CMPT 110

Simon Fraser University

MODULES, DATA TYPES, AND OPERATORS

Objectives for This PPT

- Provide a brief introduction to VB Modules.
- Introduce the **Theory of Data** in Computing Science.
- Define **data types, data structures**, and then contrast them.
- Discuss constants, variables, operators and expressions.
- Define arithmetic/logic operators in VB explicitly.
- Discuss **non-numerical data** (characters and Boolean variables) in VB.
- Dynamically versus Statically typed programming languages.
- **This PPT will NOT be on the midterm.**

Tentative Syllabus

Week	Topic	
1 (Sept. 4th)	Introduction to Course	
2 (11th)	Introduction to Programming	
3 (18th)	Programming in VB	
4 (29th)	Events	Unit 3 of the Study Guide
5 (Oct. 2nd)	Representing and Storing Values	
6 (19th)	**MIDTERM – October 9th**	
7 (16th)	Subprograms	
8 (23rd)	Decisions	
9 (30th)	Iteration	
10 (Nov. 6th)	Arrays	
11 (13th)	I/O	
12 (21st)	Graphics	
13 (27th)	Review	

- Visual Basic uses building blocks such as **Variables**, **Data Types**, **Procedures**, **Functions** and **Control Structures** in its programming environment.
- This section concentrates on the programming fundamentals of Visual Basic with the blocks specified.

Preamble

- Code in Visual Basic is stored in the form of modules.
- The three kind of modules are
  - **Form Modules**,
  - **Standard Modules,** and
  - **Class Modules**.
- A simple application may contain a single Form, and the code resides in that Form module itself. As the application grows, additional Forms are added and there may be a common code to be executed in several Forms.
- To avoid the duplication of code, a separate module containing a procedure is created that implements the common code. This is a standard Module.

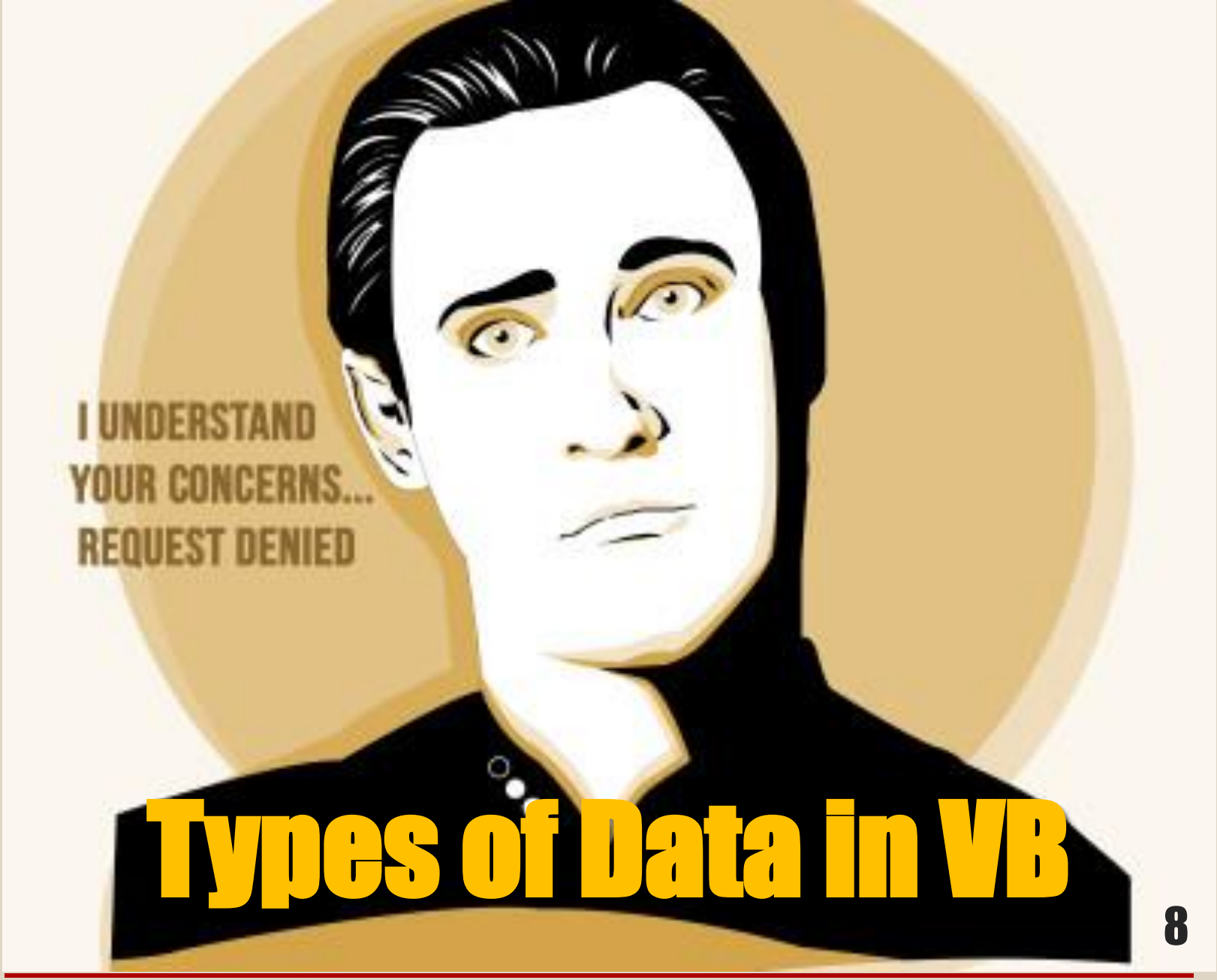
Modules

- Class module (.CLS filename extension) are the foundation of the object oriented programming in VB.
- New objects can be created by writing code in class modules.
- Each module can contain:
  - **Declarations :** May include constant, type, variable and DLL procedure declarations.
  - **Procedures :** A sub function, or property procedure that contain pieces of code that can be executed as a unit.

Modules

- These are the rules to follow when naming elements in VB - **variables, constants, controls, procedures**, and so on:
- A name must begin with a *letter*.
- It may be as much as 255 characters long.
- It must not contain a *space* or an *embedded period* or *type-declaration characters* used to specify a data type (these are ! # % $ & @).
- It must not be a reserved word (that is part of the code, like Option, for example)
- The dash (-), although legal, should be avoided because it may be *confused with the minus sign*.
- Instead of First-name use First_name or FirstName.

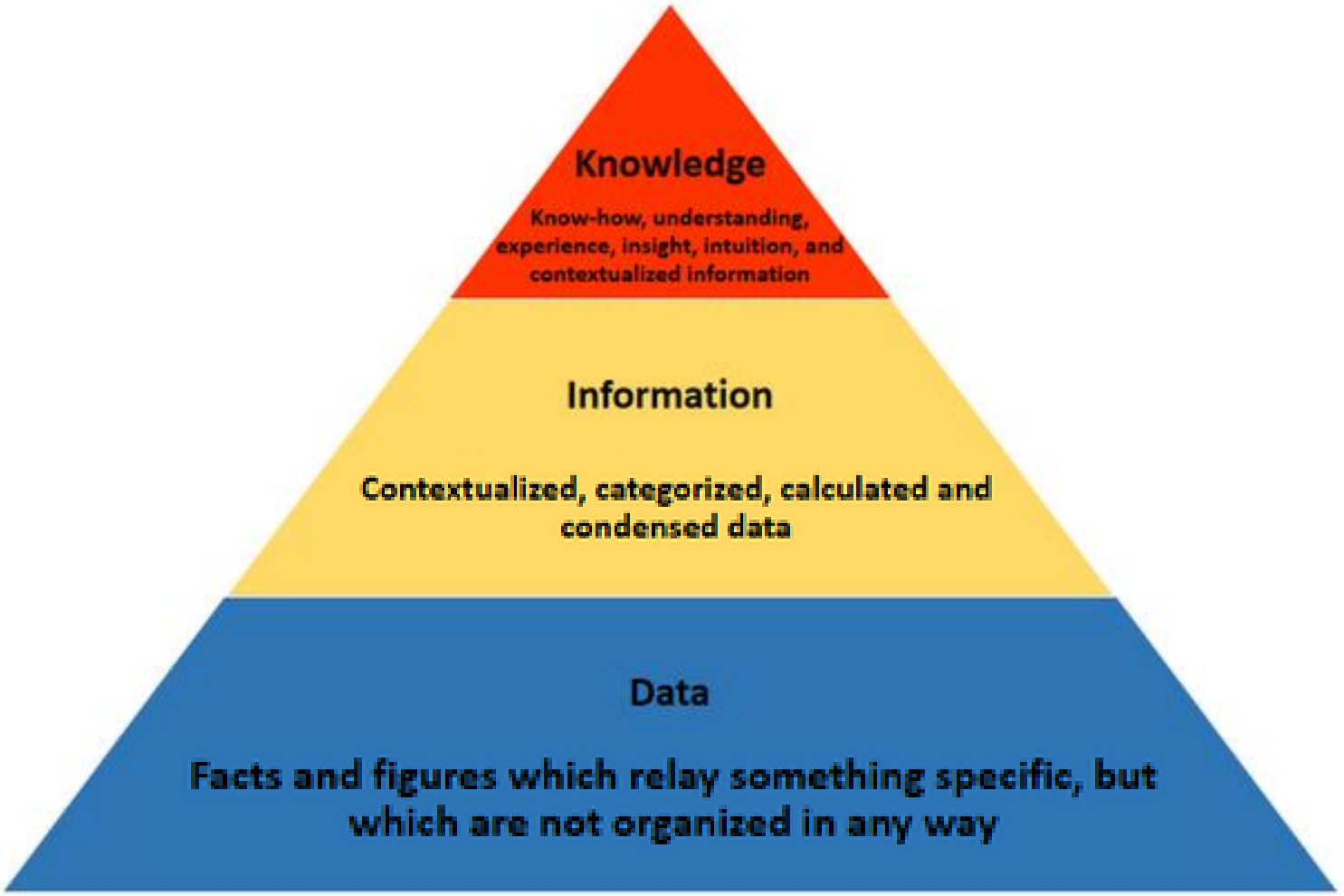
Modules



Types of Data in VB

First,

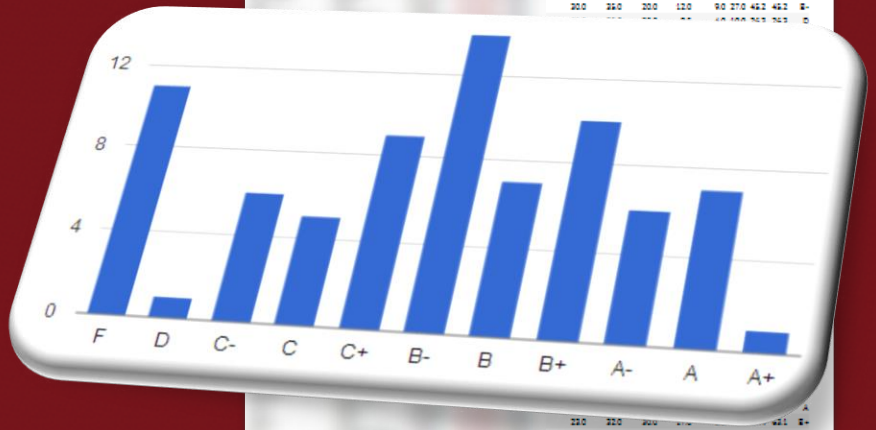
What is Data?



Taxonomy

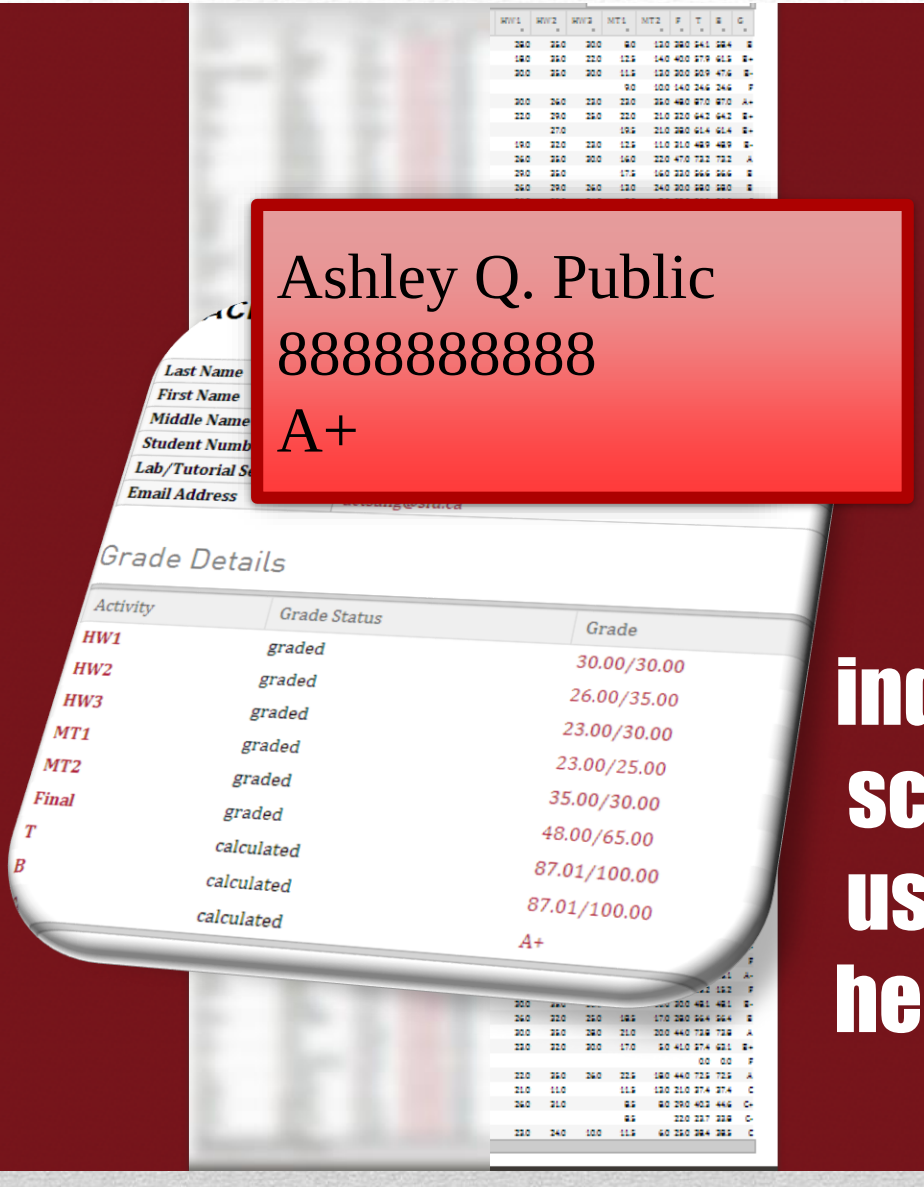
http://www.knowledge-management-tools.net/knowledge-information-data.html

