Chapter 1.Visual Stdio.Net IDE

A reference of MSDN Library for Visual Studio 2017

IT Faculty, TDM University

Contents

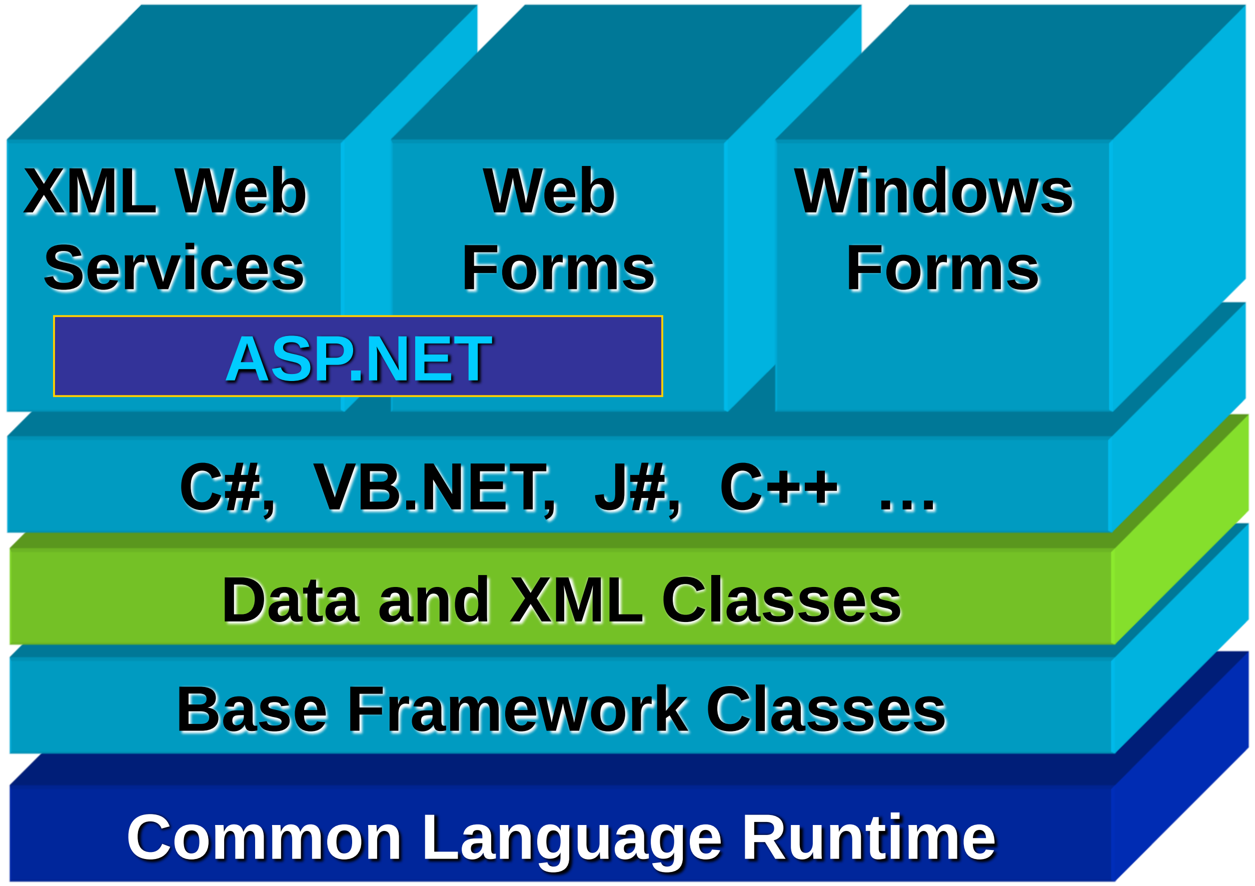
Introduction

- Visual Studio .NET
  - Microsoft's Integrated Development Environment (IDE)
  - Program in a variety of .NET languages
  - Tools to edit and manipulate several file types
- .NET initiative
  - Introduced by Microsoft (June 2000)

