VISUAL BASIC UNIT III

TOPIC OUTLINE

VISUAL BASIC INTERGRATED DEVELOPMENT ENVIRONMENT

1. Integrated Development Environment (IDE)
2. Purpose of the Visual Basic IDE
3. Main components explanations
4. Features of the Visual Basic IDE
5. Advantages of Visual Basic IDE
6. Disadvantages of Visual Basic IDE
7. Importance of Visual Basic IDE in Learning Programming

VISUAL BASIC INTERGRATED DEVELOPMENT ENVIRONMENT

An **Integrated Development Environment (IDE)** is a software application that provides programmers with a complete set of tools to develop, test, debug, and deploy programs in one place.

In **Visual Basic**, the IDE is mainly provided through **Microsoft Visual Studio**, which integrates:

a) Code writing
b) Form design
c) Debugging (**Debugging** is the process of **finding, identifying, and correcting errors (bugs)** in a computer program so that it runs correctly)
d) Compilation
e) Execution

The IDE removes the need to use separate tools, making software development faster, easier, and more organized.

Purpose of the Visual Basic IDE

The Visual Basic IDE is designed to:

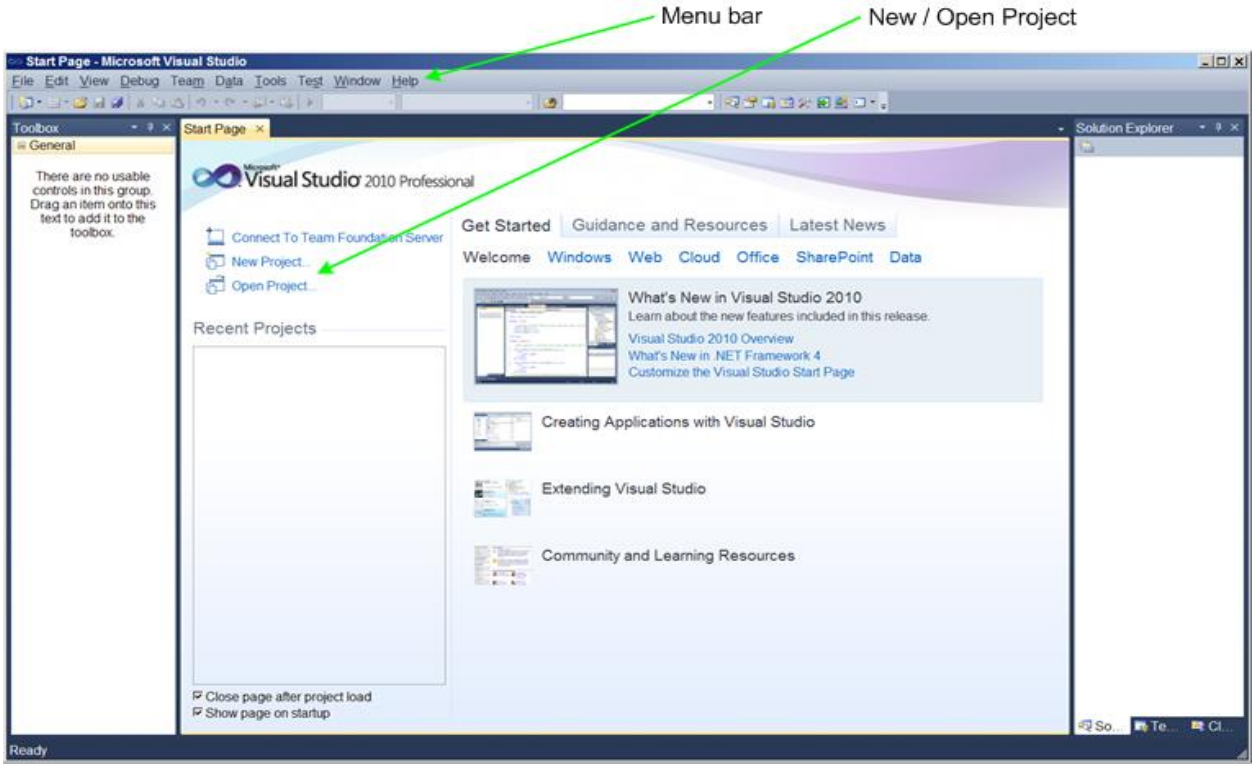
a) Simplify application development
b) Support **event-driven programming**
c) Allow **rapid application development (RAD)**
d) Reduce programming errors
e) Provide a visual way of designing user interfaces