Ashley Q. Public
8888888888
A+



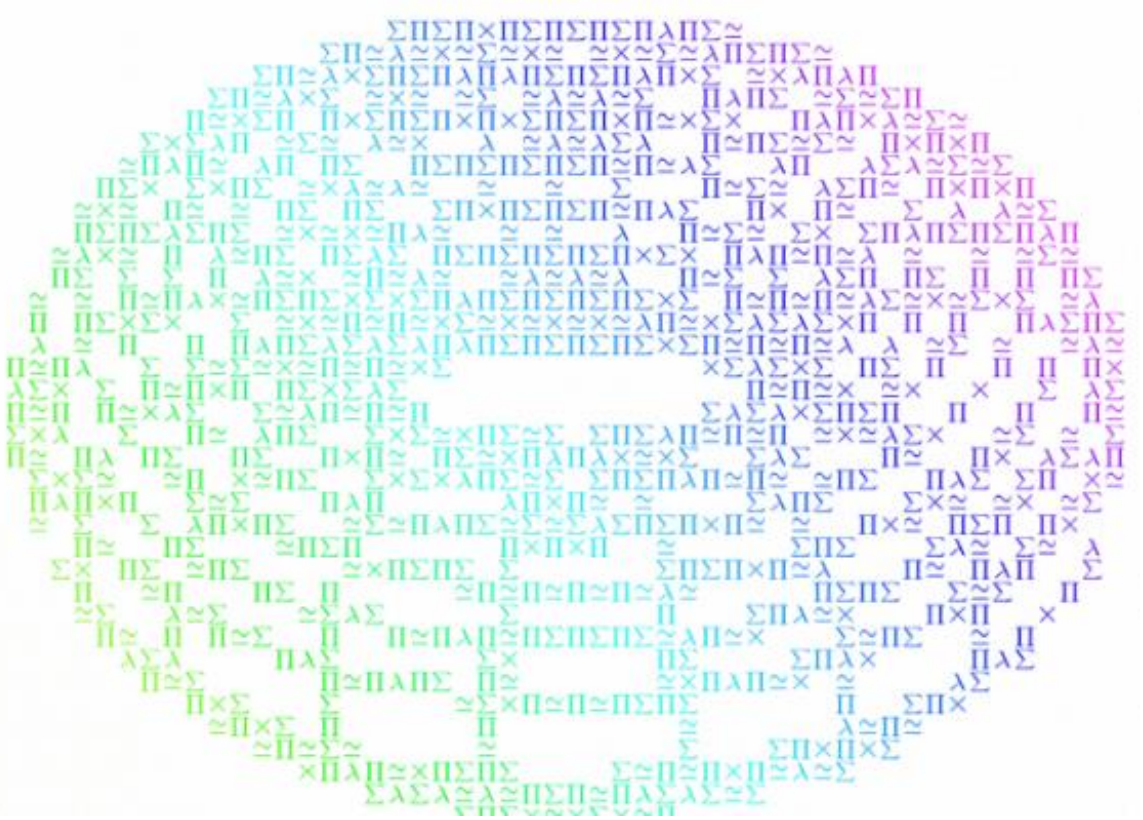
Example:

Ashley's grade is a datum used to calculate CMPT 225 statistics



...whereas her individual testing scores were data used to calculate her course grade.

Theory of Data



Type Theory and Type Systems

Type Theory:

- In mathematics, logic, and computer science, a **type theory** is any of a class of formal systems, some of which can serve as alternatives to set theory as a foundation for all mathematics. In type theory, every "term" has a "type" and operations are restricted to the terms of a certain type.
- Type theory is closely related to (and in some cases overlaps with) type systems, which are a programming language feature used to reduce bugs. The types of type theory were created to avoid paradoxes **(*i.e.*, Russell's Paradox)** in a variety of formal logics and *rewrite systems* and sometimes "type theory" is used to refer to this broader application.

Definition

Type System:

- A set of rules that assigns a property (called a *type*) to the elements of a computer program (*i.e.*, variables, expressions, functions, *etc.*) for the purpose of reducing coding errors.

Definition

Type System:

- A set of rules that assigns a property (called a *type*) to the elements of a computer program (*i.e.*, variables, expressions, functions, *etc.*) for the purpose of reducing coding errors.
- In OO, **typing** is the enforcement of a class/type of an object to ensure that the object is manipulated by valid expressions (expressions are formally defined later in this PPT).

Definition

Type System:

- A set of rules that assigns a property (called a *type*) to the elements of a computer program (*i.e.*, variables, expressions, functions, *etc.*) for the purpose of reducing coding errors.
- In OO, **typing** is the enforcement of a class/type of an object to ensure that the object is manipulated by valid expressions (expressions are formally defined later in this PPT).
- Depending on the extent of enforcement and the precise instant of enforcement, there are **strong/weak** and **static/dynamic** typings respectively – to be defined and discussed later in this PPT.

Definition


INTEGER
NUMERIC
VARCHAR
TIMESTAMP
ARRAY
TEXT
DATE
UUID
INTERVAL
CHAR
DECIMAL
TIME
SERIAL
BOOLEAN

Data Types

A data type, in programming, is a classification that specifies which type of value a variable has and what type of mathematical, relational or logical operations can be applied to it without causing an error. A string, for example, is a data type that is used to classify text and an integer is a data type used to classify whole numbers.

Definition: Data Type

A data type, in programming, is a classification that specifies which type of value a variable has and what type of mathematical, relational or logical operations can be applied to it without causing an error. A string, for example, is a data type that is used to classify text and an integer is a data type used to classify whole numbers.

Data Type	Used for	Example
String	Alphanumeric characters	hello world, Alice, Bob123
Integer	Whole numbers	7, 12, 999
Float (floating point)	Number with a decimal point	3.15, 9.06, 00.13
Character	Encoding text numerically	97 (in ASCII, 97 is a lower case 'a')
Boolean	Representing logical values	TRUE, FALSE

Definition: Data Type

Further Definitions

Data Types come in the following categories:

- **Primitive**, which can be either:
  - A *basic type* is a data type provided by a programming language as a basic building block. Most languages allow more complicated *composite types* to be recursively constructed starting from basic types.
  - A *built-in type* is a data type for which the programming language provides built-in support.
- **Non-Primitive** which can be either:
  - Composite structures of primitive types such as an array.
  - Data types that are not defined by the programming language, but are instead created by the programmer.

- The simple examples of data types given two slides back are, in fact, the common *primitive data types* utilized in computer programs include the following:
  1. Integers,
  2. Booleans,
  3. Characters,
  4. Floating Point Numbers,
  5. Alphanumeric Strings.

Data Types

Data Types

- The simple examples of data types given two slides back are, in fact, the common *primitive data types* utilized in computer programs include the following:
  1. Integers,
  2. Booleans,
  3. Characters,
  4. Floating Point Numbers,
  5. Alphanumeric Strings..
- One can go further and produce *complex/composite data types* (*i.e.*, an array), which are built upon simpler data types (*i.e.*, array elements), that are related to *data structures*.

- By default Visual Basic variables are of variant data types.
- The variant data type can store numeric, date/time or string data.
- When a variable is declared, a data type is supplied for it that determines the kind of data they can store.
- The fundamental data types in Visual Basic including *variant* are *integer*, *long*, *single*, *double*, *string*, *currency*, *byte* and *Boolean*.
- Visual Basic supports a vast array of data types.
- Each data type has limits to the kind of information and the minimum and maximum values it can hold.
- In addition, some types can interchange with some other types.

Data types in Visual Basic

1. Numeric

Byte	Store integer values in the range of 0 - 255
Integer	Store integer values in the range of (-32,768) - (+ 32,767)
Long	Store integer values in the range of (- 2,147,483,468) - (+ 2,147,483,468)
Single	Store floating point value in the range of (-3.4x10-38) - (+ 3.4x1038)
Double	Store large floating value which exceeding the single data type value
Currency	store monetary values. It supports 4 digits to the right of decimal point and 15 digits to the left

2. String

Use to store alphanumeric values. A variable length string can store approximately 4 billion characters

3. Date

Use to store date and time values. A variable declared as date type can store both date and time values and it can store date values 01/01/0100 up to 12/31/9999

4. Boolean

Boolean data types hold either a true or false value. These are not stored as numeric values and cannot be used as such. Values are internally stored as -1 (True) and 0 (False) and any non-zero value is considered as true.

5. Variant

Stores any type of data and is the default Visual Basic data type. In Visual Basic if we declare a variable without any data type by default the data type is assigned as default.

Note: Variant Data Type

- **Variant** is a data type in certain programming languages, particularly Visual Basic, OCaml, and C++when using the Component Object Model.
- In Visual Basic (and VBA) the Variant data type is a *tagged union* that can be used to represent any other data type (for example, integer, floating-point, single- and double-precision, object, etc.) except fixed-length string type and record types. **In Visual Basic, any variable not declared explicitly or the type of which is not declared explicitly, is taken to be a variant.**
- While the use of not explicitly declared variants is not recommended, they can be of use when the needed data type can only be known at runtime, when the data type is expected to vary, or when optional parameters and parameter arrays are desired. In fact, languages with a dynamic type system often have variant as the *only* available type for variables.

I'm trying to write a program in vb where a user is asked to type in a value in either the Fahrenheit Textbox or the Celsius Textbox. I want to use just **ONE** button to perform the calculations and two textboxes to display the outputs but I'm not sure I understand what's happening. The number I type into the Textbox isn't what's been calculated.

Here's the code:

```
Private Sub convertButton_Click(sender As Object, e As EventArgs) Handles convertButton.Click

    Dim FahrenheitValue, CelsiusValue As Double

    FahrenheitValue = Val(fahrenheitBox.Text)
    CelsiusValue = Val(celsiusBox.Text)

        FahrenheitValue = (9 / 5) * (CelsiusValue + 32)
        CelsiusValue = (5 / 9) * (FahrenheitValue - 32)

        celsiusBox.Text = CelsiusValue
        fahrenheitBox.Text = FahrenheitValue


End Sub
```

?

Single Versus Double Precision

Data Type

Visual Basic also classifies information into two major data types:

- Numeric data types and
- Non-numeric data types.

Table 5.1: Numeric Data Types

Type	Storage	Range of Values
Byte	1 byte	0 to 255
Integer	2 bytes	-32,768 to 32,767
Long	4 bytes	-2,147,483,648 to 2,147,483,648
Single	4 bytes	-3.402823E+38 to -1.401298E-45 for negative values 1.401298E-45 to 3.402823E+38 for positive values.
Double	8 bytes	-1.79769313486232e+308 to -4.94065645841247E-324 for negative values 4.94065645841247E-324 to 1.79769313486232e+308 for positive values.
Currency	8 bytes	-922,337,203,685,477.5808 to 922,337,203,685,477.5807
Decimal	12 bytes	+/- 79,228,162,514,264,337,593,543,950,335 if no decimal is use +/- 7.9228162514264337593543950335 (28 decimal places).

VB Numerical Data Types

Table 5.2: Nonnumeric Data Types

Data Type	Storage	Range
String(fixed length)	Length of string	1 to 65,400 characters
String(variable length)	Length + 10 bytes	0 to 2 billion characters
Date	8 bytes	January 1, 100 to December 31, 9999
Boolean	2 bytes	True or False
Object	4 bytes	Any embedded object
Variant(numeric)	16 bytes	Any value as large as Double
Variant(text)	Length+22 bytes	Same as variable-length string

VB Non-Numerical Data Types

In programming, a value written exactly as it's meant to be interpreted. In contrast, a *variable* is a name that can represent different values during the execution of the program. And a *constant* is a name that represents the same value throughout a program. But a literal is not a name -- it is the value itself.

A literal can be a number, a character, or a string. For example, in the expression,

x = 3

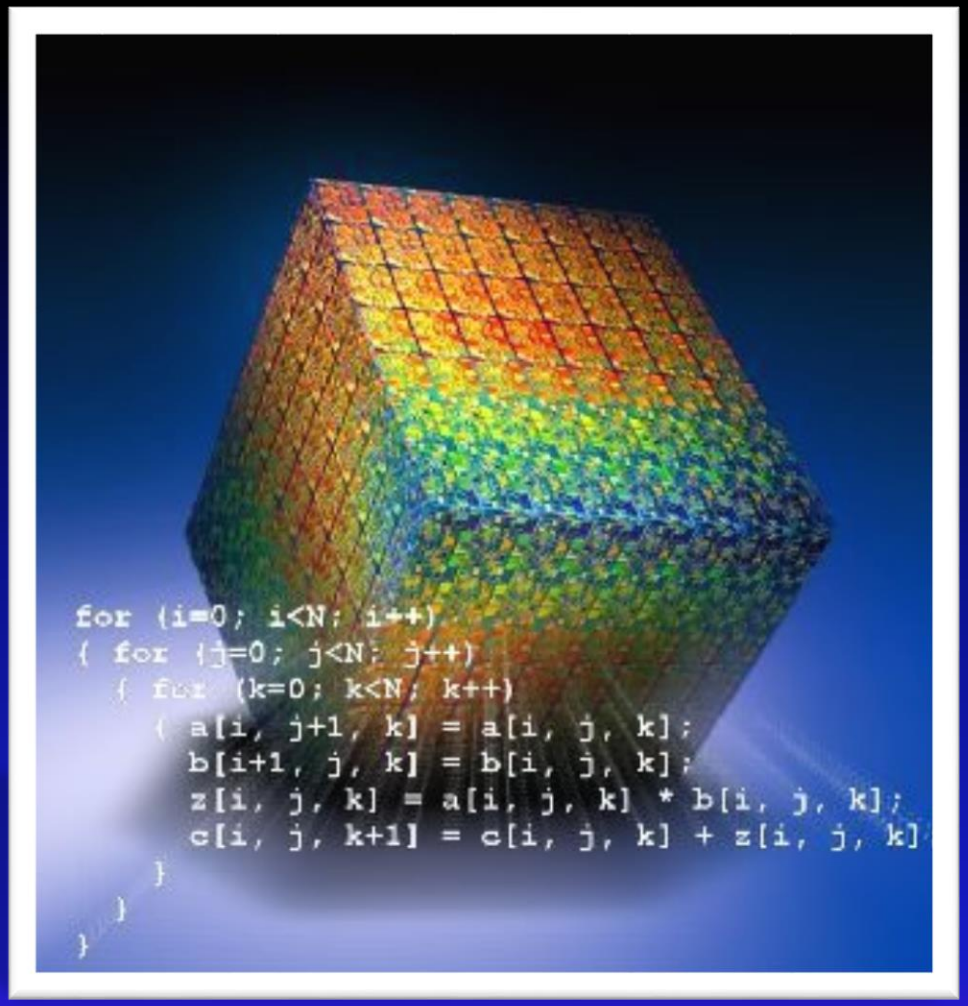
x is a variable, and **3** is a literal.

