LESSON 5

98-361 Software Development Fundamentals

LESSON 5.1

98-361 Software Development Fundamentals

Understand Windows Forms Applications and Console-based Applications

Lesson Overview

Students will understand Windows Forms applications and console-based applications.

In this lesson, you will learn about:

- The Windows Forms event model
- Visual inheritance
- The use of Multiple Document Interface (MDI) and Single Document Interface (SDI) applications
- Characteristics and capabilities of console-based applications

Review Terms

- Event–an action or occurrence, often generated by the user, to which a program might respond. Examples: key presses, button clicks, and mouse movements.
- User Interface–the portion of a program with which a user interacts.
  - Types of user interfaces (UIs) include command-line interfaces, menu-driven interfaces, and graphical user interfaces.
- Windows Forms–a rich Windows client library for building Windows client applications.

Windows Forms Applications

- Programs that run on a user’s Windows-based computer.
- Contain controls to create a UI and code to manipulate data.
- Coded in the Microsoft Visual Studio Integrated Development Environment (IDE).
- Windows Forms refers to the set of managed libraries used to simplify common application tasks.

Windows Forms Applications (continued)

- Can display information, request input from users, and communicate with computers over a network.
- A form is a visual surface on which you display information to the user.
- Controls are added and responses to user actions, such as mouse clicks or key presses, are coded.
- A control is a UI element that displays data or accepts data input.

Why use a Windows Form?

- The interface corresponds to the operating system, so the application is integrated with the desktop.
- The UI is consistent.
- It has a higher processing demand than a Web application.
- Security and reliability is important.

Windows Forms Event Model

- When a user does something to your form or one of its controls, the action generates an event.
- An event is an action that you can respond to in code.
- The application uses code to react to the event and process the event when it occurs.
- The event handler is a procedure in the code that determines what actions are performed when an event occurs.
  - When the event is raised, the event handler(s) that receive the event are executed.

Visual Inheritance

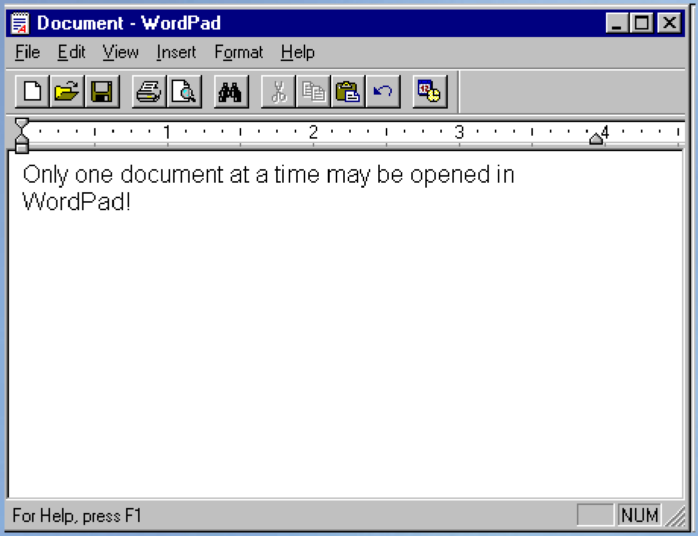
- Why use visual inheritance?
  - A project requires a form similar to one used in a previous project.
  - A basic form or control layout will be used as a template and will be modified for different situations later on.
- Form inheritance enables you to create a base form and then inherit from it and make modifications, while preserving whatever original settings you need.

