

**PENANG SANGAM HIGH SCHOOL
P.O.BOX 44, RAKIRAKI**

LESSON NOTES 19

Year/Level: 12 **Subjects: Computer Studies**

Strand:	CE Strand: CE 2 Application Packages
Sub-strand:	CE 12.2.1 Visual Basic.NET Programming (VB NET)
Content Learning Outcome:	Data types, Variable naming conventions, Scope and lifetime of variables, Visual Basic Operators, Logical Operators, Visual Basic Functions, Comparing Uppercase and Lowercase Characters

Lesson Notes

Data types

Boolean – true or false

Char – single Unicode character 0 to 65535 (unsigned)

Date – holds values that specify dates and time

Decimal – (28 -29 numbers)

Double – holds floating point numbers (15 – 16 number)

Integer – whole number

String – holds a sequence of Unicode characters

Long – very large positive and negative whole numbers

Short – signed integer number between -32768 and 32767

Single – holds floating point numbers e.g. 6.142F (7 numbers)

Variable Naming Conventions

- Identifiers must be meaningful.** Choose a name that clearly indicates its purpose. Do not abbreviate unless the meaning is obvious and do not use very short identifiers, such as X or Y.
- Include the class** (data type) of the variable.
- Begin with the data type and then capitalize** each successive word of the name. Always use mixed case for variables; uppercase for constants.

Example

```
Dim strResidentialAddress As String ' to store residential address
Dim intCounter As Integer 'to store integer
Dim intMax As Integer = 100 ' to store a maximum integer of 100
Dim dblAverage As Double ' to store the average
Const dblDiscount_Rate As Decimal = 0.2D ' to store constant rate of 20%
```

Scope and Lifetime of Variables

A variable may exist and be visible for an entire project, for only one form, or for only one procedure.

The visibility of a variable is referred to as its **scope**.

Module-level variables, also called **class-level variables**, are accessible from all procedures of a form.

A **local variable** may be used only within the procedure in which it is declared.

Block-level variable is used only within a block of code inside a procedure.

Comment statements begin with the keyword **Rem** or a single quote ('). For example:

```
Rem This is a remark
' This is also a remark
x = 2 * y ' another way to write a remark or comment
```

Visual Basic Operators

Operator	Operation
^	Exponentiation
*/	Multiplication and Division
\	Integer division (truncates) Mod/Modulus
+ -	Addition and Subtraction

Logical Operators

Operator	Operation	Effect
Not	Logical not	Simply negates an operand
And	Logical and	Returns a true if both operand is true Else it returns a False
Or	Logical or	Returns a True if either of its operands is True, Else it returns False.

Visual Basic Functions

Visual Basic offers a rich assortment of built-in **functions**. The on-line help utility will give you information on any or all of these functions and their use. Some examples are:

Function	Value Returned
Abs	Absolute value of a number
Date	Current date as a text string
Format	Date or number converted to a text string
Parse	convert value to another format
Now	Current time and date
Rnd	Random number
Sqr	Square root of a number
Time	Current time as a text string
ToString	Converts to String
Val	Numeric value of a given text string

Comparing Uppercase and Lowercase Characters

When comparing strings, the case of the characters is important. An uppercase *Y* is not equal to a lowercase *y*. Because the user may type a name or word in uppercase, in lowercase, or as a combination of cases, we must check all possibilities. The best way is to use the **ToUpper** and **ToLower** methods of the String class, which return the uppercase or lowercase equivalent of a string, respectively.

Example:
`TextString.ToUpper()`
`TextString.ToLower()`