

# FOPCS Summary 05082021

## 05 Variables, Data Types and Operator

### ▼ Value type vs Reference type (akin pointer)

**Value type: Declared, memory location given, value assigned**

- Simple type: Signed integrals short int long, Unsigned, Unicode char, IEEE binary float point float double,
- Boolean
- others: Enum types, Struct types, Nullable value types

**Reference type: reference (memory address), and this reference Var stored at heap**

- Class type C {...}
- Interface type I {...}
- Array types []
- Delegate types D (...)

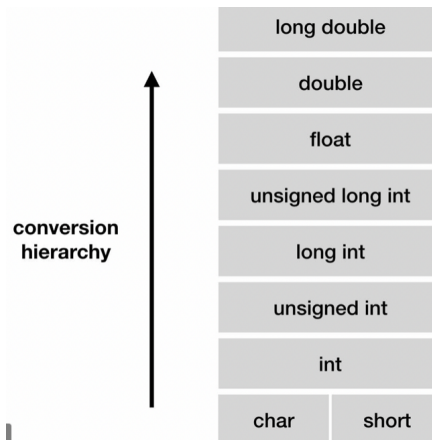
Variable: able to store changeable data ↔ Const

Type conversion (Implicit, explicit, incompatible conversion)

//implicit along conversion hierarchy, no data loss

//explicit against conversion hierarchy, truncated AKA casting

//parsing "forcing" incompatible conversion e.g Convert.ToInt32(input) or bool.Parse(input)



Variable Initialisation // Common error correction: initialise INDIVIDUALLY

Do these two codes mean the same thing?

```
double weight = 0, height = 0;
```

```
double weight, height = 0;
```

Expression and operators

Operators classification:

- Mathematical + - \* / % ++ — // + concatenation when either operands are string
- Comparison > < = !
- Logical &&And ||Or !Not leading to truth table

06 Conditionals

If else, statement blocks, nested ifs, switch case statement //for clarify

```
int Day = Convert.ToInt32(Console.ReadLine());
switch (Day)
{
    case 1:
        Console.WriteLine("Sun");
        break;
    case 2:
        Console.WriteLine("Mon"); break;
    case 3:
        Console.WriteLine("Tue"); break;
    case 4:
        Console.WriteLine("Wed"); break;
    case 5:
        Console.WriteLine("Thu"); break;
    case 6:
        Console.WriteLine("Fri"); break;
    case 7:
        Console.WriteLine("Sat"); break;
    default:
        Console.WriteLine("Out of Range"); break;
}
```