Introduction

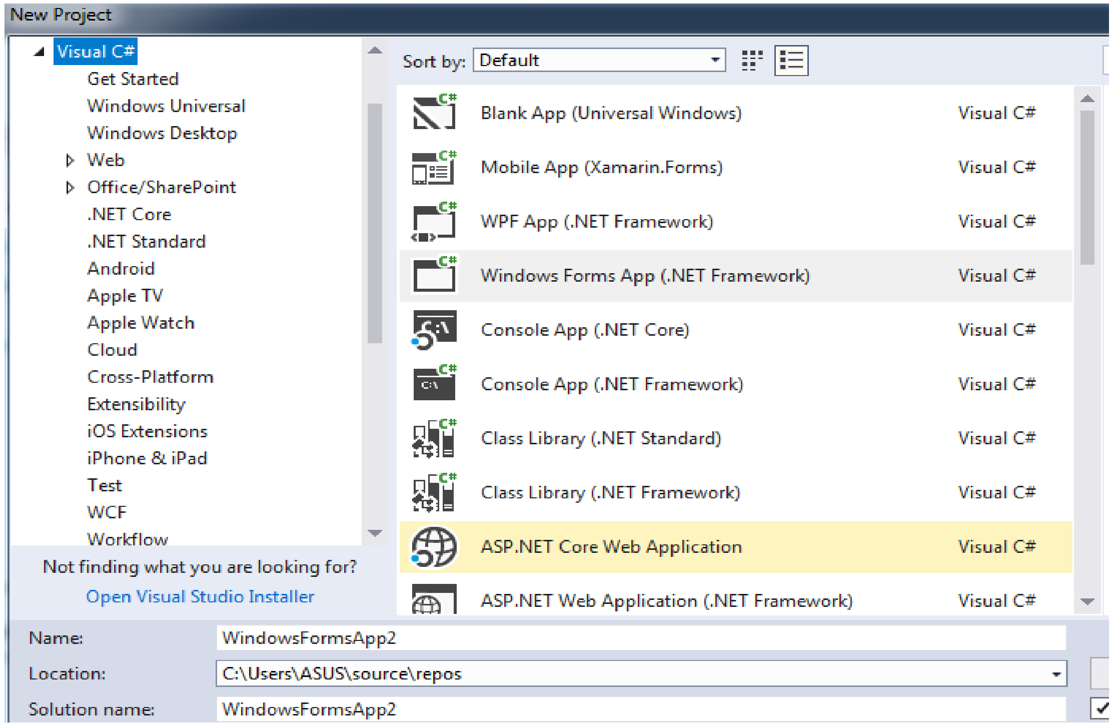
- C#
  - Developed at Microsoft by a team led by Anders Hejlsberg and Scott Wiltamuth
  - Event driven, object oriented, visual programming language
  - Based from C, C++ and Java

Microsoft.NET Framework



Visual Studio .NET IDE

- Create a new project
  - **`File -> New -> Project`** or click Create new project



New Project

Visual C#
- Get Started
- Windows Universal
- Windows Desktop
- Web
- Office/SharePoint
- .NET Core
- .NET Standard
- Android
- Apple TV
- Apple Watch
- Cloud
- Cross-Platform
- Extensibility
- iOS Extensions
- iPhone & iPad
- Test
- WCF
- Workflow

Not finding what you are looking for?
Open Visual Studio Installer

Sort by: Default

Blank App (Universal Windows)	Visual C#
Mobile App (Xamarin.Forms)	Visual C#
WPF App (.NET Framework)	Visual C#
Windows Forms App (.NET Framework)	Visual C#
Console App (.NET Core)	Visual C#
Console App (.NET Framework)	Visual C#
Class Library (.NET Standard)	Visual C#
Class Library (.NET Framework)	Visual C#
ASP.NET Core Web Application	Visual C#
ASP.NET Web Application (.NET Framework)	Visual C#

Name: WindowsFormsApp2
Location: C:\Users\ASUS\source\repos
Solution name: WindowsFormsApp2

Visual Studio .NET IDE

- C# .NET project
  - Group of related files, images, and documentations
- C# .NET solution
  - Group of projects creating one or a group of applications

Visual Studio .NET IDE

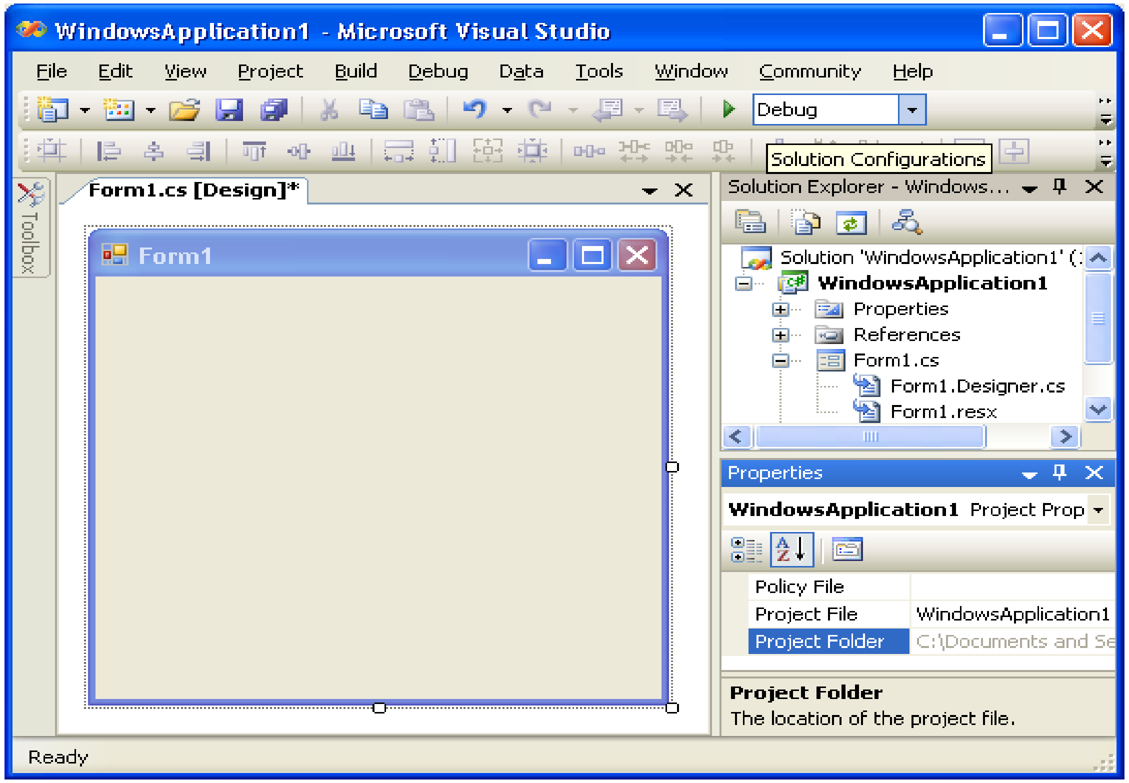
- Console applications
  - No visual components
  - Only text output
  - Two types
    - MS-DOS prompt
      - Used in Windows 95/98/ME
    - Command prompt
      - Used in windows 2000/NT/XP

