

Programming in Visual Basic

The Basics

- Computer program- set of instructions that enable computer to solve a problem or perform a task.
- Algorithm- set of well-defined steps for performing a task or solving a problem.
- Two main methods of programming:
 - 1. Procedural- set of statements that are executed by the computer, one after the other.
 - 2. Object-oriented where focus is centered on creating objects.
 - A. Object is a programming element that contains data and actions.
 - B. Attributes are data contained in the object
 - Note: In VB, attributes are called properties
 - C. Methods are actions that an object performs (pre-defined).
- Event-Driven Programming:
 - 1. operate in a GUI environment
 - 2. event is an action, clicking the mouse (User invoked).
 - 3. When event occurs, application responds by executing a special type method called an Event Procedure.
 - 4. Process is the design and writing of event procedures.

- In VB, a control (label, text box, picture box, ect.) are considered a class.
- Class is a template for an object which defines supported properties, methods, and events.
- Each form is stored as a separate file called a form module.
- Form module contains information about the control instances drawn on the form.
 - Also, contains statements that are executed when user interacts with form's control instances (ex. clicking buttons).
- Standard module contains statements that several forms can use.
- Each module contains specific statements written to perform some specific action.
- Project file lists each form and standard module (file) used in the program.
 - Also, information about the different types of controls used in each module.

- In Visual Basic, each program has a single project file, regardless of the number of modules.
- Event procedures contain a set of VB statements that execute when the user performs an action on an object, ex. clicking a command button.
- Steps for Developing an VB application
 - 1. Clearly define what application is to do. (Purpose, Input, Process, and Output).
 - 2. Visualize the application running on the computer and design its user interface.
 - 3. Make a list of controls needed.
 - 4. Define values of each control's relevant properties.
 - 5. Make a list of methods needed for each control.
 - 6. Create a flowchart or pseudo-code version of each method.
 - 7. Write the code for the methods on paper.
 - 8. Desk-check the code for errors.
 - 9. Start VB, create forms and other controls (form step 3).
 - 10. Create event procedures and other methods (step 7).
 - 11. Run application, correcting any syntax errors (grammar rules) found until you receive a clean compile.
 - 12. When you receive a clean compile, run the program with test data for input, then correct any run-time errors.

- **WAGE CALCULATOR PROGRAM**

- 1.
 - A. Purpose: To calculate the user's gross pay.
 - B. Input: Number of hours worked by hourly rate.
 - C. Process: Multiply number of hours worked by hourly pay rate. The result is the user's gross pay.
 - D. Output: Display a message indicating the user's gross pay that is formatted and written to a label.

- 2.

- 3. Object

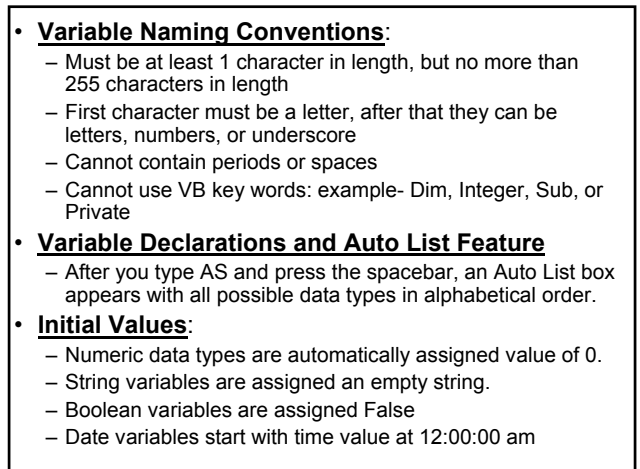
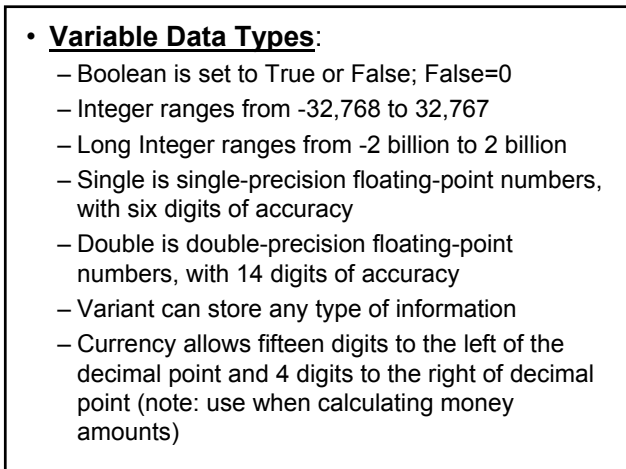
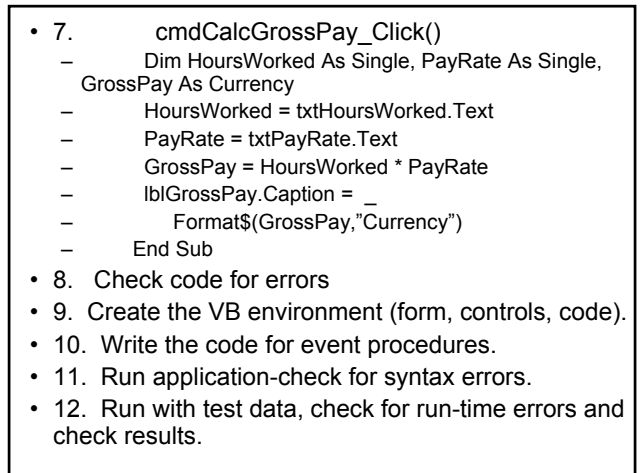
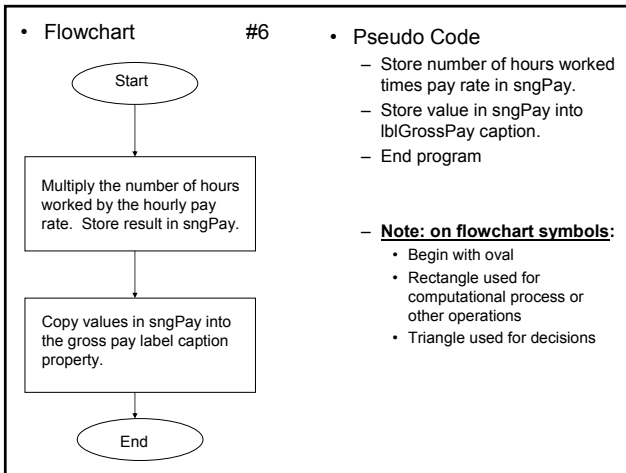
	Object Name
- Form	frmWageCalculate
- Label #1	lblHoursWorked
- Label #2	lblPayRate
- Label #3	lblGrossPayEarned
- Label #4	lblGrossPay
- Text Box #1	txtHoursWorked
- Text Box #2	txtPayRate
- Command Button #1	cmdCalcGrossPay
- Command Button #2	cmdClear
- Command Button #3	cmdExit

