



# Introduction to Visual Basic

**Dr. A. Malathi M.Sc.,M.Phil.,Ph.D.,M.Sc(Psych)**

**Assistant Professor**

**PG and Research Department of Computer Science**

**Government Arts College**

**Coimbatore - 641018**

# UNIT I

- Introduction - VB Environment – VB Fundamental: Constants, Variables, Data Types, Data Declaration – Operators – Expressions – Library Functions.

# History of Programming Languages

- Machine language
- Procedure-oriented languages
- Object-oriented languages
- Event-driven languages
- Natural languages

# History of Programming Languages

- Machine Language
  - different for each computer processor
- Procedure-Oriented Languages
  - BASIC
  - FORTRAN
  - COBOL
  - Pascal
  - C
  - C++
  - JAVA
  - Ada

# History of Programming Languages

- Object-oriented languages
  - Smalltalk
  - C++
  - Java
  - Ada 95
- Event-driven languages
  - Visual Basic
  - most Visual languages

# History of Programming Languages

- Natural languages
  - computer will accept a user's native or natural language, such as English
  - AI, ML, DL
  - Researchers continue to work in this area

# Background of Visual Basic

- BASIC
  - Beginner's All-purpose Symbolic Instruction Code
  - By John Kemeny and Thomas Kurtz
  - Designed to teach programming to beginners
  - BASIC language interpreter, one of Microsoft's first products

# Background of Visual Basic

- BASIC
  - Included with DOS
  - QBASIC - included with DOS Version 5 - subset of BASIC
- Visual Basic
  - 1992 - Visual Basic 1 for Windows 3
  - Visual Basic 4 - for Windows 95
  - Visual Basic 5 - for Office 97
  - Visual Basic 6 - for Windows 98 & Office 2000

# Visual Basic Editions

- Learning Edition
- Professional Edition
- Enterprise Edition
- Included in Visual Studio suite (analogy Office for programmers)