Visual Studio .NET IDE

- Windows applications
  - Anything that runs in the Windows OS
  - Forms with several output types
  - Contain Graphical User Interfaces (GUIs)

Visual Studio .NET IDE

- IDE after a new project


WindowsApplication1 - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Window Community Help
Debug
Solution Configurations
Toolbox
Form1.cs [Design]*
Form1
Solution Explorer - Windows...
Solution 'WindowsApplication1' (
WindowsApplication1
Properties
References
Form1.cs
Form1.Designer.cs
Form1.resx
Properties
WindowsApplication1 Project Prop
Ready

Policy File	
Project File	WindowsApplication1
Project Folder	C:\Documents and Se

Project Folder
The location of the project file.

Visual Studio .NET IDE

- Form
  - Grey rectangle in window
  - Represents the project's window
  - Part of the GUI or Graphical User Interface
    - Graphical components for user interaction
    - User can enter data (input)
    - Shows user instructions or results (output)
- Tabs
  - One tab appears for each open document
  - Used to save space in the IDE

Menu Bar and Toolbar

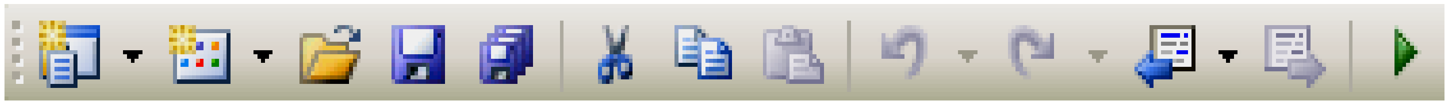
- Menu bar
  - Commands for developing and executing programs
    - Create new projects by going to **`File`** > **`New`** > **`Project`**
  - Certain menu options only appear in specific IDE modes
  - Each menu is summarized in following Figure:

Menu Bar and Toolbar

Menu	Description
File	Contains commands for opening projects, closing projects, printing projects, etc.
Edit	Contains commands such as cut, paste, find, undo, etc.
View	Contains commands for displaying IDE windows and toolbars.
Project	Contains commands for adding features, such as forms, to the project.
Build	Contains commands for compiling a program.
Debug	Contains commands for debugging and executing a program.
Data	Contains commands for interacting with databases.
Tools	Contains commands for additional IDE tools and options for customizing the environment.
Windows	Contains commands for arranging and displaying windows.
Help	Contains commands for getting help.