Literals

Literals are values that you assign to data. In some cases, we need to add a suffix behind a literal so that VB can handle the calculation more accurately. For example, we can use num=1.3089# for a Double type data. Some of the suffixes are displayed in the table below:

Suffix	Data Type
&	Long
!	Single
#	Double
@	Currency

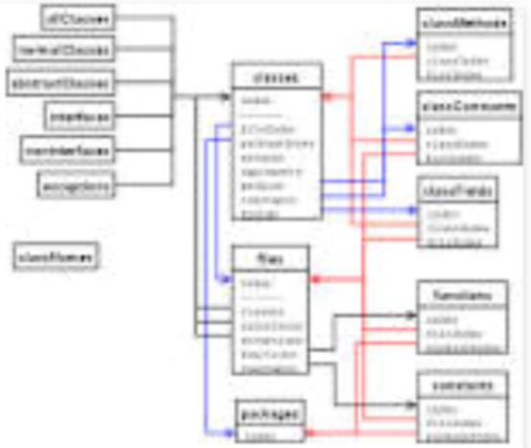
Suffixes for Literals



Aside on Data Structures

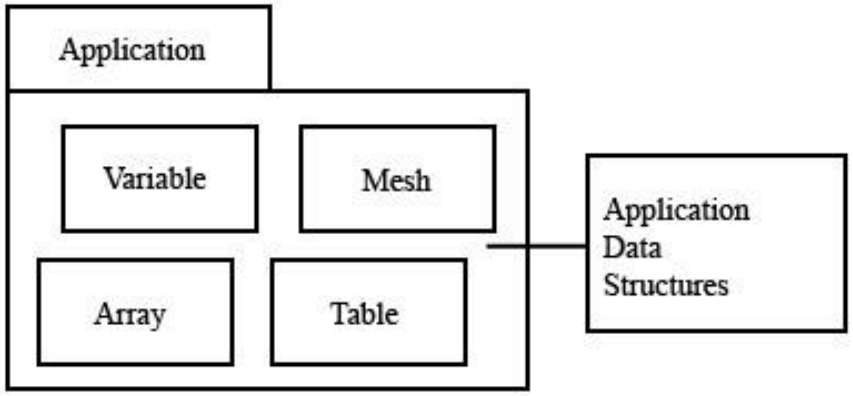
Data structure

In computer science, a data structure is a particular way of storing and organizing data in a computer so that it can be used efficiently.
Wikipedia

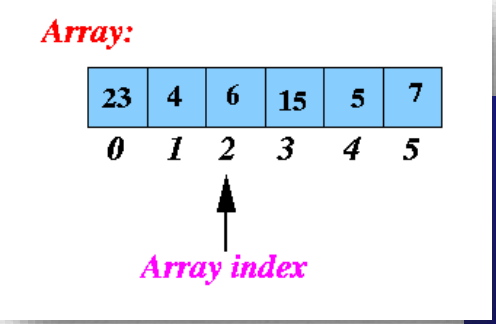


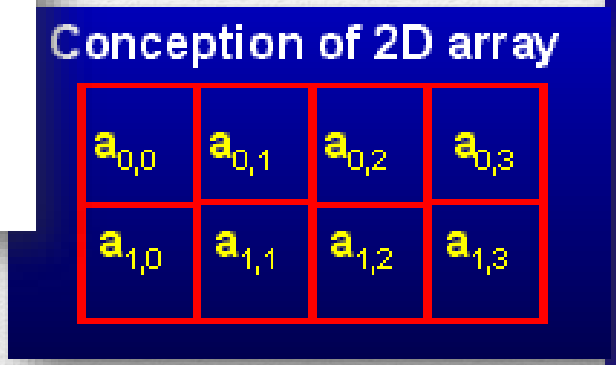
Recall this Definition

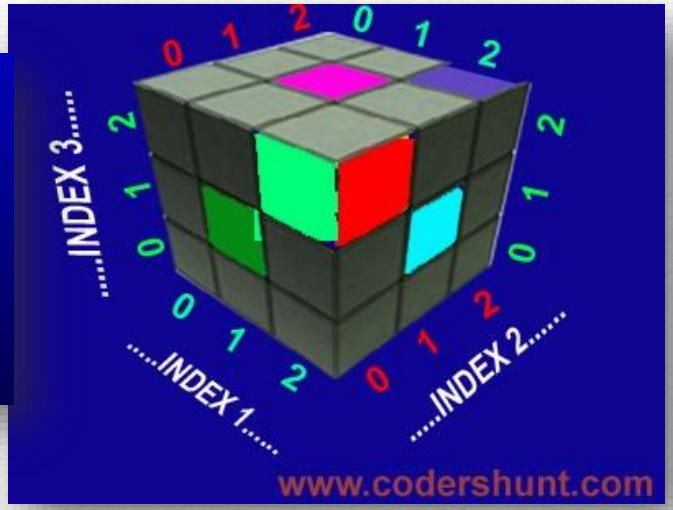
What is the difference between a *complex data type* and a *data structure*?



Question







- An *array* is an arrangements of objects that follow a specific pattern (usually in rows, columns, *etc*.).

Array Example

$$\boldsymbol{A} = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & & & A_{2n} \\ \vdots & & & \vdots \\ A_{n1} & A_{n2} & \cdots & A_{nn} \end{bmatrix}$$

- An *array* is an arrangements of objects that follow a specific pattern (usually in rows, columns, *etc*.).
- For example, this *complex data type* above is called a matrix and forms the foundation of *matrix algebra* that can be part of a mathematical algorithm.

Array Example

FACT:

An array can be looked upon as both a *complex data type* and a *data structure*.

Array Example

Array data structure - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/**Array_data_structure**
The simplest **type** of **data structure** is a linear **array**. This is also called one-dimensional **array**. In computer science, an **array data structure** or simply an **array** is a ...

Array data type - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/**Array_data_type**
In computer science, an **array** type is a **data type** that is meant to describe a ... **Array** types are often implemented by **array data structures**, but sometimes by ...

Array Example

- A *data type* is something "**visible**" to the programmer (*i.e.*, a matrix in a mathematical code).

Array Example

- A *data type* is something "**visible**" to the programmer (*i.e.*, a matrix in a mathematical code).
- A *data structure* is something "**invisible**" to the programmer, implemented behind the scenes (*i.e.*, the way the computer stores and manipulates, say, matrix data during program execution for the sake of efficiency).

Array Example

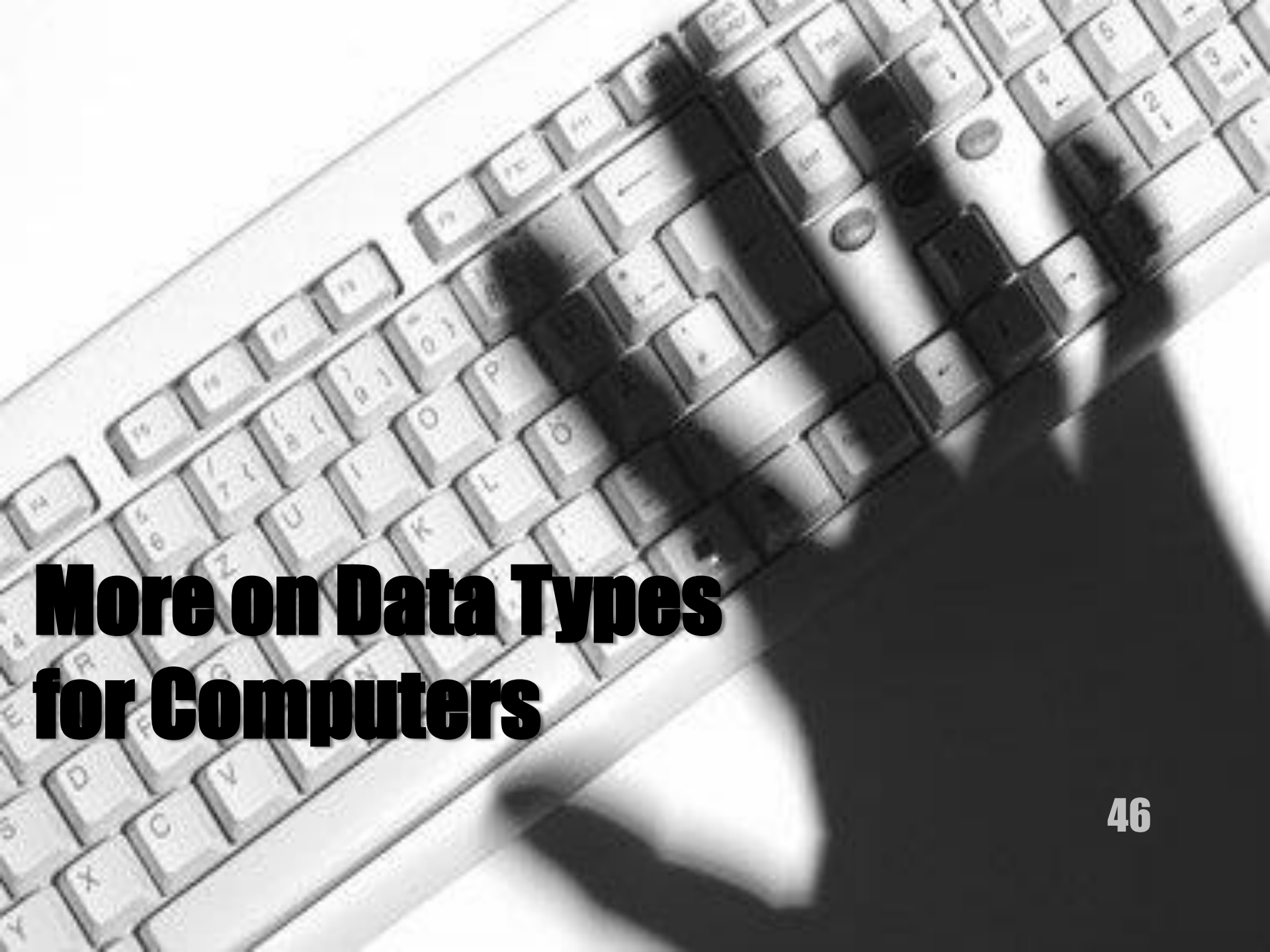
- A *data type* is something "**visible**" to the programmer (*i.e.*, a matrix in a mathematical code).
- A *data structure* is something "**invisible**" to the programmer, implemented behind the scenes (*i.e.*, the way the computer stores and manipulates, say, matrix data during program execution for the sake of efficiency).
- Thus, an array data type such as a matrix in a computer program may be efficiently implemented by an array data structure by a computer.

Array Example

- A *data type* is something "**visible**" to the programmer (*i.e.*, a matrix in a mathematical code).
- A *data structure* is something "**invisible**" to the programmer, implemented behind the scenes (*i.e.*, the way the computer stores and manipulates, say, matrix data during program execution for the sake of efficiency).
- Thus, an array data type such as a matrix in a computer program may be efficiently implemented by an array data structure by a computer.
- We will discuss this in more depth later in the course.

