27

Nullable Types

Value Types

- Value types are structures and enumerations.
- Value Types are by default non-nullable types.
- Non-nullable types don’t support 'null' values to be assigned to its variables.

Reference Types

- Reference types are classes, interfaces.
- Reference Types are by default nullable types.
- Nullable types support 'null' values assigned to its variables.
- They don’t require the following syntax.

Converting Value-Types to Nullable-Value-Types

Nullable<int> x = null;

[or]

int? x = null;

Accessing value from Nullable Type

- variable.Value
- variable.HasValue

What is null?

- Represents 'blank' value.
  - Ex: In Employee class, the 'int NoOfChildren' can be 'null'.

Null Coalescing Operator

- The 'null coalescing operator' checks whether the value is null or not.
  - It returns the left-hand-side operand if the value is not null.
  - It returns the right-hand-side operand if the value is null.
- Advantage: Simplifying the syntax of 'if statement' to check if the value is null.

Null Coalescing Operator

variableName ?? valueIfNull

Null Propagation Operator

- The "Null Propagation Operator (?.) and (? []) checks the value of left-hand operand whether it is null or not.
  - It returns the right-hand-side operand (property or method), if the value is not null.
  - It returns null, if the value is null.
- It accesses the property or method, only if the reference variable is "not null"; just returns "null", if the reference variable is "null".

Null Propagation Operator (?.)

```
referenceVariable?.fieldName;

-- is same as --

(referenceVariable == null)? null : referenceVariable.fieldName;
```