C# Fundamentals

Preparation 3 – data structures

CliZ Ware | LEARNING

Last time

- Relational operators
- If statements
- Logical operators
- *For loops*, *while*, and *do while*

Today

- We will focus on how you can store different kinds of values, and the concept behind it.

Introduction to variables

- A *variable* is a letter or a name that can store a value.

- You can store a number: height of a building; or a word: a persons name.

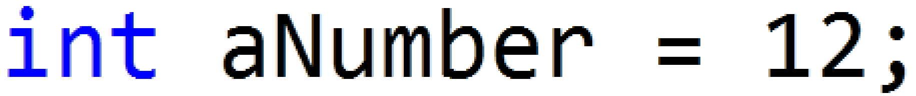
- These variables can also change their value.

Usage

- 1) **Declare the variable** – tell the program the name and kind of variable you want to use.
- 2) **Assign the variable** – give the variable a value to hold.
- 3) **Use the variable** – retrieve the value held in the variable and use it in your program.

Declare a variable

- You need to decide a *name* and a *data type*.



```
int aNumber = 12;
```

The *data type.*

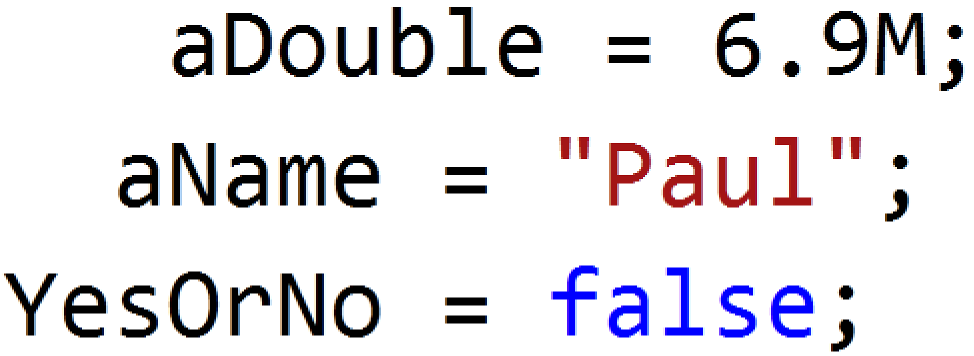
The *name*

The *value*

May only contain **integers**

Assigning variables

- In order to assign a variable, add an equal sign directly after the name, and then the value you want to assign.



```
aDouble = 6.9M;
aName = "Paul";
YesOrNo = false;
```

Other variables

- An **integer** can only store whole numbers.

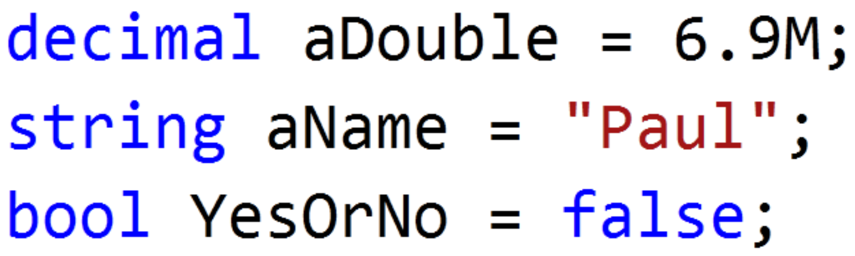
$$n \mid -2147483648 < n < 2147483647, n \in \mathbb{Z}$$

- So if you would like to store 42.5, you should use **Double**. If you would like to store text, use **String**. But, if you only need to store values *True* or *False*, you might use **Boolean**.

Declaring and assigning default value

- As you know, in order to assign a variable, you have to *declare* it first, then, on the second line, you assign it a value. *That might cause an error if the variable is used before it is* assigned.

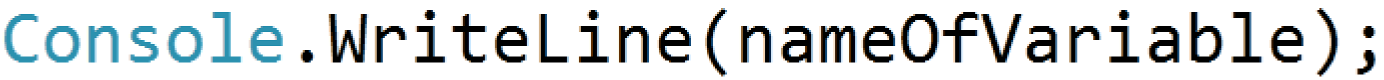
- Why not assign the variable by default?



```
decimal aDouble = 6.9M;
string aName = "Paul";
bool YesOrNo = false;
```

Display a value hold by a variable

- In **Console Application** development environment, type:



```
Console.WriteLine(nameOfVariable);
```

Data types

Int32 (or int)	32 bit integer
Int64 (or long)	64 bit integer
Boolean (or bool)	True or false
Float (or float)	Single precision floating point
Double (or double)	Double precision floating point
Decimal (or decimal)	Fixed precision floating point (financial)
DateTime	An instant in time (to 100 ns)
String (or string)	Text (as Unicode characters)

When to use what

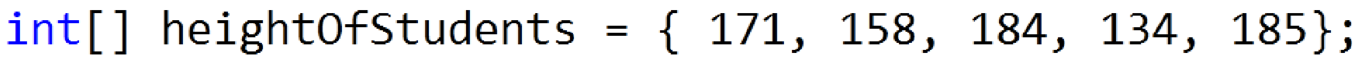
- **Integers** – when you want to work with *whole numbers.*

- **Float & Double** – when you want to perform scientific calculation (e.g. 5.81×10^8).