Array Example

End of Theory of Data



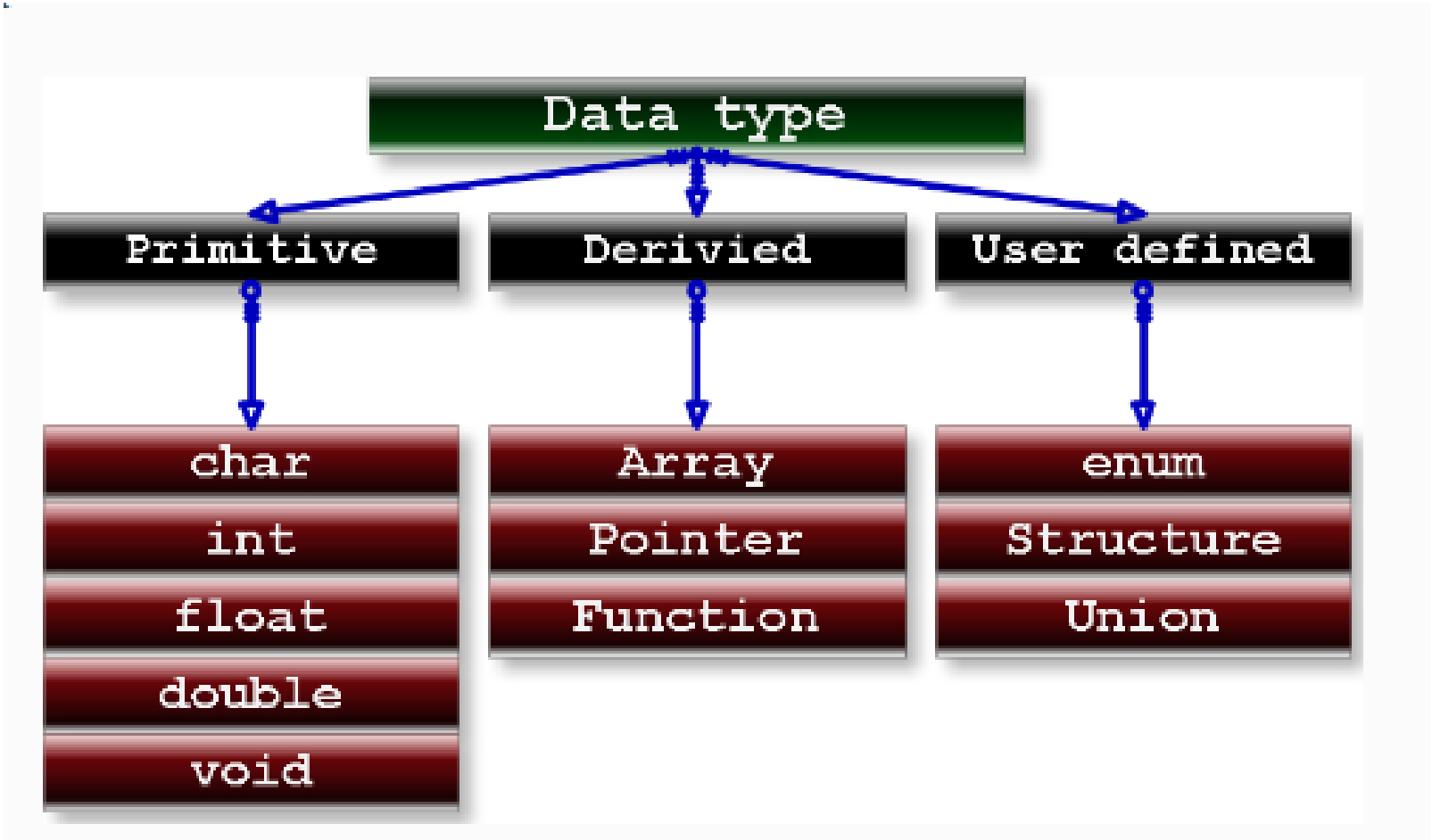
More on Data Types for Computers

First…

One difficulty for new programmers is figuring out how to represent data and information in order to describe the "real world" in a computer since the choices are so limited – the problem of determining a good abstraction.

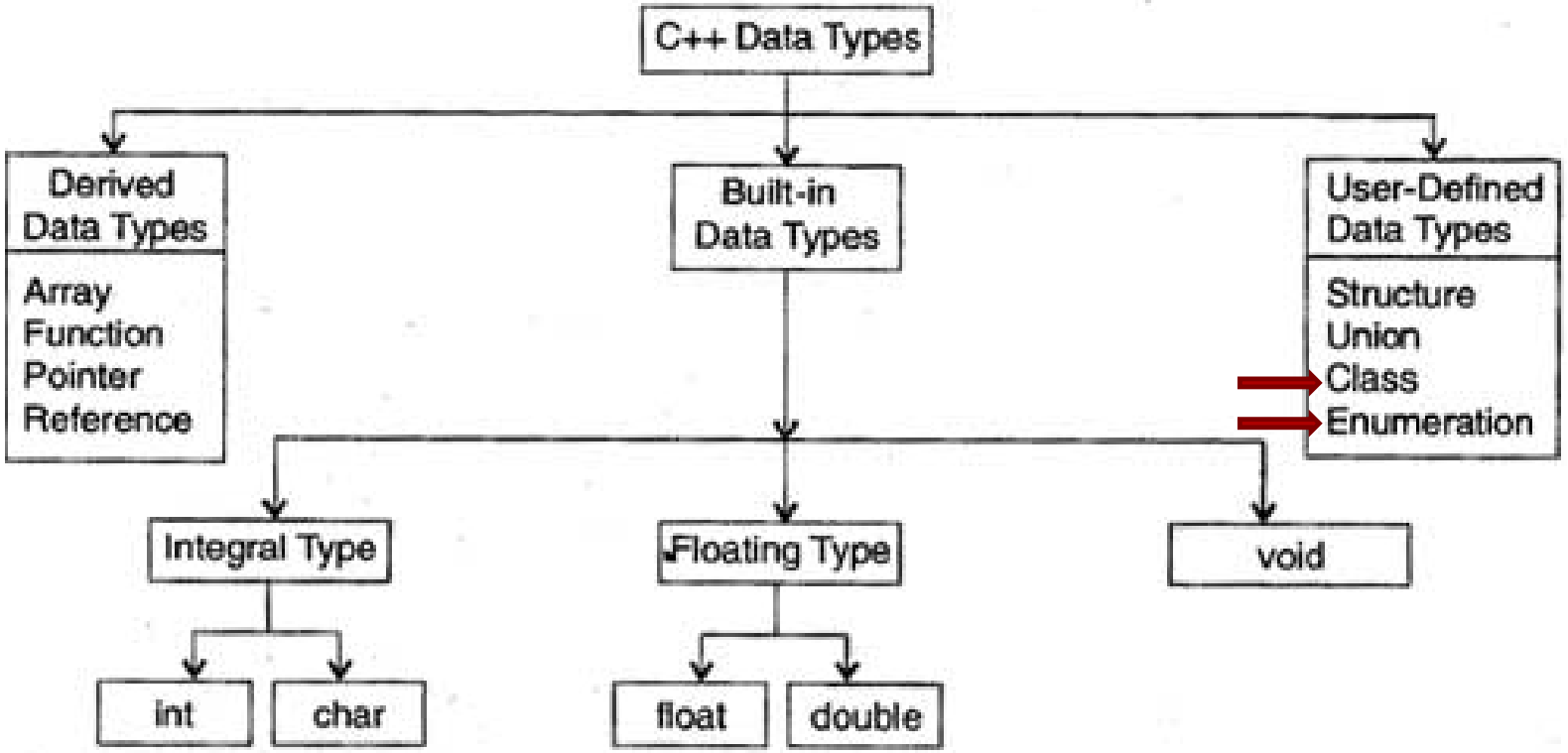
Second…

This leads us to what we have been introducing, which is the *theory of data classes*.



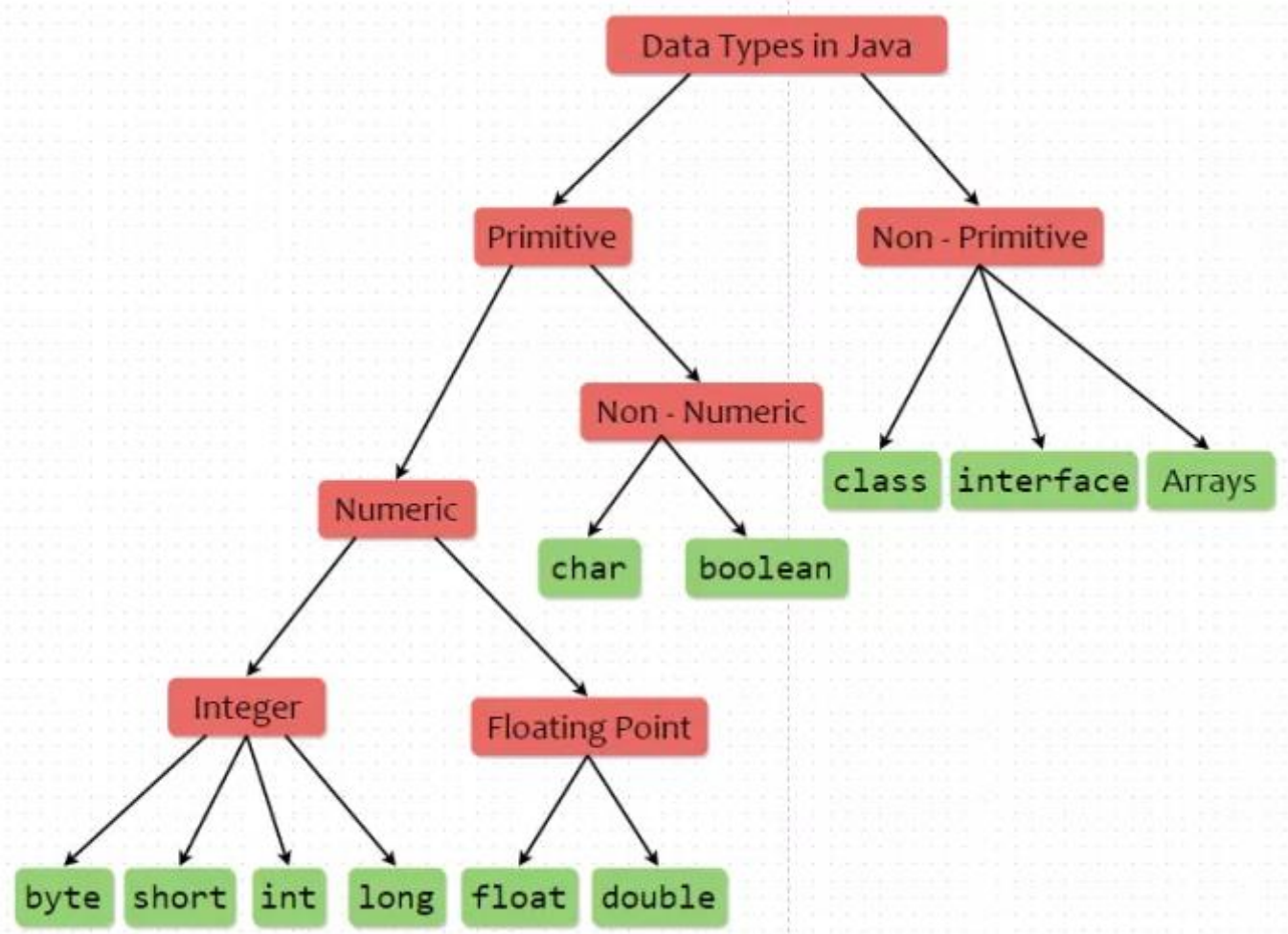
Example: C Taxonomy

http://60minutestutorials.com/ctutorial/data-types-of-c/



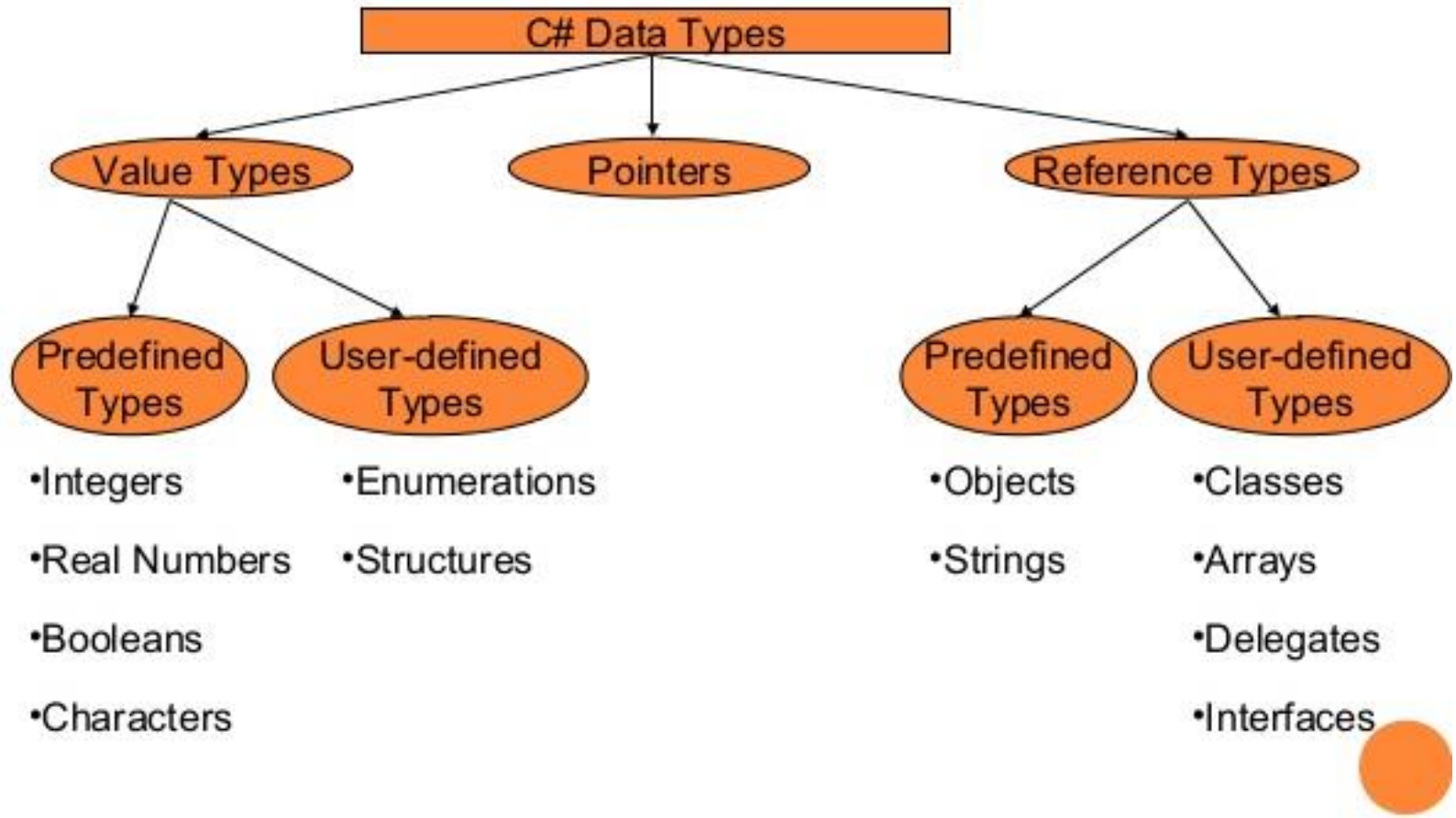
Example: C++ Taxonomy (OO)