Menu Bar and Toolbar

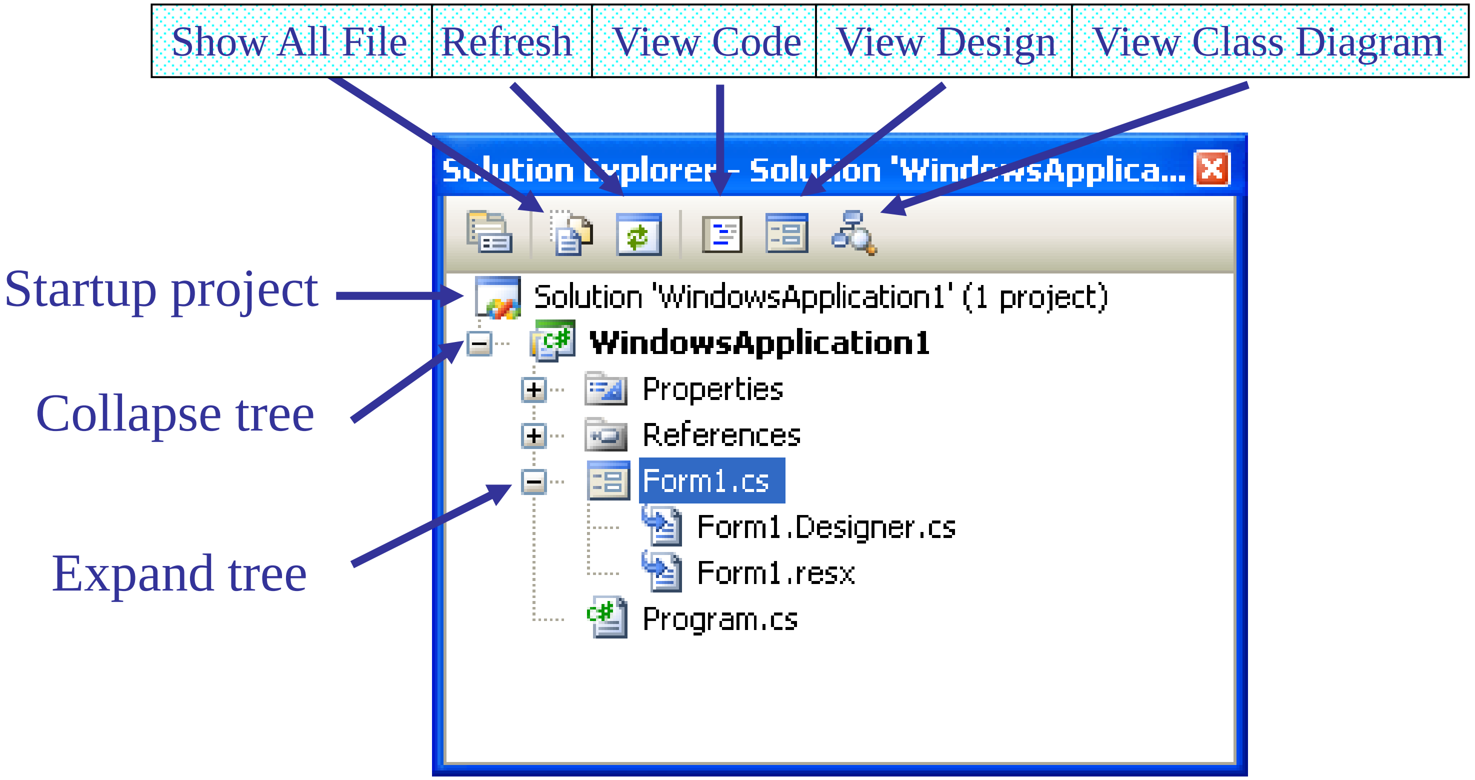
- Toolbar



  - Contains commonly used commands as icons
  - Used rather than navigating through menus
  - Simply click the icon to use the command
    - Some icons have down arrows that offer additional commands
    - Holding the mouse over an icon displays a tool tip
      - Tool tips briefly state what the icons are or do.

Visual Studio .NET Windows

- The `Solution Explorer`



Visual Studio .NET Windows

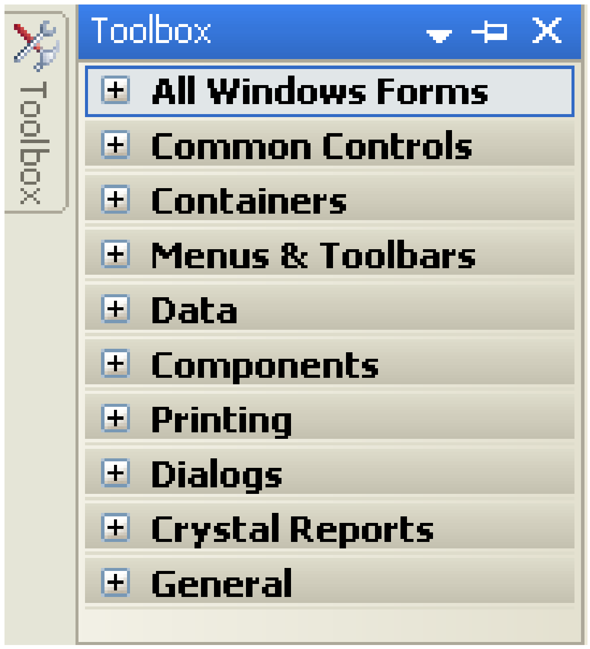
- The **`Solution Explorer`**
  - Lists all files in the solution
  - Displays the contents or a new project or open file
  - The start up project is the project that runs when the program is executed
    - It appears in bold in the **`Solution Explorer`**
  - The plus and minus images expand and collapse the tree
    - Can also double click on the file name to expand/collapse

