022

### ▼ Professor

- Austrian
- EDU:
  - BSc Business Informatics
  - MBA in Financial Management
  - Master in computer science
  - Executive Master in Positive Leadership, Strategy and Transformation (IE)
- Work:
  - 30 years IT
  - 20 years Development
  - 10 years as CEO/Owner
- 3 years teaching at universities (Spain and Austria)

### ▼ What if I fail?

- This class is about learning about data structures and algorithms
- Use your own computer, write for yourself, even try more things if you get comfortable.

### ▼ Rules:

- Always be on time (and present)
- Interrupt when you have a question or different opinion
- Ask/Remind me for a break if I forget it
- Contribute → chocolate time

### ▼ The content

1. Introduction/terms
2. Array
3. Stack
4. Queue
5. Linked Lists
6. recursion
7. Tree
8. Bubble sort
9. Selection sort
10. Modern languages and the support of data structures

### ▼ The journey

- It is your journey → take it
- The journey ends:
  - after 1800 minutes
    - 1200 in class
    - 600 beside that
- Group works for discovery
- Class discussions
- Adding knowledge step by step
- Whenever you think you are 'lost', can help to ASK! Maybe is part of the journey.

### ▼ The slides

Slides for the direction

You get them after class

When there is a task and you will need the content from before you will get the slides with the task.

# Introduction

## Data



### Data:

- “Facts and statistics collected together for reference or analysis”
- “The quantities, characters, or symbols on which operations are performed by a computer, which may be stored and transmitted in the form of electrical signals and recorded on magnetic, optical, or mechanical recording media”
- Data is a sequence of symbols: this is data, 12345, ##+
- Information is data with:
  - a meaning
  - a context

→ “The purpose of computing is insight, not numbers”

## Data type

- In mathematics we use types for variables (e.g. integer, real, sets, etc.)
- Abstract data type (ADT)
- The datatype in programming language defines the set of values:
  - a constant
  - a variable
  - the result of a function/operator
- Primitive datatypes are provided by the programming language (e.g. int)
- User defined datatypes allow the extension (e.g. struct concept in C#)

## Data structure



### Data structure:

- “An arrangement of data in memory locations to represent values of the carrier set of an abstract data type”
- “Is a particular way of storing and organising data in a computer so it can be used efficiently”

→ Support certain operations, each with a:

- meaning: what does the operation do/return

- performance: how efficient is the operation

## Algorithms

- “A high level, language-dependent description of a step-by-step process”
- An algorithm is the step-by-step unambiguous instructions to solve given problems”
- inputs, outputs, guaranteed to terminate and produce correct result
- e.g.:
  - Sorting a list of words alphabetically
  - Finding the fastest path between two points on a map

### ▼ Lets get practical

You receive a deck of cards. Find the highest cars!

Rules:

- you can only look at the top of the card of the deck
- you are allowed to put one card aside
- after each card you take from the top you have to return one card to finished deck

Write the algorithm for the task:

```
array = [4, 1, 3, 2, 3, 2]
max = array[0]
i = 1
while (i < len(array)):
    if card > max:
        max = card
    i+=1
return max
```

## Intro to code

### ▼ Creating a project in VisualStudioCode

1. Open Visual Studio Code
2. Click "open folder" and create a new folder where your files (code) will be stored, e.g. "Lectures"
3. "yes, trust" (it is your folder, you should trust it hahaha)
4. In Visual Studio Code, click "Terminal", "New Terminal"
5. Type dotnet new console --framework net6.0
6. Now click on the left side on "Program.cs" (our main file)
7. Click on "Console"

8. Click the bulb, "Convert to
9. Program.Main style program"
10. Click on "Run", "Add Configuration"
11. Select ".NET 5+ and .NET Core"
12. Finally, click "Run", "Start without debugging"

## C# basics

```
int x = 5;
int y = "A";
int x = 6; int y = x;
int z;
string s = "hello";
char c = 't';
char n = '1';
int z = x + 3;

Console.WriteLine("some");
Console.WriteLine(z);
Console.WriteLine("z: " + z);
```

## C# methods

```
public int MultiplyWithTwo(int number) #accessibility + return data type(or void) + name of method + input parameter(s)
{ #start of method body
    int result = number * 2;
    return result; #return value - mandatory if return type specified (not if void)
} #finish of method body

public int MultiplyWithTwo(int number){
    return number*2;
}
```

## Arrays (basically used almost everywhere)



### Array:

→ An array consists of components which are all of the same type, called its base type.

- each element of an array is called **“item”** or **“element”**
- the access to an element is done with the position in the array
- an array is a **random access structure** (we will understand later when we create e.g., linked list)
- on creation of the array a fixed space memory is reserved, the position of each element can be calculated (because of the fixed size of the elements), **the array has a fixed size** (number of elements)

## Consequences of arrays

- Because of the upfront allocation of memory
  - **Resize is not possible** (e.g. adding another element)
- Access with position (also called index) (e.g. x[1])
  - Because it is done with integer numbers, we can also calculate the position
  - Commonly one of the places where you will find bugs in software systems
- The index starts with **0**

## Arrays in C#

```
int[] array = new int[10]; #declaration of an integer array
                        |
                        size of the array

array[0] = 1;
array[1] = 17;
array[10] = whatever; #the terminal will give you an error because you have accessed
                        #an index outside what you allocated (max index 9 for size 10)
```

## Accessing arrays with loops in C#

```
for(int i = 0; i<10; i++){
    Console.WriteLine("pos " + i + ":" + array[i]); #position of the value and the value
}
```

## Array operations

- Initialise
- write a value into the array
- read a value from the array
- iterate the array (go from one index to the next)
- property "Length" can be used (some languages use size)
- resize?
  - we learned that the computer allocates a fixed space for the array
  - how can we now make the array bigger or smaller?

## Types of arrays

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

int[][] multi = new int[5][];
for (int i = 0; i<multi.Length; i++){
    multi[i] = new int[3]
}
```

## Declare / Initialise

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

int[] direct = new int{1,2,3,4,5,6,7};

int[] copy = direct; #careful it is the same as python, points to the same array
                     #if you change the values of direct, the values of copy
                     #change as well
```

### ▼ Test our assumption (changing direct and seeing copy)

```
private static void Main(string[] args)
{
    int[] array1 = new int[2];
    int[] array2 = array1;

    for(int i = 0; i<2; i++){
        array1[i] = 1;
    }

    for(int i = 0; i<2; i++){
        Console.WriteLine(array1[i]);
    }
    for(int i = 0; i<2; i++){
        Console.WriteLine(array2[i]);
    }
}
```

### ▼ Assessment (finding the minimum value of an array)

```
internal class Program
{
    private static int MinValue(int[] input){
        int min = input[0];
        for (int j=0;j<input.Length; j++){
            if (min > input[j]){
                min = input[j];
            }
        }
        return min;
    }

    private static void Main(string[] args)
    {
        int[] myarray = new int[7];

        for(int i = 0; i<myarray.Length; i++){
            myarray[i] = myarray.Length-i;
        }

        for(int i = 0; i<myarray.Length; i++){
            Console.Write(myarray[i]);
        }

        Console.WriteLine("The minimum value in the array is: " + MinValue(myarray));
    }
}
```