http://ecomputernotes.com/cpp/introduction-to-oop/data-types-in-cpp



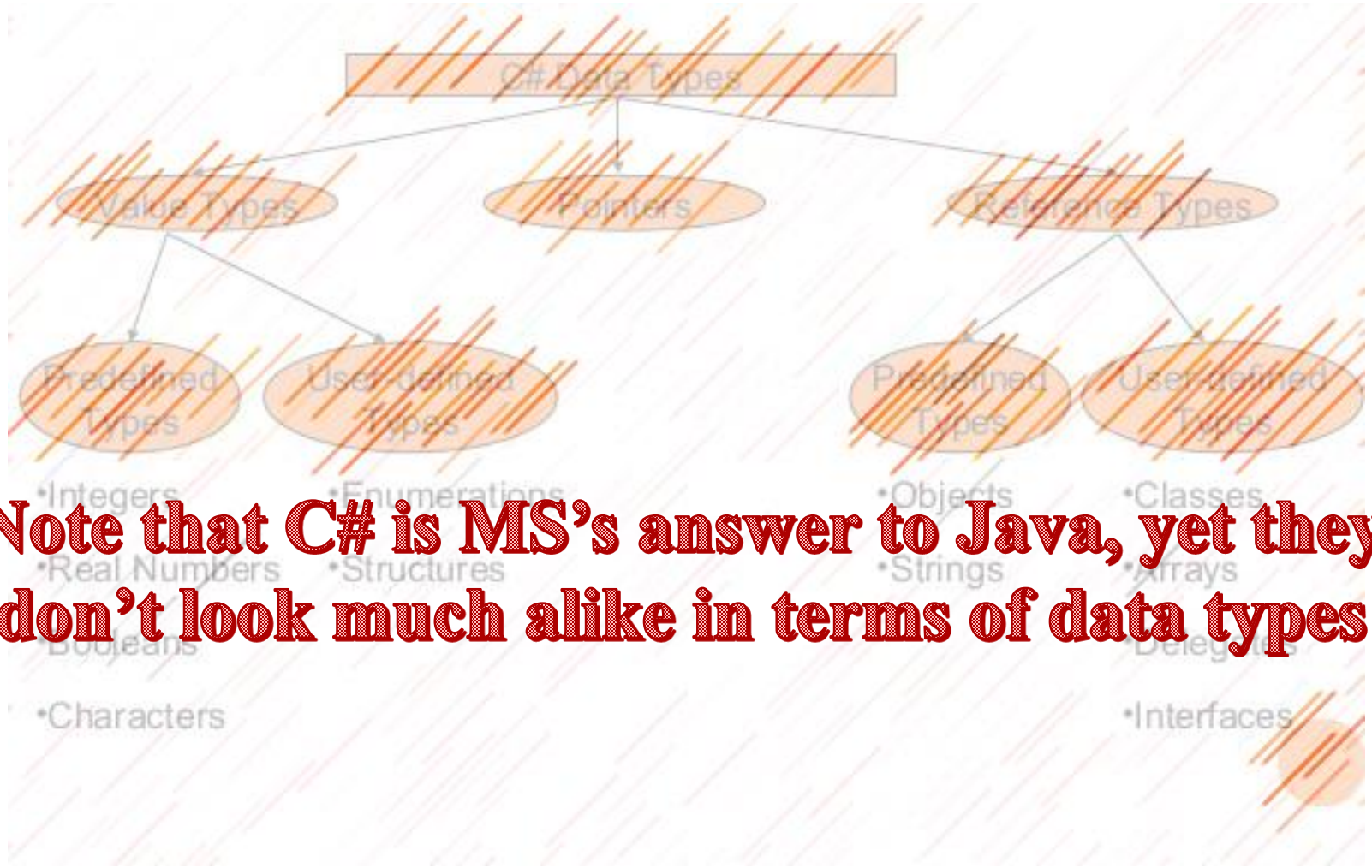
Example: Java Taxonomy

http://theoryofprogramming.com/2015/05/27/java-tutorials-data-types-input-operators-in-java/



Example: C# Taxonomy

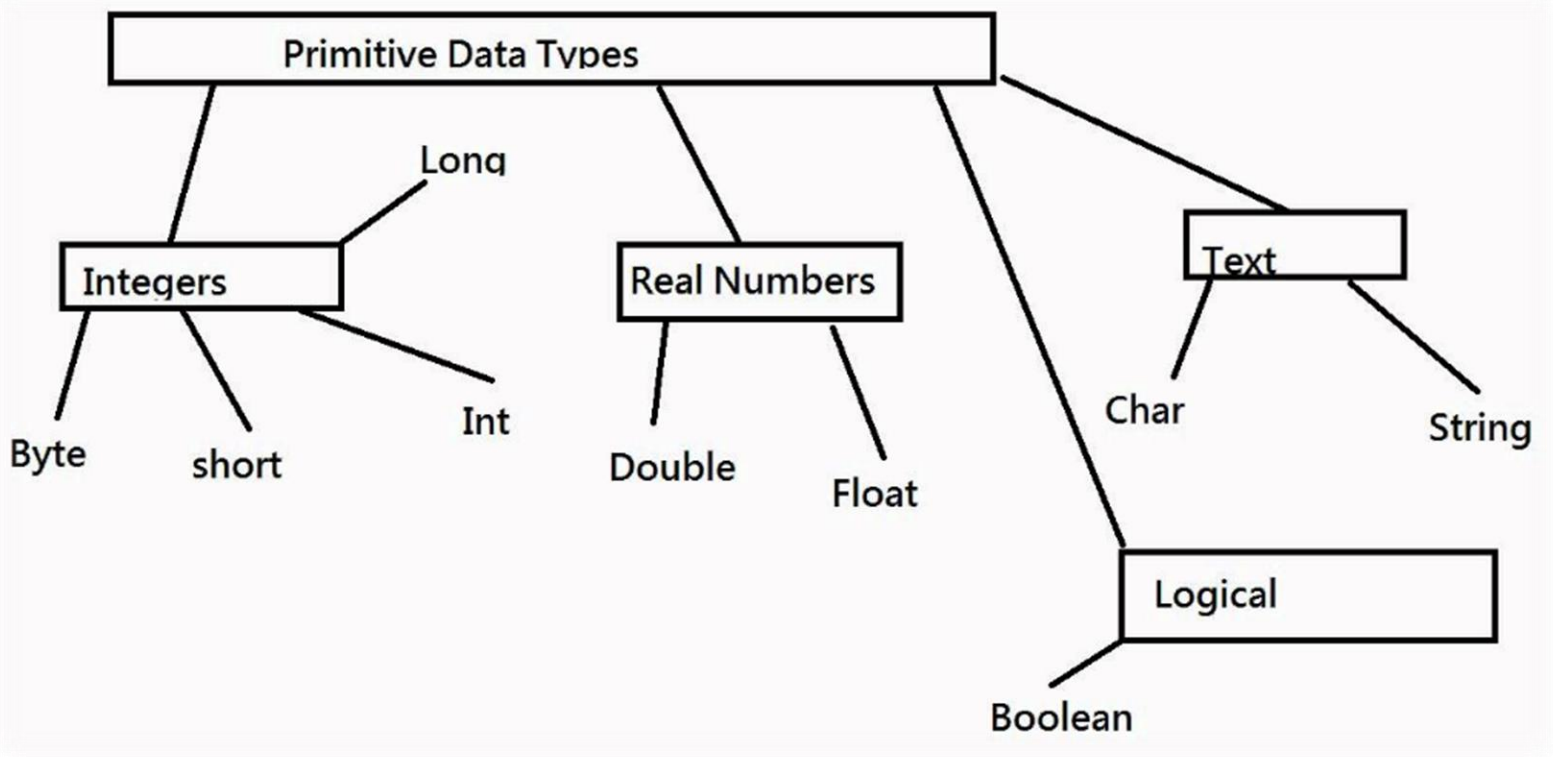
https://www.slideshare.net/shehrevard/programming-in-c-20414692



Note that C# is MS's answer to Java, yet they don't look much alike in terms of data types

Example: C# Taxonomy

https://www.slideshare.net/shehrevard/programming-in-c-20414692



Example: VB Taxonomy

VIDEO: https://www.youtube.com/watch?v=I1QHyPMfVdM

- Array
- Double (double-precision floating-point number)
- Integer
- Long (double-precision integer)
- Object
- Record
- Single (single-precision floating-point number)
- String
- Variant

VB Data Types Revisited

VIDEO: https://www.youtube.com/watch?v=I1QHyPMfVdM

Another Classification of Data

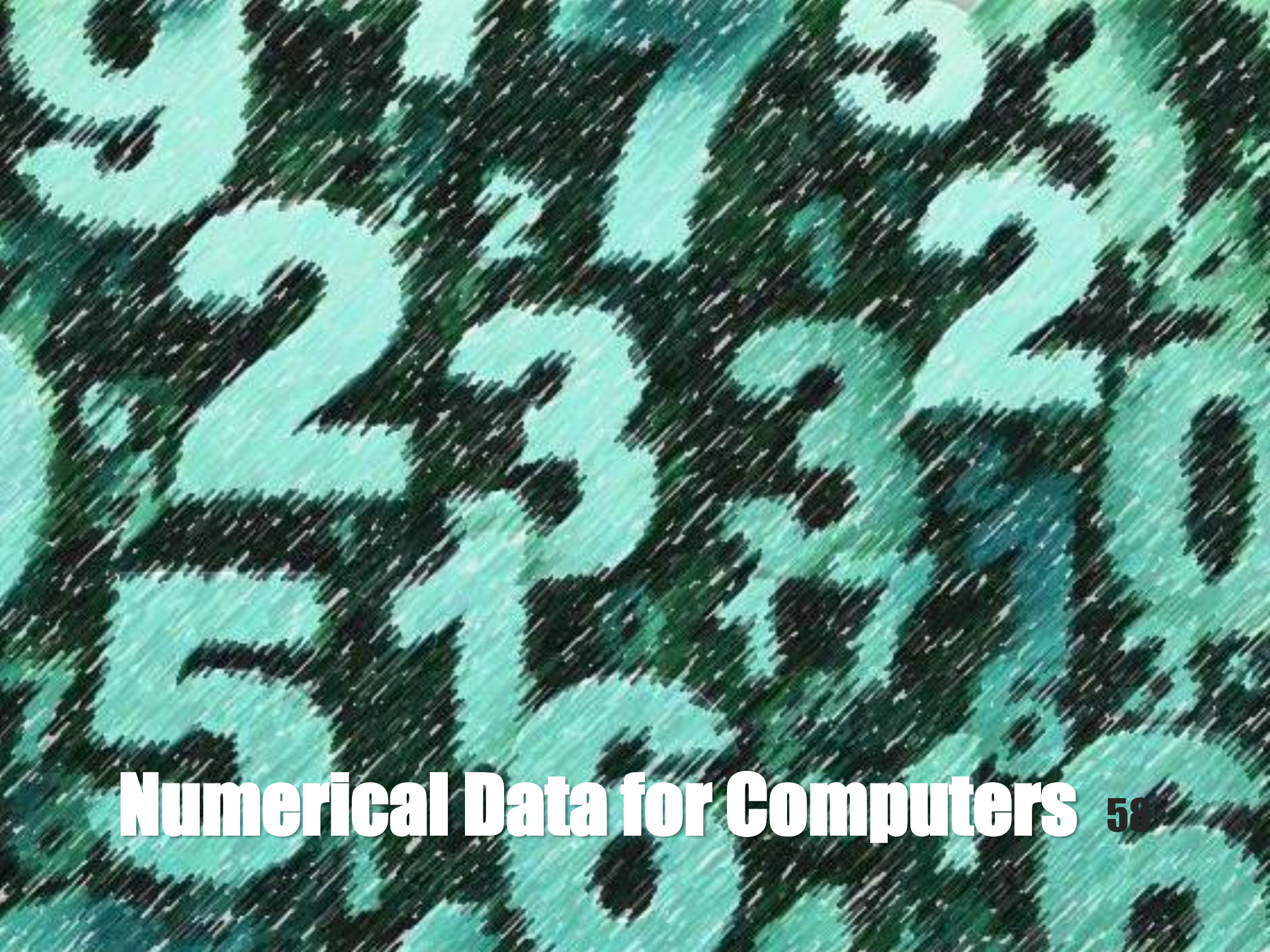
One can broadly classify data as either numeric or non-numeric. Examples of non-numerical data are:

- Characters,
- Strings (concatenation of characters),
- Booleans (True, False),
- *etc.*

Another Classification of Data

One can broadly classify data as either numeric or non-numeric. Examples of non-numerical data are:

- Characters,
- Strings (concatenation of characters),
- Booleans (True, False),
- *etc.*


Numerical Data for Computers
58

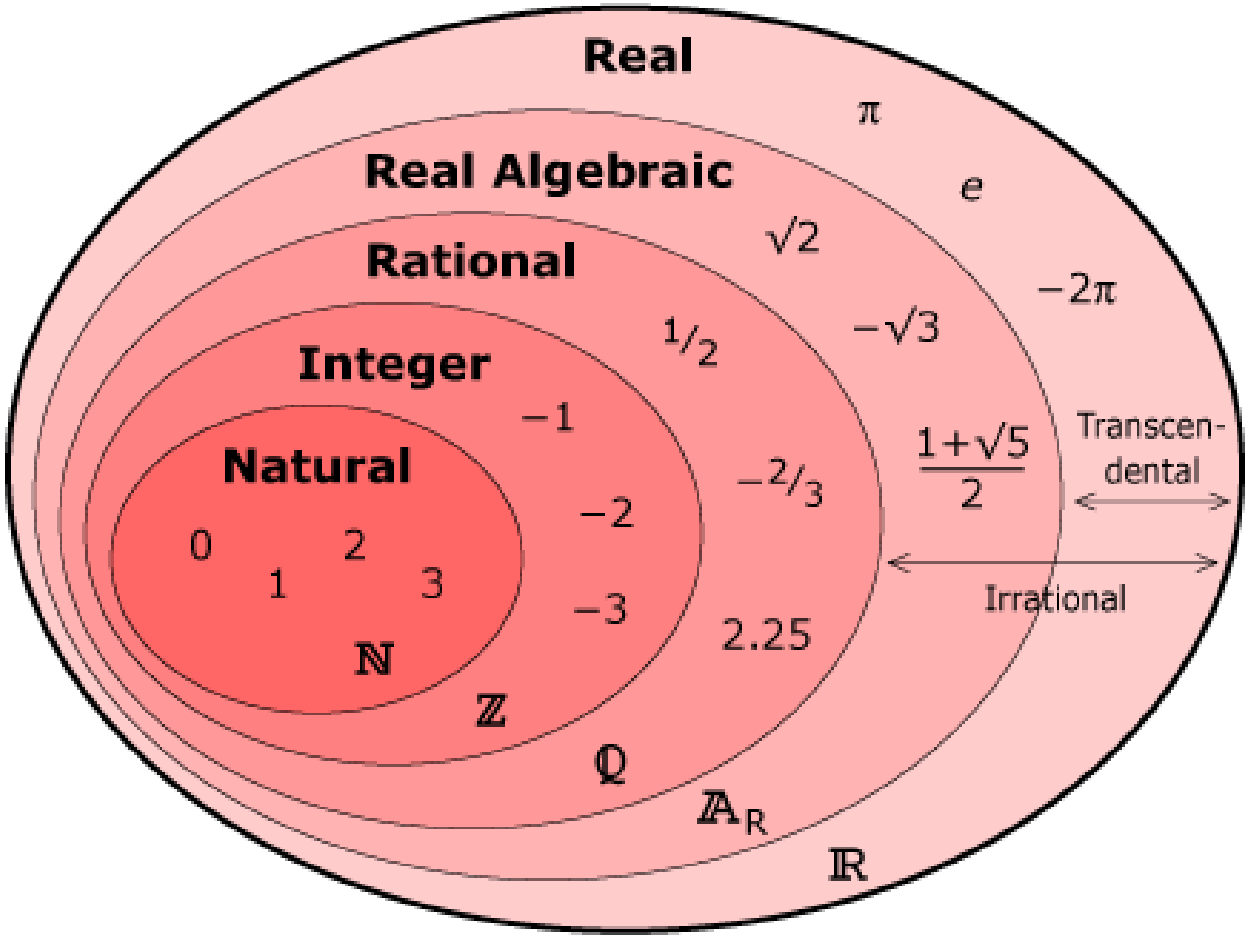
- Recall that the Central Processing Unit (CPU) contains the Arithmetic/Logic Unit (ALU – the circuit that performs the basic arithmetic and logic operations) and the CU (traffic cop).