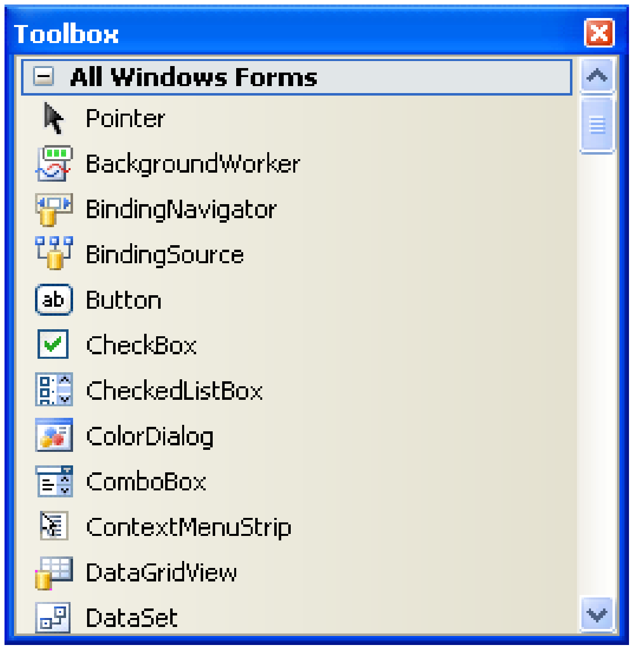
Visual Studio .NET Windows

- The **Solution Explorer**
  - **Solution Explorer** toolbar
    - The **Show All files** icon Shows all files
    - The **Refresh** icon reloads files in the solution
    - The **View Code** icon shows code of selected object
    - The **View Design** icon shows design of selected object
    - Icons change based on selected file

Visual Studio .NET Windows

- The `Toolbox`


Toolbox
- All Windows Forms
- Common Controls
- Containers
- Menus & Toolbars
- Data
- Components
- Printing
- Dialogs
- Crystal Reports
- General


Toolbox
- All Windows Forms
  - Pointer
  - BackgroundWorker
  - BindingNavigator
  - BindingSource
  - Button
  - CheckBox
  - CheckedListBox
  - ColorDialog
  - ComboBox
  - ContextMenuStrip
  - DataGridView
  - DataSet

Visual Studio .NET Windows

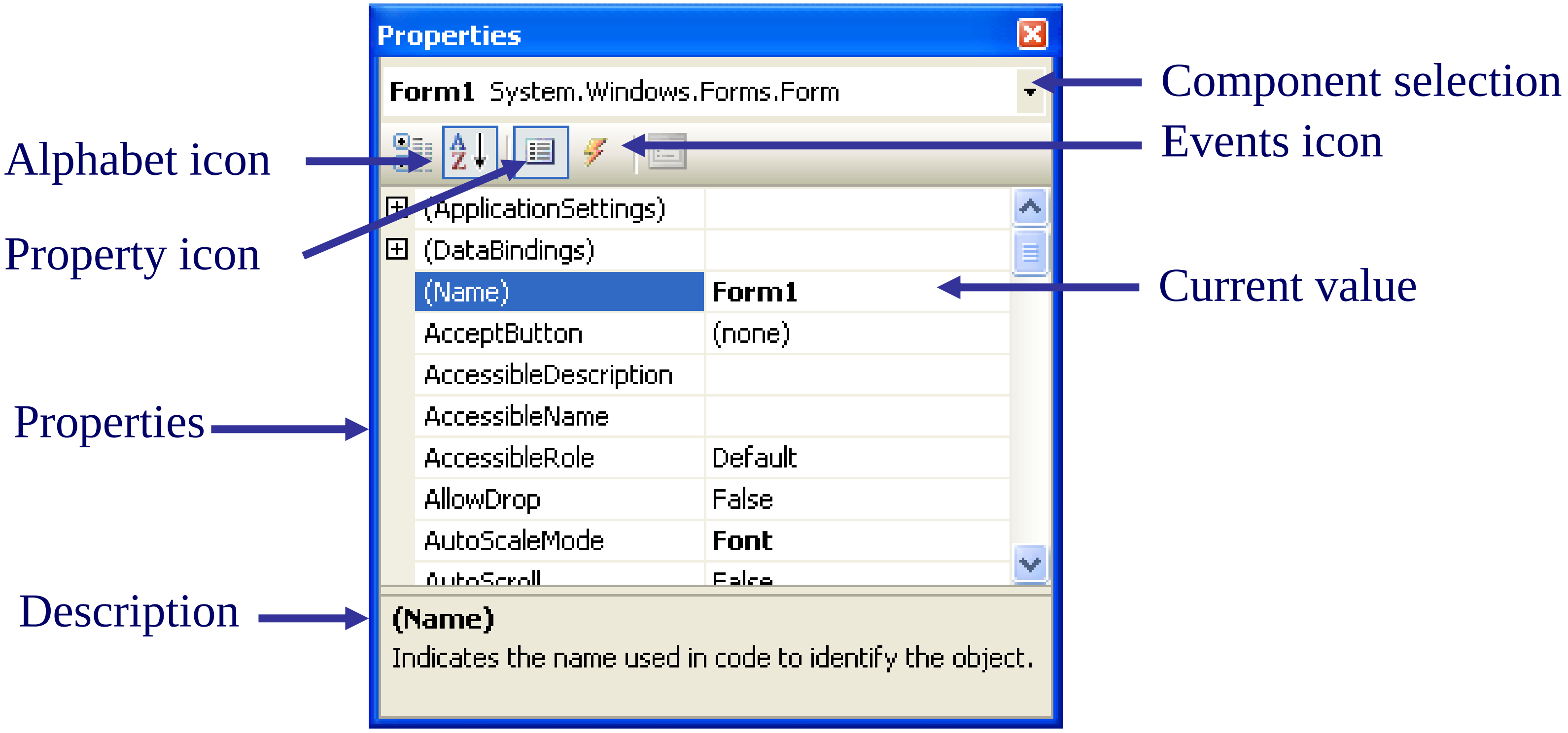
- The `Toolbox`
  - Contains reusable controls
    - Controls customize the form
    - Visual programming allows 'drag and drop' of controls
  - Black arrows at bottom are used to scroll through items
  - Mouse pointer icon allows user to deselect current control
  - No tool tips
    - Each icon is labeled with a its name