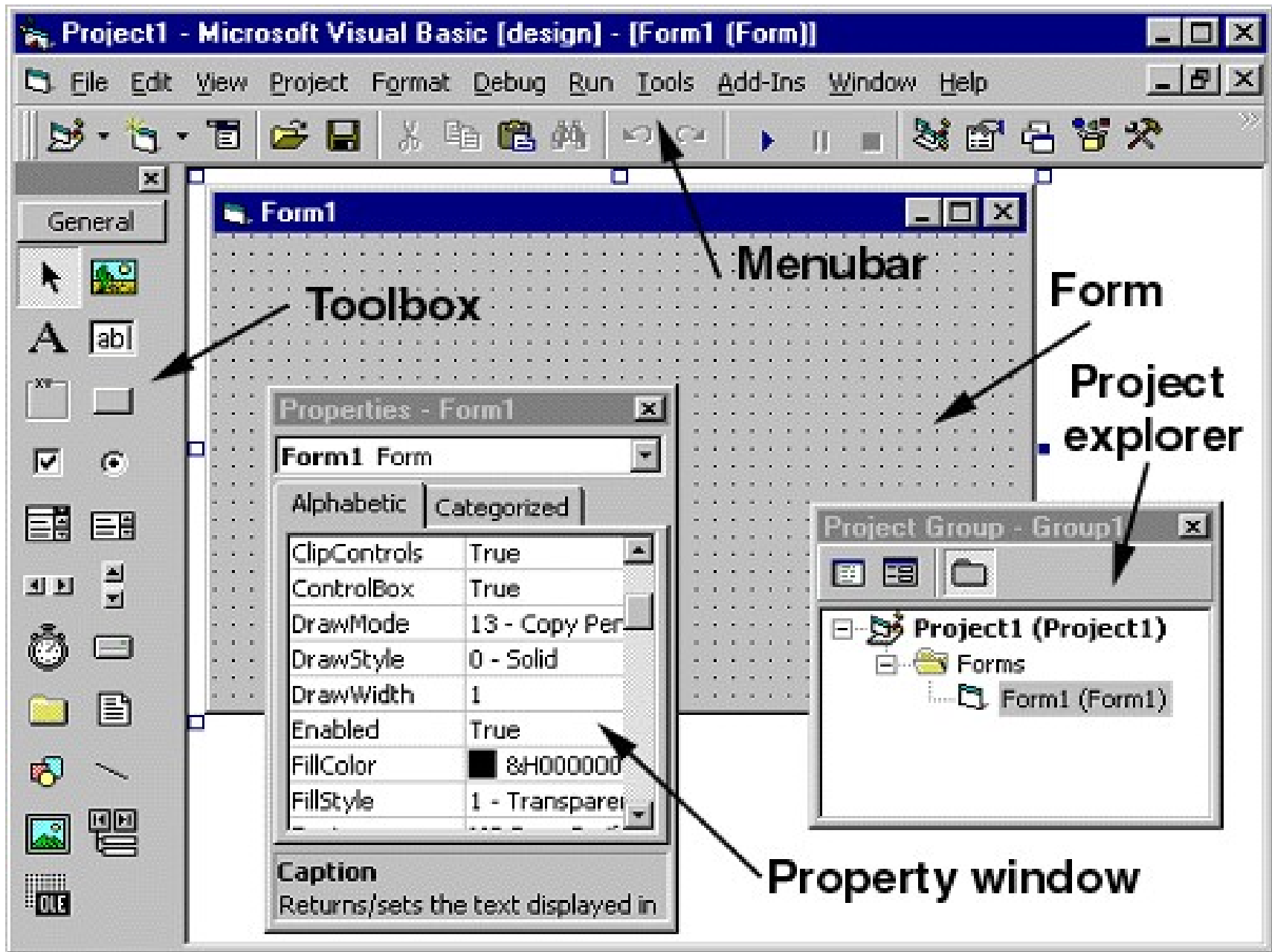
# Introduction to Visual Basic

## Hands-On Exercise 1

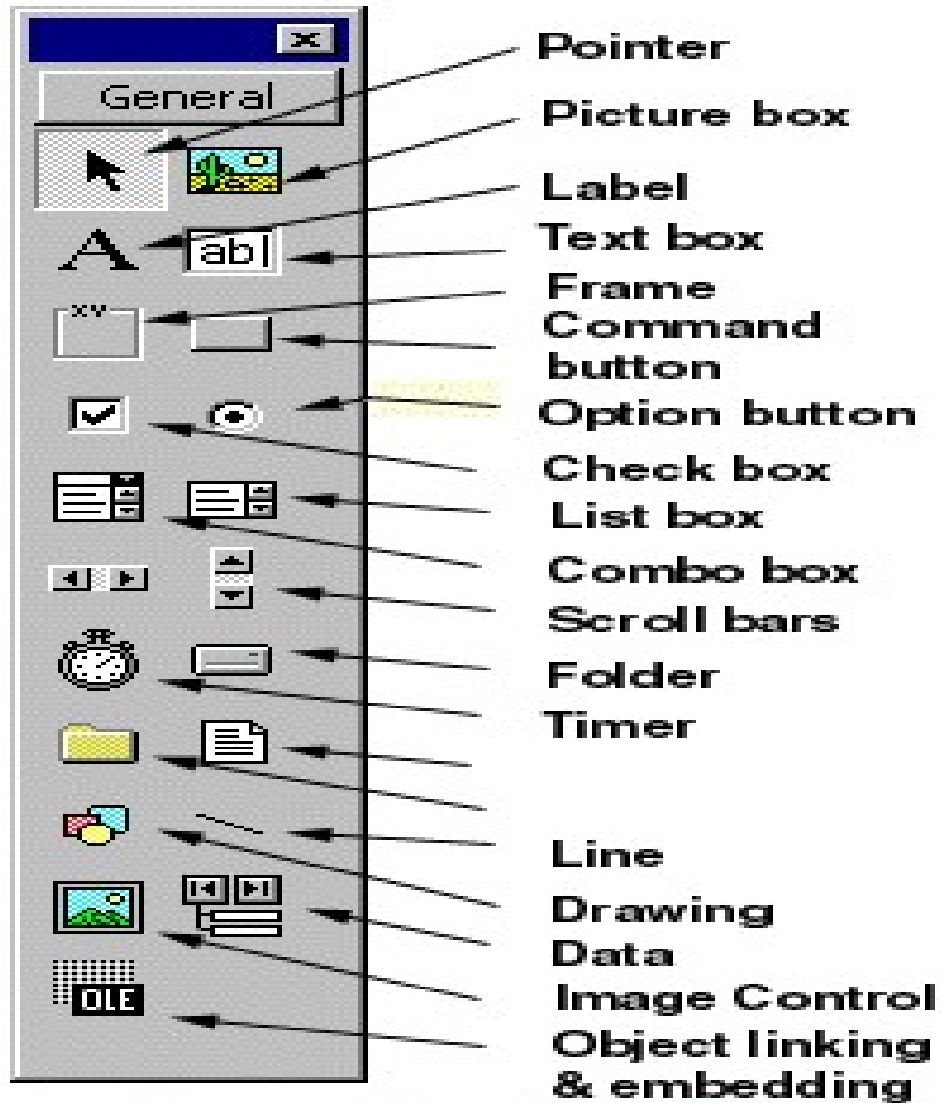
- Start Microsoft Visual Basic 6.0
- Open the Welcome Project
- Open the Welcome Form
- Run the Welcome Project
- End the Welcome Project
- Exit Visual Basic