- 4. Label

	Property
- Form	Caption: Wage Calculator
- Label #1	Caption: Number of Hours Worked
- Label #2	Caption: Hourly Pay Rate
- Label #3	Caption: Gross Pay Rate
- Label #4	Caption: \$0.00
- Text Box #1	Clear Text
- Text Box #2	Clear Text
- Cmd. Button #1	Caption: Calculate Gross Pay
- Cmd. Button #2	Caption: Clear
- Cmd. Button #3	Caption: Exit

- 5. Method

	Description
- cmdCalcGrossPay_Click	- Multiplies number of hours worked by hourly rate. These values are entered into txtHoursWorked and txtPayRate text boxes. The result is stored in lblGrossPay caption property.
- cmdClear_Click	- Clears both text boxes and result label. Return focus to first text box.
- cmdExit_Click	- End the application.



- Arithmetic Operations
 - + Addition
 - - Subtraction
 - * Multiplication
 - Mod Modulus
 - / Division
 - \ Integer Division
 - ^ Exponentiation
- Order of Operation
 - Parentheses ()
 - ^
 - Negation -
 - *,/
 - \
 - Mod
 - +,-

- **Modulus Example:**
 - Dim Leftover AS Integer
 - Leftover = 17 Mod 3
 - Remainder stored in Leftover = 2
- **Option Explicit**
 - If not present, VB will not require that a variable be declared before it is used.
 - If you enter these statements:
 - Dim V1 As Integer, V2 As Integer
 - Temp = V1 + V2
 - VB will create a variant type and set it to zero
 - If present, VB will notify you the variable was not created.
- **Const ConstName As DataType = value**
 - Creates a constant which cannot be changed while the program is running

- **Calculation Ex. 1**
 - $5 * 2 * 4 = 13$
 - $2 ^ 3 * 4 + 3 = 35$
 - $10/2 - 3 = 2$
 - $8 + 12 * 2 - 4 = 28$
 - $6 - 3 * 2 + 7 - 1 = 6$
 - $10 \text{ Mod } 3 ^ 1 ^ 2 - 8 = -7$
- **Calculation Ex. 2**
 - $(5 + 2) * 4 = 28$
 - $10/(5-3) = 5$
 - $8 + 12 * (6-2) = 56$
 - $(6 - 3) * (2 + 7) / 3 = 9$
 - $-5 - 8 \text{ Mod } 4 + 7 * (2^2+2) = 37$
 - $8 + 10 \sqrt{2} * 5 - 16 \sqrt{2} = -15$

<u>Control Prefix</u>	<u>Control</u>	<u>Default Property</u>
- chk	Check Box	Value
- cbo	Combo Box	Text
- cmd	Command Button	Value
- fra	Frame	Caption
- hsb	Horizontal Scroll Bar	Value
- img	Image	Picture
- lbl	Label	Caption
- lin	Line	Visible
- lst	List Box	Text
- opt	Option Button	Value
- pic	Picture Box	Picture
- shp	Shape	Shape
- tmr	Timer	Enabled
- Vsb	Vertical Scroll Bar	Value