- Recall that the Central Processing Unit (CPU) contains the Arithmetic/Logic Unit (ALU – the circuit that performs the basic arithmetic and logic operations) and the CU (traffic cop).
- A computer program mostly processes numerical and logical data into useful information of one sort or another, in part, through these circuits.

Numerical Data for Computers

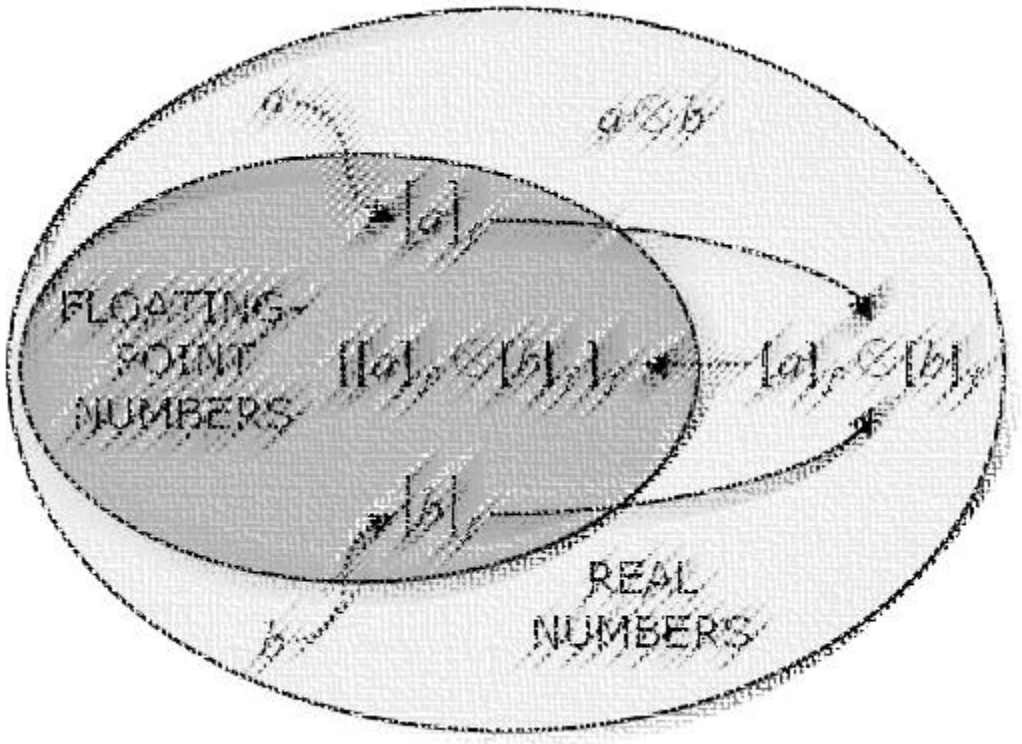
- Recall that the Central Processing Unit (CPU) contains the Arithmetic/Logic Unit (ALU – the circuit that performs the basic arithmetic and logic operations) and the CU (traffic cop).
- A computer program mostly processes numerical and logical data into useful information of one sort or another, in part, through these circuits.
- Von Neumann computers are binary and, therefore, *integer in nature*.

Numerical Data for Computers



Common Number Sets

http://thinkzone.wlonk.com/Numbers/NumberSets.htm



Floating Point Numbers

- Clearly, real numbers are more challenging to represent in a computer than integers.

Floating Point Numbers

Floating Point Numbers

- Clearly, real numbers are more challenging to represent in a computer than integers.
- We represent real numbers on computers as **floating point decimal numbers**:
  - The term, floating point, means there are no fixed number of digits before and after the decimal point (*e.g.*, the decimal point can *float*).

- Clearly, real numbers are more challenging to represent in a computer than integers.
- We represent real numbers on computers as **floating point decimal numbers**:
  - The term, floating point, means there are no fixed number of digits before and after the decimal point (*e.g.*, the decimal point can *float*).
- Note that real numbers require more computing power to process them – some computers have a *floating point unit* chip (FPU) dedicated to this task.

Floating Point Numbers

Floating Point Numbers

- Clearly, real numbers are more challenging to represent in a computer than integers.
- We represent real numbers on computers as **floating point decimal numbers:**
  - The term, floating point, means there are no fixed number of digits before and after the decimal point (*e.g.*, the decimal point can *float*).
- Note that real numbers require more computing power to process them – some computers have a *floating point unit* chip (FPU) dedicated to this task.
- **Most floating point numbers are only *approximations* when represented in a computer (good to 1:10^{15}).**

Statement: The number, one, is represented by the symbol, 1.

Question

Statement: The number, one, is represented by the symbol, 1.

Question: What is the data type of the second-to-last character in this statement?

Question

- Numbers can be numeric as in an arithmetic expression, or characters as in the previous statement.

Answer

- Numbers can be numeric as in an arithmetic expression, or characters as in the previous statement.
- There are *functions* for conversion from one data type to another.

Answer

- Numbers can be numeric as in an arithmetic expression, or characters as in the previous statement.
- There are *functions* for conversion from one data type to another.
- Clearly, operators include *functions* (later topic) given below in VB:

FUNCTION	CONVERTS ITS ARGUMENT To
CBool	Boolean
CByte	Byte
CChar	Unicode character
CDate	Date
CDb1	Double
CDec	Decimal
Clnt	integer (4-byte integer, 1nt32)
CLng	1ong (8-byte integer, 1nt64)
CObj	Object
CShort	Short (2-byte integer, Inti6)
CSng	Single
CStr	String

Answer

Non-Numerical Data Types

Boolean Data Types

True and **False** are VB's Boolean variables.

Boolean Data Types


String Data Types

String (computer science) - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/String_(computer_science)
In **computer** programming, a **string** is traditionally a sequence of characters, either as a ... For any two **strings** s and t in Σ*, their concatenation is **defined** as the ...
Formal theory - String datatypes - Text file strings - Non-text strings

Example: A byte is an 8-bit binary string (*i.e.*, 10110110)

Definition

- A *string literal* is the notation for representing a string value within the text of a computer program.
- In Visual Basic **string literals are enclosed by double quotes**.
- A string in Visual Basic is a sequence of *Unicode* characters.

Strings in VB

```
Option Strict On


Module Example

    Sub Main()

        Dim str1 As String = "There are 10"
        Dim str2 As String = " apples"

        Console.WriteLine(str1 + str2)

        Console.WriteLine("The length of the first string is " _
            + str1.Length.ToString() + " characters")

    End Sub

End Module
```

Example

Declare in Memory

I see "**Declare in Memory**" as a better meaning as it describes what it does in current versions of the language, but it is not the proper meaning. In fact, at the origins of Basic Dim was only used to declare arrays. For regular variables no keyword was used, instead their type was inferred from their name. Jun 23, 2009

vb6 - What does DIM stand for in Visual Basic and BASIC? - Stack ...
https://stackoverflow.com/questions/.../what-does-dim-stand-for-in-visual-basic-and-basi...

```
Dim str1 As String = "There are 10"
Dim str2 As String = " apples"
```

Prabir Das replied Jul 15, 2013. **Dim** stands for dimension, used to declare variables in **visual basic** with proper datatype **definition**. Jul 15, 2013

Visual Basic - What Does DIM Mean? - Toolbox for IT Groups
visualbasic.ittoolbox.com/.../visualbasic-l/visual-basic-what-does-dim-mean-5256053

Example

In the preceding example, we create two string variables. Then we add them and compute the length of the first string.

```
Dim str1 As String = "There are 10"
```

A string variable is declared and initiated.

```
Console.WriteLine(str1 + str2)
```

Two strings are concatenated. We use the + operator to add two strings.

```
Console.WriteLine("The length of the first string is " _
    + str1.Length.ToString() + " characters")
```

The `Length` property is used to determine the length of the string.

```
$ ./basics.exe
There are 10 apples
The length of the first string is 12 characters
```

Running the example gives this result.

Example

http://zetcode.com/lang/visualbasic/strings/

Methods that Return Strings

The following table lists several useful methods that return new string objects.

Method name	Use
String.Format	Builds a formatted string from a set of input objects.
String.Concat	Builds strings from two or more strings.
String.Join	Builds a new string by combining an array of strings.
String.Insert	Builds a new string by inserting a string into the specified index of an existing string.
String.CopyTo	Copies specified characters in a string into a specified position in an array of characters.

VB String Creation

- Composite types are derived from more than one primitive type. This can be done in a number of ways. The ways they are combined are called data structures. Composing a primitive type into a compound type generally results in a new type, e.g. *array-of-integer* is a different type to *integer*.
- Common examples include *Arrays*, *Records*, *Unions*, *Sets and Objects*.

Composite Data Types

- An **array** stores a number of elements of the same type in a specific order. They are accessed randomly using an integer to specify which element is required (although the elements may be of almost any type). Arrays may be fixed-length or expandable.
- A **list** is similar to an array, but its contents are strung together by a series of references to the next element.

Composite Data Types

- **Record** (also called tuple or struct) are among the simplest data structures. A record is a value that contains other values, typically in fixed number and sequence and typically indexed by names. The elements of records are usually called *fields* or *members*.

Composite Data Types

- A **union** type definition will specify which of a number of permitted primitive types may be stored in its instances, e.g. "float or long integer". Contrast with a record, which could be defined to contain a float *and* an integer; whereas, in a union, there is only one type allowed at a time.

Composite Data Types

- A **union** type definition will specify which of a number of permitted primitive types may be stored in its instances, e.g. "float or long integer". Contrast with a record, which could be defined to contain a float *and* an integer; whereas, in a union, there is only one type allowed at a time.
- A **tagged union** (also called a variant, variant record, discriminated union, or disjoint union) contains an additional field indicating its current type, for enhanced type safety.

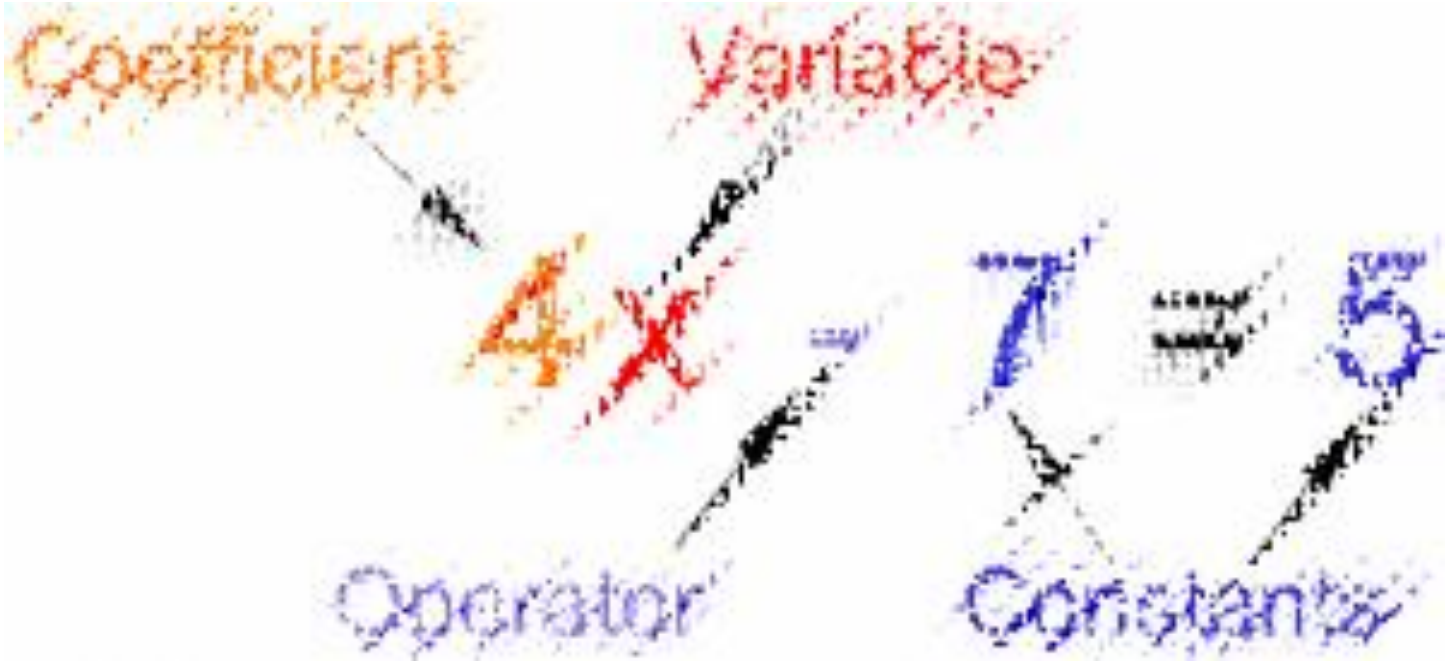
Composite Data Types

- A **set** is an abstract data structure that can store certain values, without any particular order, and no repeated values. Values themselves are not retrieved from sets, rather one tests a value for membership to obtain a boolean "in" or "not in".