Visual Studio .NET Windows

- The **`Toolbox`**
  - **`Toolbox`** can be hidden on left side of IDE
    - Mouse over it to expand it
    - When the mouse is no longer over it, the toolbar goes away
    - The pin icon is used disable auto hide

Visual Studio .NET Windows

- The **`Properties`** window



Properties

Form1 System.Windows.Forms.Form

⊞ (ApplicationSettings)	
⊞ (DataBindings)	
(Name)	**Form1**
AcceptButton	(none)
AccessibleDescription	
AccessibleName	
AccessibleRole	Default
AllowDrop	False
AutoScaleMode	**Font**
AutoScroll	False

(Name)
Indicates the name used in code to identify the object.

Visual Studio .NET Windows

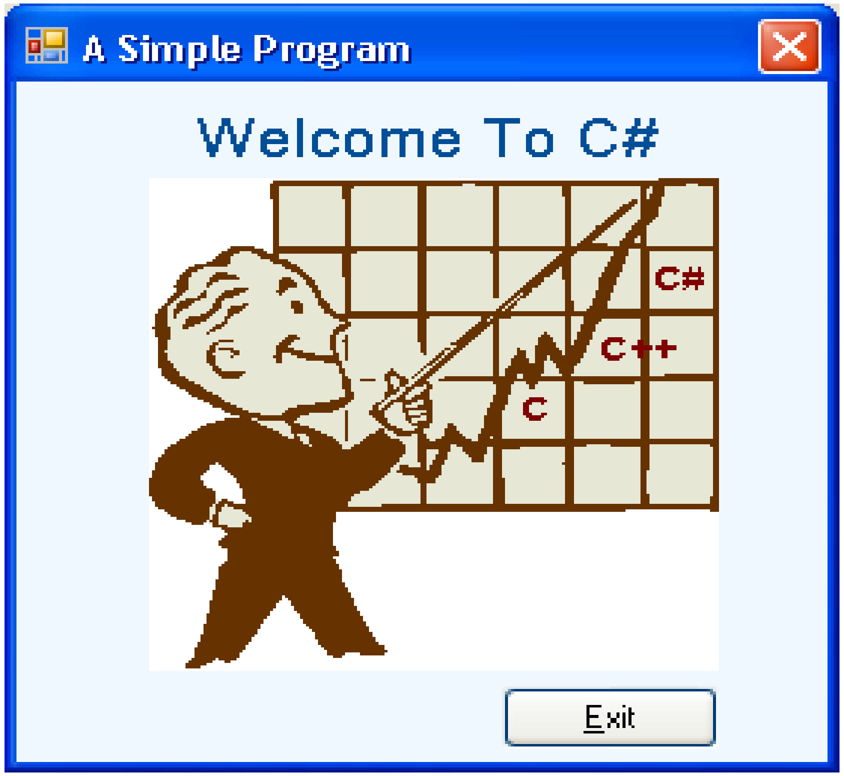
- The `Properties` window
  - Manipulate the properties of a form or control
  - Each control has its own set of properties
    - Properties can include size, color, text, or position
  - Right column is the property
  - Left column is the property value

Visual Studio .NET Windows

- The **`Properties`** window
  - Icons
    - The **`Alphabetic`** icon arranges the properties alphabetically
    - The **`Property`** icon shows the properties of control
    - The **`Event`** icon allows reactions to user actions
  - Users alter controls visually without writing code
  - The component selection dropdown list shows what control is being altered and what other controls can be altered