Visual Basic IDE

The steps to access the visual basic IDE are as follows

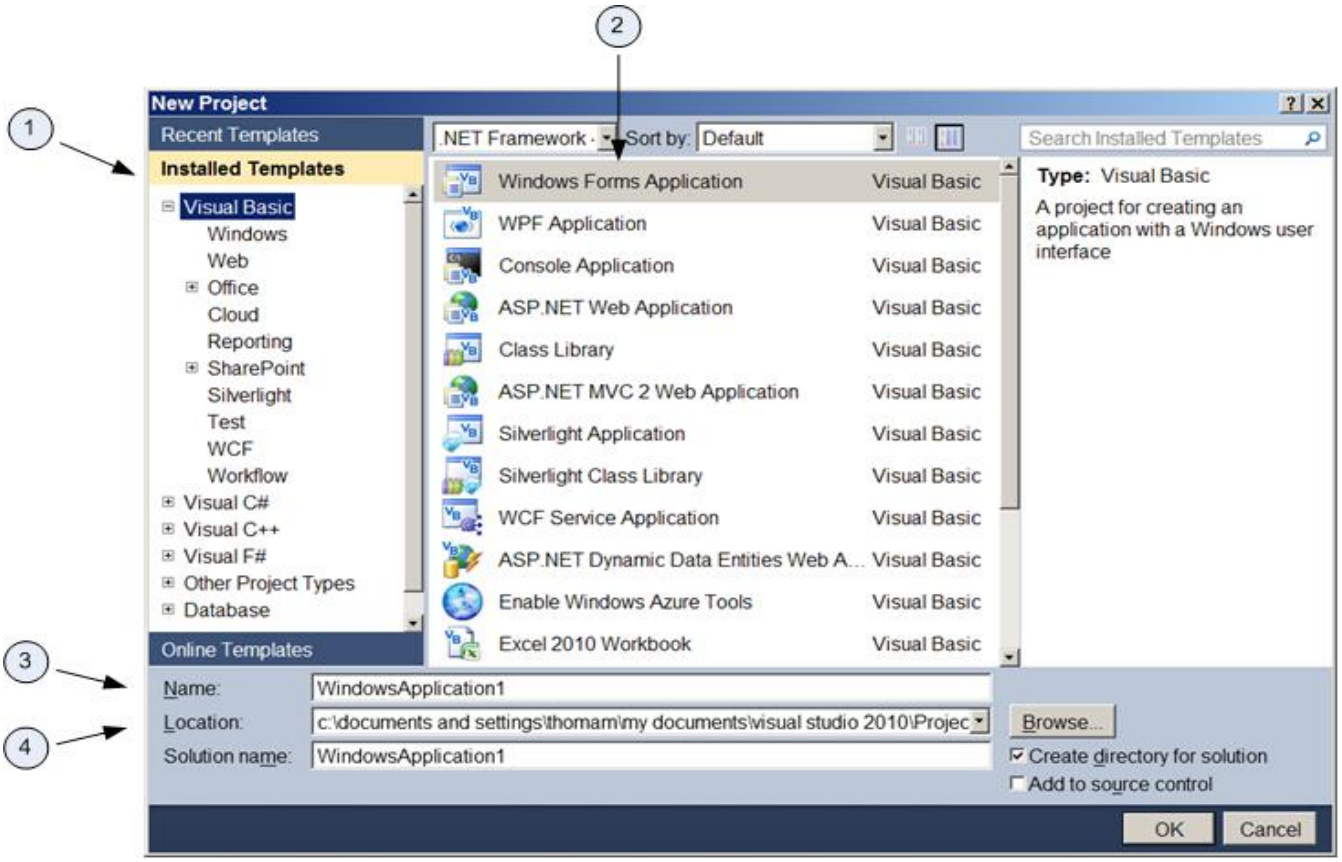
a) double click the Visual Studio.Net 2010 desktop icon, **or**
b) click Start → Programs → Microsoft Visual Studio 2010 → Microsoft Visual Studio 2010 from the Start menu.