# VB Environment





# TOOLBOX



**Diagram showing Toolbox**

**components**

**Properties - Form1** [X]

Form1 Form

Alphabetic | Categorized

(Name)	Form1
Appearance	1 - 3D
AutoRedraw	False
BackColor	<input type="checkbox"/> &H800000
BorderStyle	2 - Sizable
<b>Caption</b>	Form1
ClipControls	True
ControlBox	True
DrawMode	13 - Copy Pe
DrawStyle	0 - Solid
DrawWidth	1
Enabled	True

**Caption**  
Returns/sets the text displayed

# Visual Basic 6 Interactive Development Environment

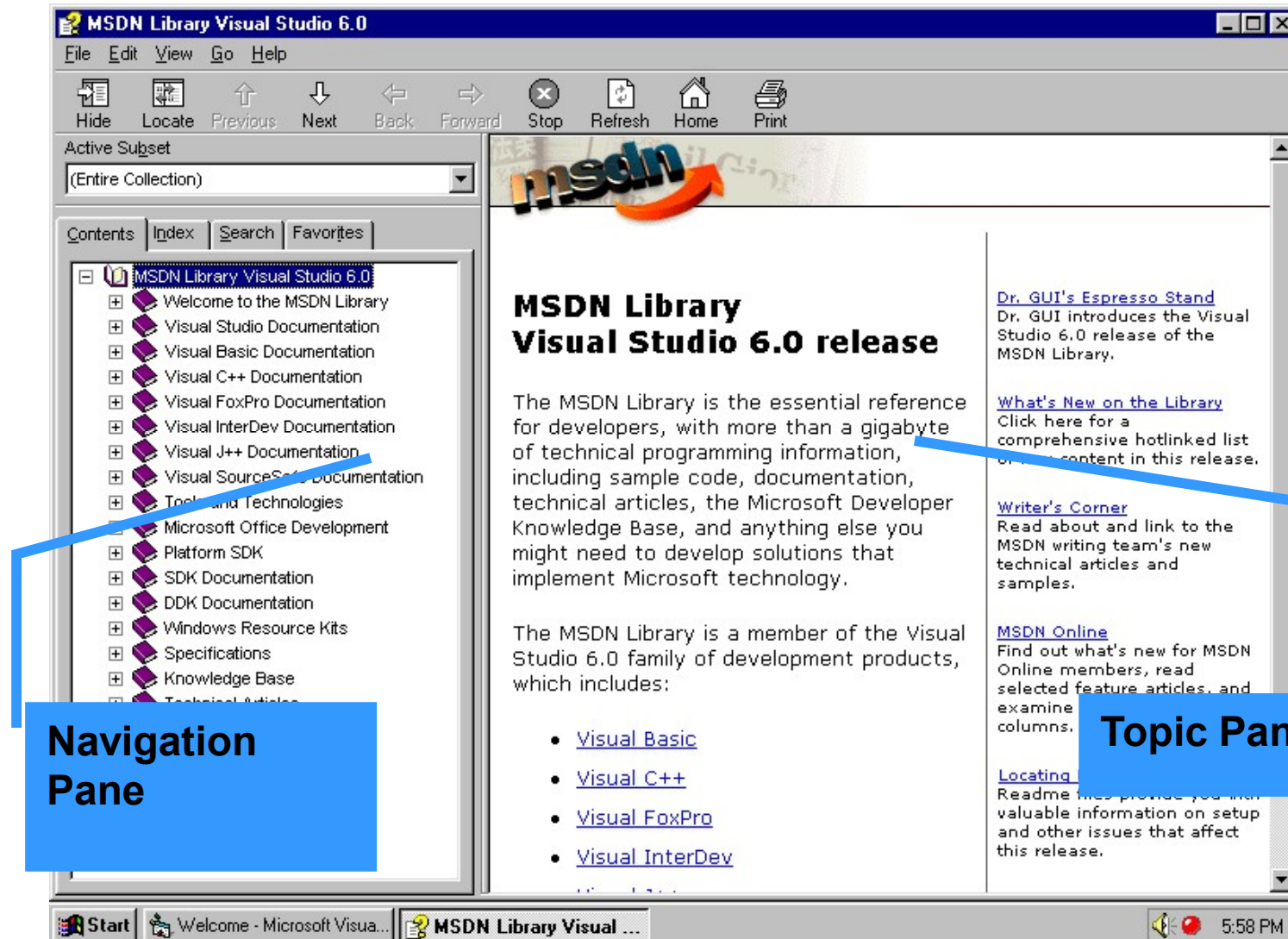
The screenshot displays the Visual Basic 6 IDE interface. At the top is a menu bar and a toolbar. Below the toolbar is the main workspace, which is divided into several windows:

- Toolbox:** Located on the left side, it contains various controls like buttons, text boxes, and clocks. A callout box labeled "Toolbox" points to this area.
- Form Designer Window:** The central window showing a visual representation of the "AlarmForm" with a clock icon and the text "1:30pm". A callout box labeled "Form Designer Window" points to this window.
- Code Editor Window:** Below the Form Designer, it shows the source code for the form. A callout box labeled "Code Editor Window" points to this window. The code includes:

```
Option Explicit
Dim AlarmTime
Const conMinimized = 1

Private Sub Form_Click()
    AlarmTime = InputBox("Enter alarm time", "VF")
    If AlarmTime = "" Then Exit Sub
    If Not IsDate(AlarmTime) Then
        MsgBox "The time you entered"
    Else
```
- Project Explorer:** Located in the top right, it shows the project structure. A callout box labeled "Project Explorer" points to this window.
- Properties window:** Located in the middle right, it shows the properties for the selected "AlarmForm" control. A callout box labeled "Properties window" points to this window. The "Caption" property is set to "Alarm Clock".
- Form Layout Window:** Located in the bottom right, it shows a small preview of the form layout. A callout box labeled "Form Layout Window" points to this window.

# MSDN Library



# MSDN Library

- Visual Basic Documentation
  - Documentation Map
  - What's New
  - Programmer's Guide
  - Samples
  - Reference Guidebooks
  - Component Tools Guide
  - Data Access Guide

# VB Fundamentals

- Constant
- Variable
- Data Types
- Declaration

# Variables and Constants

## Variable

- Memory locations that hold data that can be changed during project execution
- Example: customer's name
- **Named Constant**
  - Memory locations that hold data that cannot be changed during project execution
  - Example: sales tax rate

# Variables and Constants

- In Visual Basic when you declare a Variable or Named Constant
  - An area of memory is reserved
  - A name is assigned called an Identifier
  - Follow rules and naming conventions

# Data Types

- Boolean
- Byte (0 to 255)
- Char
- Date
- String
- Decimal
- Object
- Short (-32,768 to 32,767)
- Integer (-2,147,483,648 to 2,147,483,647)
- Long (larger whole numbers)
- Single (floating point accuracy to 6 digits)
- Double (floating point accuracy to 14 digits)

# Data Types – Memory Usage

- Boolean – 2 bytes
- Byte – 1 byte
- Char – 2 bytes
- Date – 8 bytes
- String – varies
- Decimal – 16 bytes
- Object – 4 bytes
- Short – 2 bytes
- Integer – 4 bytes
- Long – 8 bytes
- Single – 4 bytes
- Double – 8 bytes

# Default Values for Data Types

<b>Data type</b>	<b>Default (Initial) value</b>
All numeric types	Zero (0)
Boolean	False
Char	Binary 0
String or Object	Empty
Date	12:00 a.m. on January 1, 0001

# Declaring Variables

- Declared inside a procedure using a Dim statement
- Declared outside a procedure using Public, Private or Dim statements
- Always declare the variable's data type
- May declare several variables with one statement
- Use IntelliSense to assist in writing statements

# Data Declaration

- The Dim statement is used to associate variables with specific data types.
- The Dim statement is written as  
**Dim** variable name 1 **As** data type 1, variable name 2 **As** data type 2, etc.