Simple Program

- Design A Simple Program

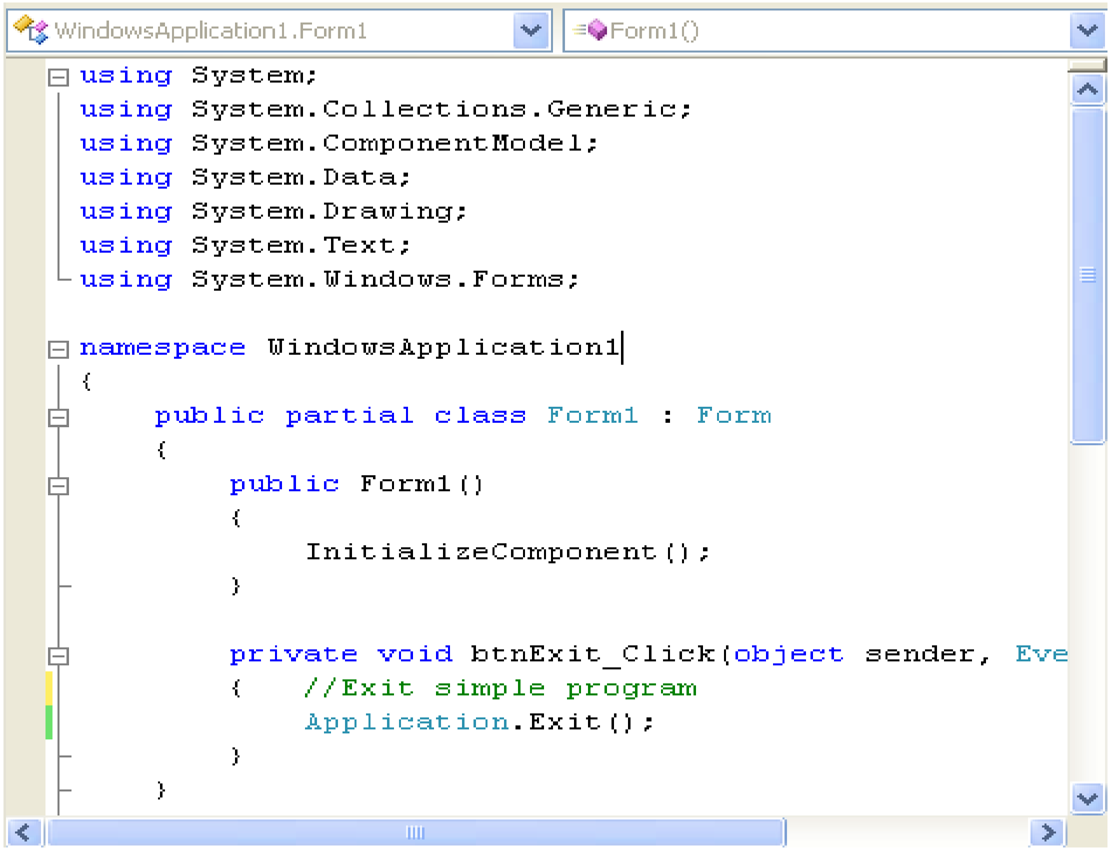


Simple Program

- Save the project
  - In the **Solution Explorer** select **File > Save**
  - Using **Save All** will save the source code and the project
- Run the project
  - In run mode several IDE features are disabled
  - Click **Build Solution** in the **Build** menu to compile the solution
  - Click **Debug** in the **Start** menu or press the ***F5*** key