Composite Data Types

- An **object** contains a number of data fields, like a record, and also a number of subroutines for accessing or modifying them, called *methods*.

Composite Data Types



Constants and Variables

- Let us reconsider two basic “well known” functions:
  - Monomial: x^n
  - Exponential: n^x

Constants and Variables

Constants and Variables

- Let us reconsider two basic "well known" functions:
  - Monomial: x^n
  - Exponential: n^x
- They are composed of two fundamental units – a *constant* and a *variable*.

- In mathematics, a constant is the term used to describe a parameter that never varies such as the number π.

Constants and Variables

- In mathematics, a constant is the term used to describe a parameter that never varies such as the number π.
- An unspecified constant is given an alphabetical character and is usually assigned from the beginning of the alphabet, such as "*a*".

Constants and Variables

- In mathematics, a constant is the term used to describe a parameter that never varies such as the number π.
- An unspecified constant is given an alphabetical character and is usually assigned from the beginning of the alphabet, such as "*a*".
- On the other hand, a variable is an alphabetical character that represents a number that is unspecified or unknown and is usually assigned from the end of the alphabet, such as "*x*".

Constants and Variables

- In mathematics, a constant is the term used to describe a parameter that never varies such as the number π.
- An unspecified constant is given an alphabetical character and is usually assigned from the beginning of the alphabet, such as "*a*".
- On the other hand, a variable is an alphabetical character that represents a number that is unspecified or unknown and is usually assigned from the end of the alphabet, such as "*x*".
- For example, *a*, *b*, and *c* are all constants whereas *x* is unknown in the famous *expression* known as the quadratic equation:

$$ax^2 + bx + c = 0$$

Constants and Variables

Definition: Expression (CS)

A statement combining values, constants, variables, operators, and functions, interpreted according to the syntax of the programming language at issue, which computes and then produces an *evaluation* (numerical, string, logical, *etc.*)

Expression

For most programming languages:

- A program constant is an immediate value found in an expression.

Variables in Programming

Variables in Programming

For most programming languages:

- A program constant is an immediate value found in an expression.
- A program variable is a **memory location**, associated with a symbolic name (an *identifier*), which contains some known or unknown information.

Variables in Programming

For most programming languages:

- A program constant is an immediate value found in an expression.
- A program variable is a **memory location**, associated with a symbolic name (an *identifier*), which contains some known or unknown information.
- The value of the variable may change during execution (for example, by setting, retrieving and updating data).

Variables in Programming 101

For most programming languages:

- A program constant is an immediate value found in an expression.
- A program variable is a **memory location**, associated with a symbolic name (an *identifier*), which contains some known or unknown information.
- The value of the variable may change during execution (for example, by setting, retrieving and updating data).

These definitions do not necessarily correspond to that given in mathematics insofar as a computing variable may not be part of an equation or formula as is always the case with a mathematical variable.

Variables in VB

The As clause in the declaration statement allows you to define the data type or object type of the variable you are declaring. You can specify any of the following types for a variable:

- An elementary data type, such as `Boolean`, `Long`, or `Decimal`
- A composite data type, such as an array or structure
- An object type, or class, defined either in your application or in another application
- A .NET Framework class, such as Label or TextBox
- An interface type, such as IComparable or IDisposable

You can declare several variables in one statement without having to repeat the data type. In the following statements, the variables `i`, `j`, and `k` are declared as type `Integer`, `l` and `m` as `Long`, and `x` and `y` as `Single`:

Declaring Data Types in VB

VARIABLES

Variables are declared using the Dim keyword, Dim is short for (Dimension).

SYNTAX

```
Dim MyVariable As DataType
```

The above code creates a variable called MyVariable with no value. The example below creates two variables with data type of string and one of type integer I will use these variables throughout.

```
Dim Name As String = "thecodingguys"
Dim Year As Integer = 2013
```

Variables in VB

Exercise

Variable Name	Valid?	Invalid?	Good Variables Name? If not why?
GPA			
Count#1			
Sgrosspay			
GradePointAverage			
Attempted.Hours			
Attempted-Hours			
Interest Rate			
A2			
2A			
End			
Printitout			
MONEYAFTERALLTAXES			

Variables in VB

http://virtual.parkland.edu/kcouch/CIS122/Week2/Exercises/goodname%20badname%20answers.htm

Exercise

Variable Name	Valid?	Invalid?	Good Variables Name? If not why?
GPA	YES		Yes. Assuming the application has something to do with grades this abbreviation is an acceptable variable name.
Count#1		YES	Invalid character #
Sgrosspay		YES	Invalid character $
GradePointAverage	YES		Yes - Descriptive
Attempted.Hours		YES	Invalid character . The period has special meaning in Visual Basic and may not be used in programmer supplied names.
Attempted-Hours		YES	Invalid character -
Interest Rate		YES	You may not use a space in a variable name
A2	YES		Not good name – Not descriptive
2A		YES	Must begin with a letter
End		YES	End is a reserved word. Reserved words have a special meaning to the compiler and may not be used for programmer supplied names
Printitout	YES		Not the greatest name. It is not clear what it means.
MONEYAFTERALLTAXES	YES		Not the greatest. All uppercase is valid but not helpful in reading the variable name. Also, while it's length does not violate the rules it is probably too long to have to work with in code.

http://virtual.parkland.edu/kcouch/CIS122/Week2/Exercises/goodname%20badname%20answers.htm

Arrays

Arrays are similar to variables, however arrays can hold more than one value.

Syntax

```
Dim MyArray() As DataType = {Values Comma Separated}
```

Example

```
Dim MyGamesOf2013() As String = {"GTAV", "Battlefield 3"}
Dim MyMoviesOf2013() As String = New String(3) {"The Amazing
Spiderman", "The Expendables", "X-Men", "Rise of the planet of the
apes"}
```

Arrays in VB

https://www.thecodingguys.net/resources/visual-basic-cs.pdf

STRINGS

CONCATENATION

Concatenation is done through the & symbol, like the following:

```
Console.WriteLine("Hello " & "World")
```

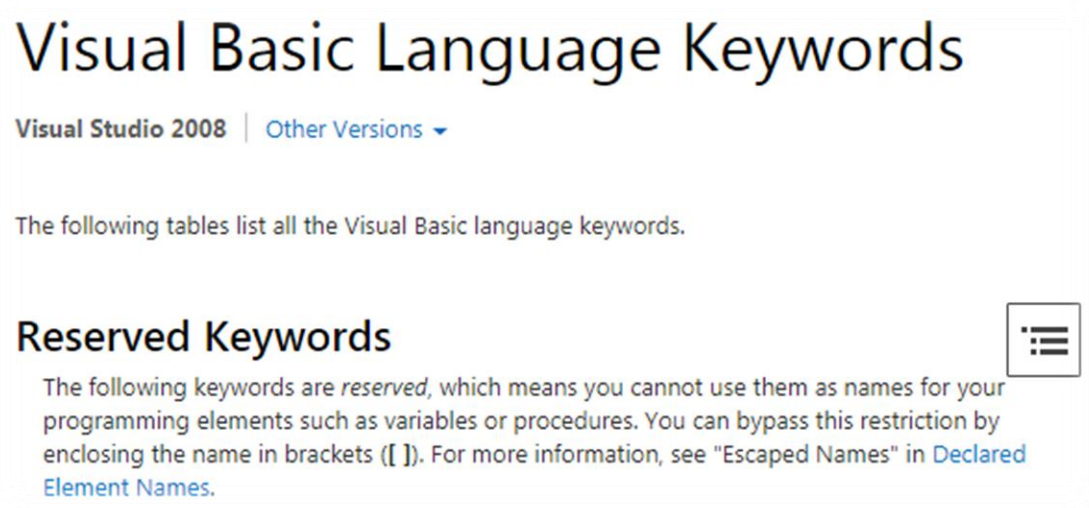
STRING.FORMAT

Formats a string, the following example prints out *£5,00*

```
Console.WriteLine(String.Format("{0:C}", 5))
```

In the example above, we want to format the number 5 and show the currency symbol. The {0:C} is the formatting we want to do, in this case it means format the first argument (0) and apply the currency symbol. Many more formatting types are available see this MSDN reference.

Strings in VB



Visual Basic Language Keywords

Visual Studio 2008 | Other Versions

The following tables list all the Visual Basic language keywords.

Reserved Keywords

The following keywords are *reserved*, which means you cannot use them as names for your programming elements such as variables or procedures. You can bypass this restriction by enclosing the name in brackets ([ ]). For more information, see "Escaped Names" in Declared Element Names.

AddHandler	AddressOf	Alias	And
AndAlso	As	Boolean	ByRef
Byte	ByVal	Call	Case
Catch	CBool	CByte	CChar
CDate	CDec	CDbl	Char
CInt	Class	CLng	CObj
Const	Continue	CSByte	CShort
CSng	CStr	CType	CUInt
CULng	CUShort	Date	Decimal
Declare	Default	Delegate	Dim
DirectCast	Do	Double	Each
Else	ElseIf	End	**EndIf**
Enum	Erase	Error	Event
Exit	False	Finally	For
Friend	Function	Get	GetType
GetXMLNamespace	Global	**GoSub**	GoTo
Handles	If	IIf	Implements
Imports (.NET Namespace and Type)	Imports (XML Namespace)	In	Inherits
Integer	Interface	Is	IsNot
Let	Lib	Like	Long
Loop	Me	Mod	Module
MustInherit	MustOverride	MyBase	MyClass
Namespace	Narrowing	New	Next
Not	Nothing	NotInheritable	NotOverridable
Object	Of	On	Operator
Option	Optional	Or	OrElse
Overloads	Overridable	Overrides	ParamArray
Partial	Private	Property	Protected
Public	RaiseEvent	ReadOnly	ReDim
REM	RemoveHandler	Resume	Return
SByte	Select	Set	Shadows
Shared	Short	Single	Static
Step	Stop	String	Structure
Sub	SyncLock	Then	Throw
To	True	Try	TryCast
TypeOf	**Variant**	**Wend**	UInteger
ULong	UShort	Using	When
While	Widening	With	WithEvents
WriteOnly	Xor	#Const	#Else
#ElseIf	#End	#If	=
&	&=	*	*=
/	/=	\	\=
^	^=	+	+=
-	-=	>> Operator (Visual Basic)	>>= Operator (Visual Basic)
<<	<<=		

Note:

EndIf, **GoSub**, **Variant**, and **Wend** are retained as reserved keywords, although they are no longer used in Visual Basic. The meaning of the **Let** keyword has changed. **Let** is now used in LINQ queries. For more information, see Let Clause (Visual Basic).

Reserved Kew Words in VB

Microsoft

https://msdn.microsoft.com/en-us/library/ksh7h19t(v=vs.90).aspx

Unreserved Keywords

The following keywords are not reserved, which means you can use them as names for your programming elements. However, doing this is not recommended, because it can make your code hard to read and can lead to subtle errors that can be difficult to find.

Aggregate	Ansi	Assembly	Auto
Binary	Compare	Custom	Distinct
Equals	Explicit	From	Group By
Group Join	Into	IsFalse	IsTrue
Join	Key (Visual Basic)	Mid	Off
Order By	Preserve	Skip	Skip While
Strict	Take	Take While	Text
Unicode	Until	Where	#ExternalSource
#Region			

Unreserved Kew Words in VB

Microsoft

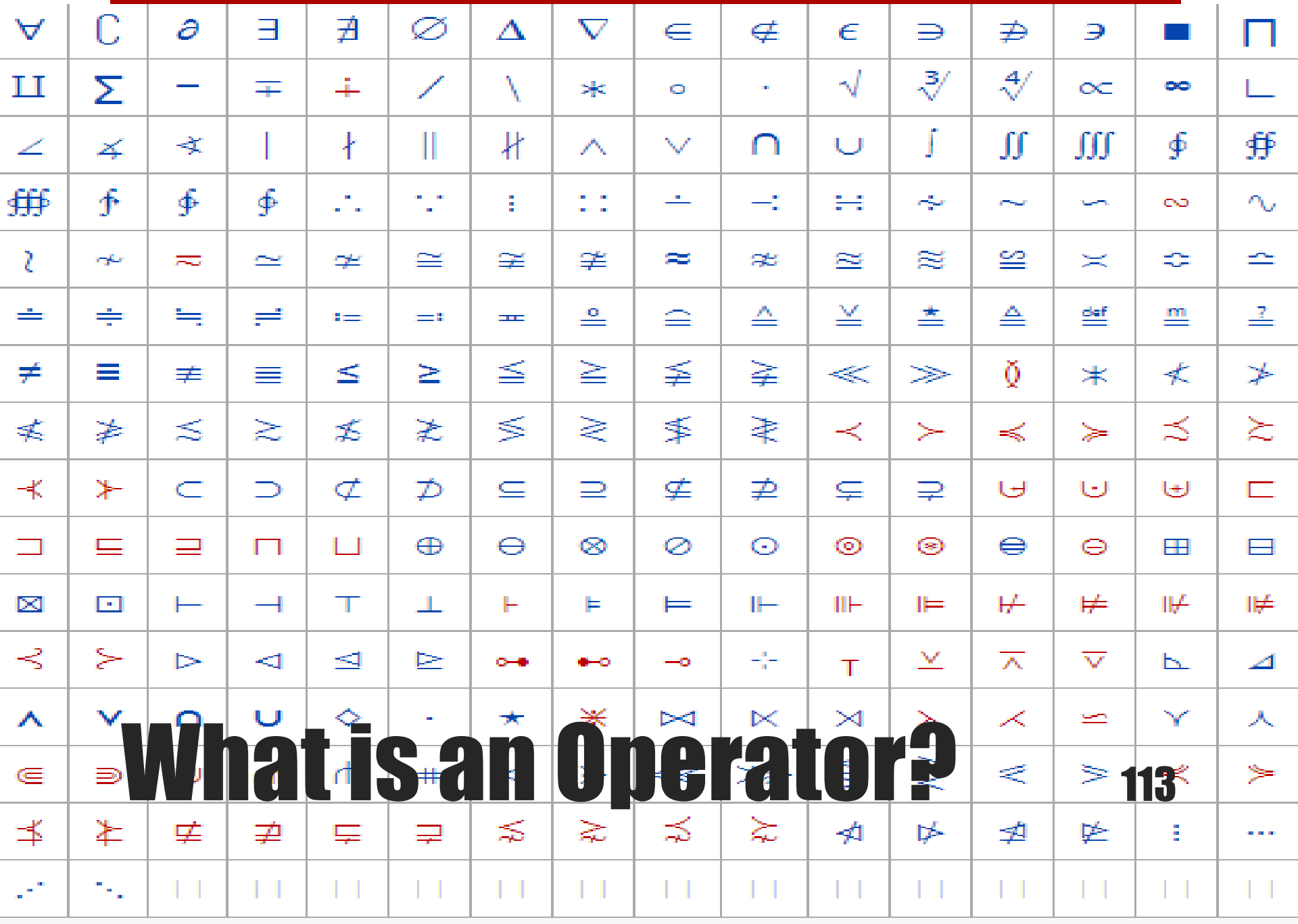
https://msdn.microsoft.com/en-us/library/ksh7h19t(v=vs.90).aspx

```
Module DataTypes
   Sub Main()
      Dim b As Byte
      Dim n As Integer
      Dim si As Single
      Dim d As Double
      Dim da As Date
      Dim c As Char
      Dim s As String
      Dim bl As Boolean
      b = 1
      n = 1234567
      si = 0.12345678901234566
      d = 0.12345678901234566
      da = Today
      c = "U"c
      s = "Me"
      If ScriptEngine = "VB" Then
         bl = True
      Else
         bl = False
      End If
      If bl Then
         'the oath taking
          Console.Write(c & " and," & s & vbCrLf)
          Console.WriteLine("declaring on the day of: {0}", da)
          Console.WriteLine("We will learn VB.Net seriously")
          Console.WriteLine("Lets see what happens to the floating point variables:'
          Console.WriteLine("The Single: {0}, The Double: {1}", si, d)
      End If
      Console.ReadKey()
   End Sub
```