Upon completing either of these, Visual Studio.Net 2010 should launch, and you should see something like:



Take a minute and examine this screen capture. This is the Visual Studio Start Page (notice the Start Page tab), however, we are not into the Visual Basic environment yet. Notice the Menu Bar as well as the New Project and Open Project selectors pointed to by the green arrows. Either of these will enable you to determine which .Net project type to open.
From the Menu Bar, select the Tools menu, Import and Export Settings, Reset All Settings, No, then select General Development Settings. This should help your User Interface (UI) more closely resembling the screen Images in this section.

Next, on the Menu Bar, select File → New → Project menu or select New Project from the Start page and you should see a window something like:

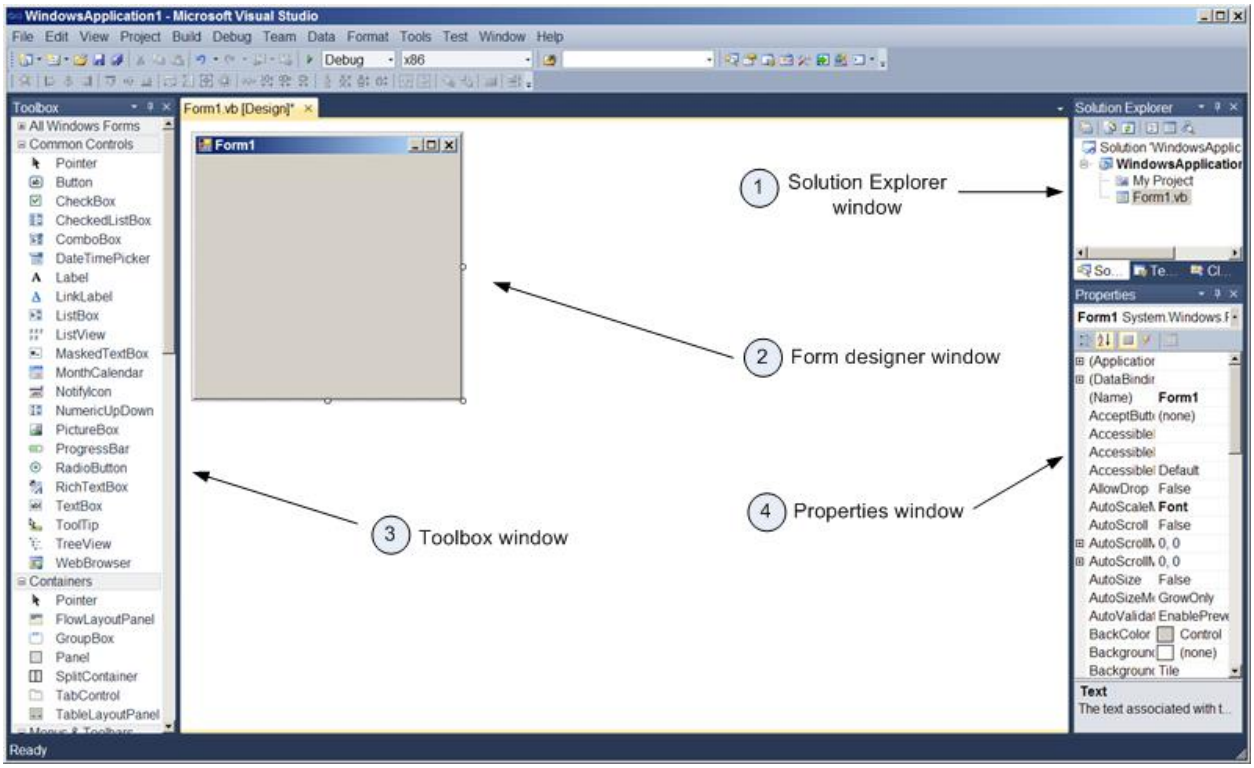


This is the New Project form (see the title bar). This is where selections for the type of New Project will be made. There are several important selections to be made on this form:

a) **Project Templates** - this is where the .Net language of choice is selected. For us, this should always be Visual Basic and Windows.
b) **Project Applications** - this is where we can select from the standard set of predefined applications. For us, this should always be a Console Application or a Windows Forms Application (what's the difference?).
c) **Name** - the name to be given to the VB.Net application. The name should always be indicative of what the application does, e.g. TaxCalculator (not WindowsApplication1).
d) **Location** - the file system location where the application folder and files will reside