Simple program

- View code



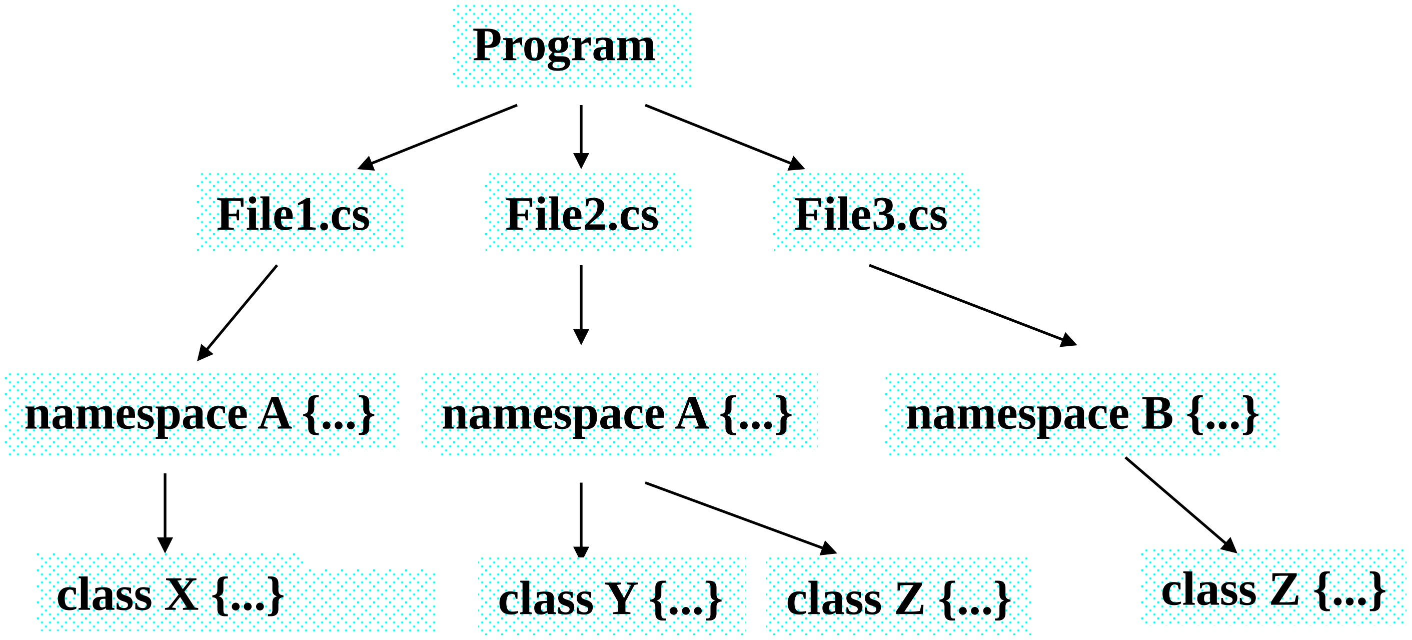
```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace WindowsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void btnExit_Click(object sender, Eve
        {   //Exit simple program
            Application.Exit();
        }
    }
```

Structure of C# Programs

- Structure of C# Programs



Structure of C# Programs

- Namespaces
  - Groups related C# features into a categories
  - Allows the easy reuse of code
  - Many namespaces are found in the .NET framework library
  - Must be referenced in order to be used
  - Example:
    - using System.Text;
    - using System.Windows.Forms;
    - namespace WindowsApplication1

Structure of C# Programs

- Namespaces in the Framework Class Library
  - **`System`**: Contains essential classes and data types (such as **`int`**, **`double`**, **`char`**, etc.). Implicitly referenced by all C# programs.
  - **`System.Data`**: Contains classes that form ADO .NET, used for database access and manipulation.
  - **`System.Drawing`**: Contains classes used for drawing and graphics.
  - **`System.IO`**: Contains classes for the input and output of data, such as with files.

Structure of C# Programs

- Namespaces in the Framework Class Library
  - `System.Windows.Forms`: Contains classes used to create graphical user interfaces.
  - `System.Xml`: Contains classes used to process XML data.
- Keywords
  - Words that cannot be used as variable or class names or any other capacity.
  - Example: **class**
  - All keywords are lowercase.

Structure of Class

- Struture of the class

```
class <classname> {

    ... fields, constants ...
    ... methods ...
    ... constructors, destructors ...

    ... properties ...
    ... events ...

    ... indexers ...
    ... overloaded operators ...
    ... nested types (classes, structs, enums,)...
}
```

Structure of Class

- Class names can only be one word long (i.e. no white space in class name)
- Each class name is an identifier
  - Can contain letters, digits, and underscores (_)
  - Cannot start with digits
  - Can start with the at symbol (@)

Structure of Class

- Example of the class

```
class rectangle{
    private float a, b;                          //fields
    public rectangle(float x-0, float y=0){ //Constructor
        a = x; b = y;
    }
    public void init(float x, float y){     //Method
        a = x; b = y;
    }
    public float area(){                         //Method
        returb a*b;
    }
}
```

Object

- Object classes encapsulate (wrap together) data and methods.
- Objects can hide implementation from other objects (information hiding)
- Methods: units of programming.
- User-defined type: class written by a programmer.

Object

- Member Access Modifiers
  - **`public`**: Member is accessible wherever an instance of the object exists.
  - **`private:`** Members is accessible only inside the class definition
- Object must be created with **new**
  - Example: rectangle r = new rectangle();

Structure of Class

- Methods
  - Building blocks of programs
  - Method Overloading
    - if they have different numbers of parameters, or
    - if they have different parameter types, or
    - if they have different parameter kinds (value, ref/out)
  - The **`Main`** method
    - Each console or windows application must have exactly one
    - All programs start by executing the **`Main`** method

Structure of Class

- Constructors for Classes
  - Initializes objects of the class
  - Can take arguments
  - Cannot return values
  - There may be more then one constructor per class (overloaded constructors)
  - A construcor may call another constructor with *this*.

Structure of Class

- Default Constructor
  - If no constructor was declared in a class, the compiler generates a parameterless default constructor
  - If a constructor was declared, no default constructor is generated.

Structure of Class

- Destructors
  - Called for an object before it is removed by the garbage collector.
  - Base class destructor is called automatically at the end.
  - No **public** or **private**