Examples:

```
Dim Counter As Integer
```

```
Dim Area As Single
```

```
Dim StudentName As String
```

```
Dim StudentName As String * 30
```

# Named Constant

- It is also possible to define named constants in Visual Basic.
- variables can be reassigned different values within a program, whereas named constants remain unchanged throughout a program.
- The Const statement is used to declare a named constant.
- This statement has the general form
- **Const** constant name **As** data type = value
- **Const** TaxRate **As** Single = 0.28

# Suffixes

<u>Suffix</u>	<u>Data Type</u>
%	integer & long integer
!	Single
#	double
\$	string

<u>Variable</u>	<u>Data Type</u>
Index%	integer Counter& long integer
TaxRate!	single
Ratio#	double
CustomerName\$	string

# Declaration Examples

```
Dim customerNameString    As String  
Private totalSoldInteger  As Integer  
Dim temperatureSingle    As Single  
Dim priceDecimal         As Decimal  
Private priceDecimal     As Decimal
```

# Conversion Methods

<b>Method</b>	<b>Convert To</b>
Integer.Parse	Integer
Decimal.Parse	Decimal
.ToString	String

# OPERATORS AND EXPRESSIONS

- Special symbols, called arithmetic operators, are used to indicate arithmetic operations such as addition, subtraction, multiplication, division and exponentiation.
- These operators are used to connect numeric constants and numeric variables, thus forming arithmetic expressions

# Arithmetic Operations

<b>Operator</b>	<b>Operation</b>
+	Addition
-	Subtraction
*	Multiplication
/	Division
\	Integer Division
Mod	Modulus – Remainder of division
^	Exponentiation

# Hierarchy of Operations

- Order of precedence in arithmetic expressions from highest to lowest
  1. Any operation inside parentheses
  2. Exponentiation
  3. Multiplication and division
  4. Integer division
  5. Modulus
  6. Addition and subtraction

# INSERTING PARENTHESES

- Note the use of parentheses to control order of precedence

**$3+4*2 = 11$       Multiply then add**  
 **$(3+4)*2 = 14$       Parentheses control: add then multiply**  
 **$8/4*2 = 4$  Same level, left to right: divide then multiply**

# Using Calculations in Code

- Perform calculations in assignment statements
- What appears on right side of assignment operator is assigned to item on left side
- Assignment operators  
=, +=, -=, \*=, /=, \=, &=

# Displaying Messages in Message Boxes

- Use Show method of MessageBox to display message box, a special type of window
- Arguments of Show method
  - Message to display
  - Optional Title Bar Caption
  - Optional Button(s)
  - Optional Icon

# String Expressions

- Numerical operations cannot be performed on string constants or string variables.
- strings and string variables can be concatenated (i.e., combined, one behind the other).
- In Visual Basic we use either the ampersand (&) or the plus sign (+) as a string concatenation operator.
- Str1 = "TEN"
- Str2 = "THOUSAND"
- Then the string expression
- Str1 & " " & str2 & " DOLLARS"

# ASSIGNING VALUES TO VARIABLES

- The equal sign (=) is used to assign a numeric or string value to a variable. The general form is
- Variable = Expression
- $X = 12.5$
- $C_{max} = X$
- $Area = 3.141593 * Radius ^ 2$
- Label = "Name: "

# DISPLAYING OUTPUT – THE Print STATEMENT

- The Print statement is used to display information within the currently active form, beginning in the upper left corner.
- it is very convenient for displaying the results of very simple programs, and it provides a way to view the results of small program segments during the development of a large project.
- The Print statement consists of the keyword Print, followed by a list of output items.
- The output items can be numeric constants, string constants, or expressions.
- Each new Print statement will begin a new line of output. An empty Print statement will result in a blank
- `Print "Name:", Student, X, (C1 + C2) / 2`

# Library Function

- Visual Basic contains numerous library functions that provide a quick and easy way to carry out many mathematical operations, manipulate strings, and perform various logical operations.
- These library functions are prewritten routines that are included as an integral part of the language.
- They may be used in place of variables within an expression or a statement.
- presents several commonly used library functions.
- A library function is accessed simply by stating its name, followed by whatever information must be supplied to the function, enclosed in parentheses.
- A numeric quantity or string that is passed to a function in this manner is called an argument.
- Once the library function has been accessed, the desired operation will be carried out automatically.
- The function will then return the desired value

# Examples

- Abs()
- Cos()
- Len()
- Rnd()
- Str()
- Sqt()

# References

- SCHAUM'S OUTLINE OF THEORY AND PROBLEMS OF PROGRAMMING WITH VISUAL BASIC
- [http://media.pearsoncmg.com/ph/esm/deitel/vb htp 2010/odeexamples.html](http://media.pearsoncmg.com/ph/esm/deitel/vb_htp_2010/odeexamples.html)

# The End