Once you click the OK button in the New Project form, you will launch the Visual Basic.Net Integrated Development Environment (henceforth referred to as the IDE) window. It is in this

window that most, if not all development will be coordinated and performed. The startup IDE will look something like the following:



Main components explanations

1. Title Bar

Located at the top of the IDE window Displays:

a) Name of the project
b) Name of the current file
c) Name of the IDE (Visual Studio)

Example:

StudentSystem – Microsoft Visual Studio

2. Menu Bar

Located below the Title Bar.Contains drop-down menus for commands

Common Menus:

a) **File** – New, Open, Save, Close
b) **Edit** – Cut, Copy, Paste, Undo
c) **View** – Toolbox, Solution Explorer
d) **Project** – Add forms, modules, references
e) **Build** – Compile and build the project
f) **Debug** – Debugging tools
g) **Run** – Start or stop the program
h) **Help** – Documentation and support

3. Toolbar

Contains **shortcut buttons** for commonly used commands

Examples:

a) Save
b) Run (▶ Start)
c) Stop (■)
d) Undo / Redo
e) Build Solution

Advantage: Saves time compared to using menus.

4.Solution Explorer

Displays the structure of the entire project Shows:

a) Forms
b) Modules
c) Classes
d) References
e) Resources

Uses:

a) Navigate through project files
b) Add or remove files
c) Manage multiple projects in one solution

5. Form Designer (Design Window)

Used to design the graphical user interface (GUI). Allows drag-and-drop of controls

Common Controls:

a) Button
b) Label
c) TextBox
d) ComboBox
e) ListBox
f) PictureBox

Features:

a) Visual layout design
b) Resize and move controls
c) Set control properties visually

6. **Toolbox**

Contains all controls used in Visual Basic. Controls are grouped logically

Examples of Controls:

a) Common Controls
b) Containers
c) Menus & Toolbars
d) Data Controls

Usage:

Drag a control from the Toolbox to the Form Designer

7. Properties Window

Used to set and modify properties of forms and controls. Examples of Properties:

a) Name
b) Text
c) Font
d) Size
e) Back Color
f) Enabled
g) Visible

Importance:

Controls behavior and appearance without writing code

8. Code Editor (Code Window)

Where Visual Basic code is written .Supports:

a) Syntax highlighting
b) IntelliSense (auto-completion)
c) Error detection

Example:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    MessageBox.Show("Hello World")
End Sub
```

9. Object Browser

Displays **classes, methods, properties, and events.** Helps understand available objects and libraries Benefits of object browser is that **,** learn VB libraries and avoid memorizing syntax

10. Error List Window

Displays: Syntax errors, Warnings and Messages. Error list window helps quickly identify and fix errors

11. Debugging Tools

The IDE provides powerful debugging tools:

a) Breakpoints

- Pause execution at a specific line

b) Watch Window

- Monitor variable values during execution

c) Immediate Window

- Execute code while debugging

d) Step Controls

- Step Into
- Step Over
- Step Out

Purpose: Find and fix logical and runtime errors.

12. Output Window

Displays: Build results, Debug messages and Compilation status

Features of the Visual Basic IDE

Event-driven programming support

a) Drag-and-drop interface design
b) Automatic code generation
c) Integrated compiler
d) Integrated debugger
e) Project management
f) Database connectivity tools

Advantages of Visual Basic IDE

a) Easy to learn for beginners
b) Faster application development
c) Reduced syntax errors
d) Visual design capability
e) Good debugging support

Disadvantages of Visual Basic IDE

a) Limited low-level hardware control
b) Slower performance compared to C++
c) Windows-dependent
d) Less suitable for system-level programming

7. Importance of Visual Basic IDE in Learning Programming

a) Helps beginners understand programming concepts
b) Encourages logical thinking
c) Ideal for building:

- Desktop applications
- Educational systems

- Business applications