Examples of Data Types in VB

https://www.tutorialspoint.com/vb.net/vb.net_data_types.htm

OPERATORS


What is an Operator?

What is an Operator?

Definition:

An operator in a programming language is a symbol that tells the compiler or interpreter to perform specific mathematical, relational or logical operation and produce final result

Highest	P	Parentheses	( )
	E	Exponentiation	**
	M	Multiplication	*
	D	Division	/
	A	Addition	+
Lowest	S	Subtraction	-

Priority of Arithmetic Operators

Operator Precedence and Associativity

Operator	Associativity
()	Left to right
++ -- ! + - (unary)	Right to left
* / %	Left to right
+ -	Left to right
< <= > >=	Left to right
== !=	Left to right
&&	Left to right
||	Left to right
?:	Right to left
= += -= *= /=	Right to left

Ultimate Programming Tutorials

Priority of VB Operators

http://www.ultimateprogrammingtutorials.info/2017/01/operator-precedence-and-associativity.html

Arithmetical Operators

Operators	Description	Example	Result
+	Add	5+5	10
-	Substract	10-5	5
/	Divide	25/5	5
\	Integer Division	20\3	6
*	Multiply	5*4	20
^	Exponent (power of)	3^3	27
Mod	Remainder of division	20 Mod 6	2
&	String concatenation	"George"&" "&"Bush"	"George Bush"

Relational Operators

Operators	Description	Example	Result
>	Greater than	10>8	True
<	Less than	10<8	False
>=	Greater than or equal to	20>=10	True
<=	Less than or equal to	10<=20	True
<>	Not Equal to	5<>4	True
=	Equal to	5=7	False

Logical Operators

Operators	Description
OR	Operation will be true if either of the operands is true
AND	Operation will be true only if both the operands are true

Filter

- Operators
  - Operator Precedence
  - Operators Listed by Functionality
  - Data Types of Operator Results
  - DirectCast Operator
  - TryCast Operator
  - New Operator
  - Arithmetic Operators
  - Assignment Operators
  - Bit Shift Operators
  - Comparison Operators
  - Concatenation Operators
  - Logical-Bitwise Operators
  - Miscellaneous Operators
- Properties
- Queries
- Statements
- XML Comment Tags
- XML Axis Properties
- XML Literals
- Error Messages

Operators (Visual Basic)

Invalid Date • 1 minutes to read • Contributors

In This Section

Operator Precedence in Visual Basic

Operators Listed by Functionality

Data Types of Operator Results

DirectCast Operator

TryCast Operator

New Operator

Arithmetic Operators

Assignment Operators

Bit Shift Operators

Comparison Operators

Concatenation Operators

Logical/Bitwise Operators

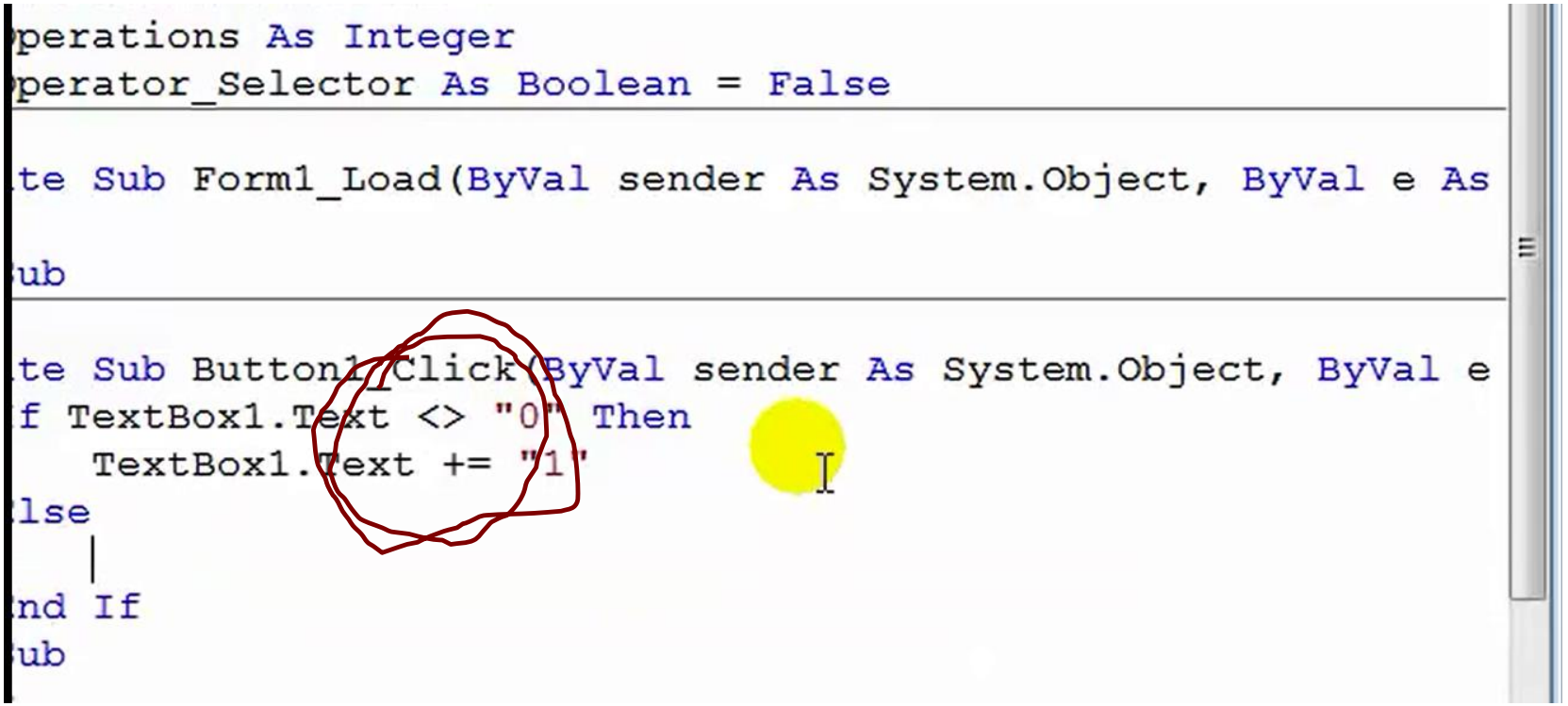
Miscellaneous Operators

Your Rosetta Stone for VB Operators

The Complete List of Operators in VB

Microsoft

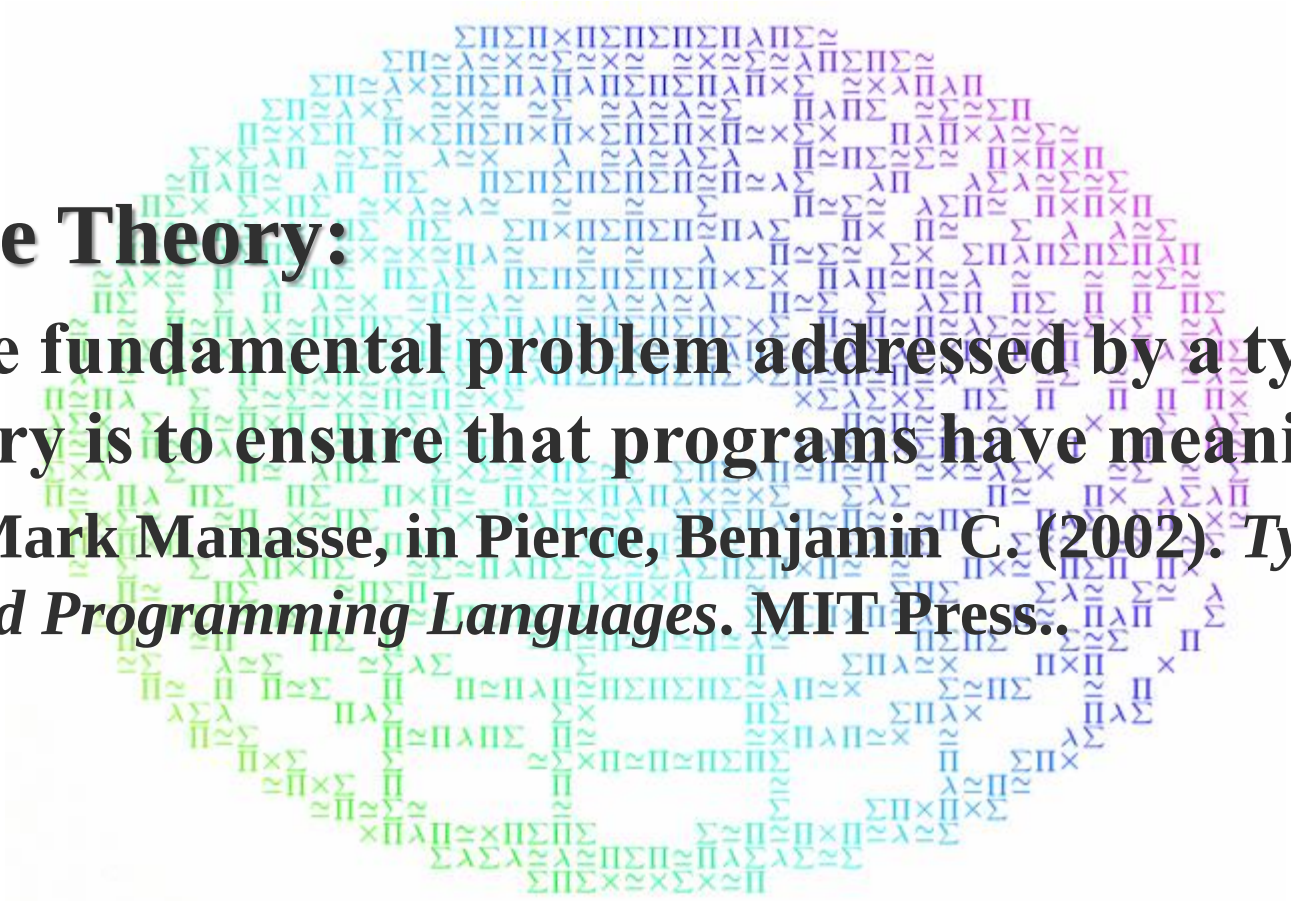
Example: From Building a Calculator Video



The Complete List of Operators in VB

Microsoft https://docs.microsoft.com/en-us/dotnet/visual-basic/language-reference/operators/

End Operator Unit



Type Theory:

"The fundamental problem addressed by a type theory is to ensure that programs have meaning."

- Mark Manasse, in Pierce, Benjamin C. (2002). *Types and Programming Languages*. MIT Press..

Recall Type Theory

Typing

Typing assigns a data type, giving meaning to a sequence of bits such as a value in memory or some object such as a variable.

Definition

- The hardware of a general purpose computer is unable to discriminate between, for example, a memory address and an instruction code, or between a character, an integer, or a floating-point number, because it makes no intrinsic distinction between any of the possible values that a sequence of bits might *mean*.

Typing

- The hardware of a general purpose computer is unable to discriminate between, for example, a memory address and an instruction code, or between a character, an integer, or a floating-point number, because it makes no intrinsic distinction between any of the possible values that a sequence of bits might *mean*.
- Associating a sequence of bits with a type conveys that meaning to the programmable hardware to form a *symbolic system* composed of that hardware and some program.

Typing

Type Checking

The process of verifying and enforcing the constraints of types. This may be implemented either at compile-time (a static check) or run-time (a dynamic check).

Definition

A programming language is said to be **dynamically** typed, or just '**dynamic**', when the majority of its **type** checking is performed at run-time as opposed to at compile-time. In **dynamic typing**, **types** are associated with values not variables. Oct 4, 2009

static typing - What is the difference between statically typed and ...
stackoverflow.com/.../what-is-the-difference-between-statically-typed-and-dynamically-t...

Dynamic vs. Static Type Checking

VB is both *statically and dynamically typed*

Dynamic vs. Static Type Checking

https://www.thecodingguys.net/resources/visual-basic-cs.pdf

Remainder of VB Syntax

Presentation Terminated