- **Decimal** – when working with financial calculations (i.e. money).

Arrays

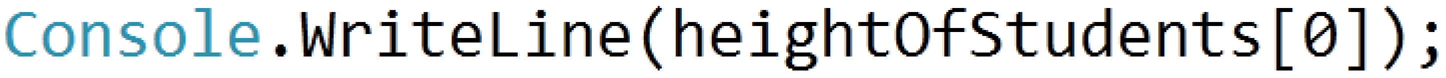
- Used to store numbers in a specific order.
- C# arrays are *zero indexed*.
- **[]** indicates that it is an array



```
int[] heightOfStudents = { 171, 158, 184, 134, 185};
```

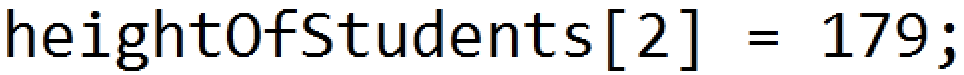
Arrays

- To retrieve an element from an array:



```
Console.WriteLine(heightOfStudents[0]);
```

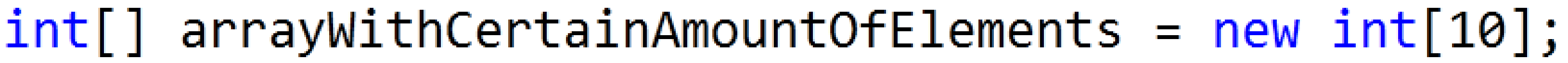
- To change the value of an element:



```
heightOfStudents[2] = 179;
```

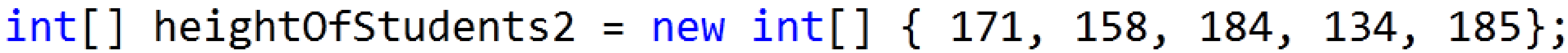
Arrays

- If you only need an array that contains a certain amount of elements, do following:

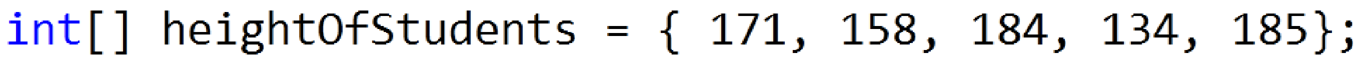


```
int[] arrayWithCertainAmountOfElements = new int[10];
```

- Another way of declaring heightOfStudents:



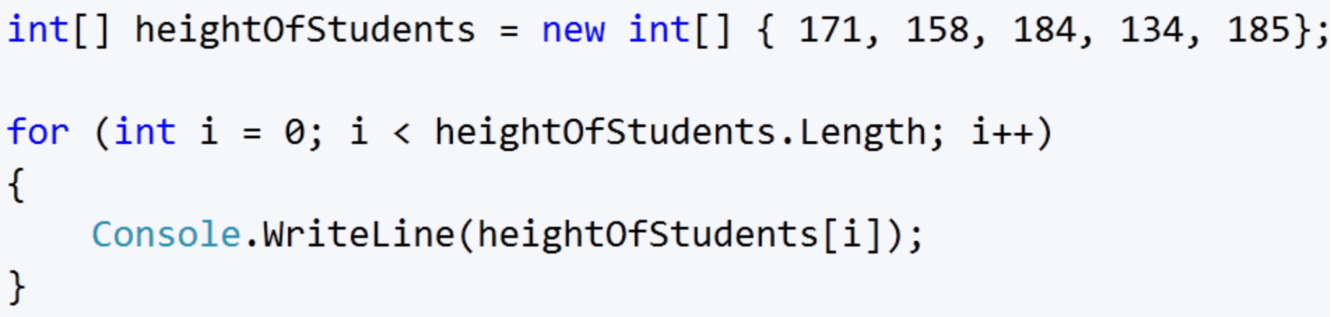
```
int[] heightOfStudents2 = new int[] { 171, 158, 184, 134, 185};
```



```
int[] heightOfStudents = { 171, 158, 184, 134, 185};
```

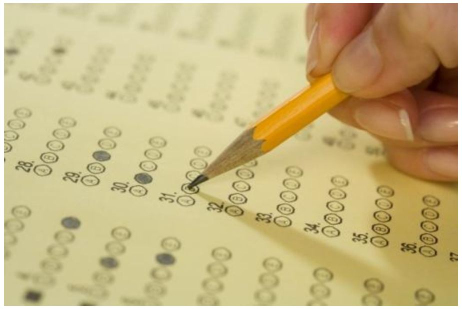
Demo

- In this demo we will display all the elements of an integer array on the screen.



```
int[] heightOfStudents = new int[] { 171, 158, 184, 134, 185};

for (int i = 0; i < heightOfStudents.Length; i++)
{
    Console.WriteLine(heightOfStudents[i]);
}
```



Conclusion

- heightOfStudents.Length is the amount of elements in an array – in our case it's **5**.
- However, we can only make 4 iteration, because arrays are *zero indexed*.

Summary

- We have:
  - – gone through how to create a variable of different *data types*.
  - – worked with arrays (lists)

Thank you for watching!

- Next lecture is at 13.10.2012.

- Any questions, please let me know:
  - vcs@clizware.net
  - vcs@artemlos.net