Visual Inheritance: Examples

```
//Visual Basic
Public Class Form2
   Inherits Namespace1.Form1
```

Single Document Interface (SDI)

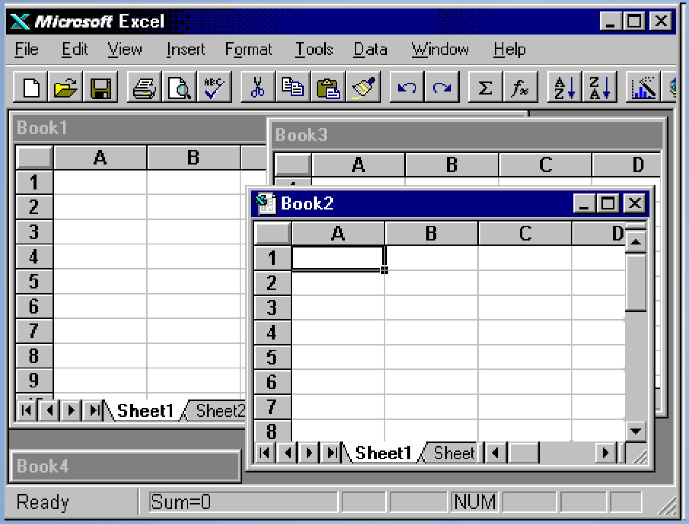
- Each document frame window is separate from others.
- Each window contains its own menu and toolbar and does not have a background or parent window.



Only one document at a time may be opened in WordPad!

Multiple Document Interface (MDI)

- Multiple document frame windows may be open in the same instance of an application.
- An MDI application has a window within which multiple MDI child windows, which are frame windows themselves, can be opened, each containing a separate document.



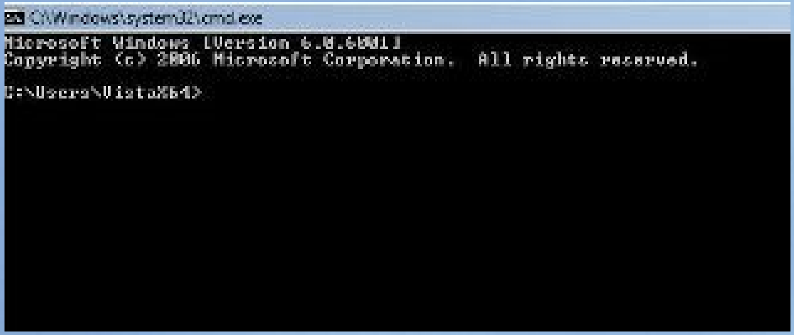
A version of Microsoft Excel using an MDI

Choosing between SDI and MDI

- Why use SDI?
  - When it is unlikely that more than one window is needed.
  - Example: When using a calendar application, it is unlikely that you would need more than one calendar open at a time.
- Why use MDI?
  - When more than one window is needed.
  - Example: When processing insurance claims, an employee often needs to work on more than one claim at a time and has to compare claims side by side.

Console-based Applications

- A program that uses a text-only interface and usually requires only a keyboard for input.
- They typically run in a command window, such as the Win32 console.
- Images and video cannot be displayed.
- Useful on older computers that may be slow to render image content.
- Why use a console application?
  - Speed of deployment
  - Ease of use



```
Microsoft Windows [Version 6.0.6001]
Copyright (c) 2006 Microsoft Corporation.  All rights reserved.

C:\Users\VistaX64>
```

LESSON 5.1

98-361 Software Development Fundamentals

No Student Lab 5.1

Microsoft

LESSON 5.2

98-361 Software Development Fundamentals

Understand Windows Services

Lesson Overview

Students will understand Windows Services.

In this lesson, you will learn about:

- Characteristics and capabilities of Windows Services

Review Terms

- Service—a program or routine that provides support to other programs

Windows Service

- An executable that carries out specific functions and is designed not to require user involvement.
- It often is configured to start alongside the operating system and run in the background.
- Why use a Windows service?
  - When you want a program to start automatically when the operating system starts
  - When your program does not require user interaction, and therefore may not need a user interface (UI)
  - When you need long-running functionality

How are Windows Services different?

- The compiled executable file that a service application project creates must be installed on the system before the project can function.
- You must create installation components for service applications.
  - The installation components install and register the service on the computer and create an entry for your service with the Windows Services Control Manager.
- Windows Service applications run in their own security context and are started before the user logs into the Windows computer on which he or she is installed.

Windows Service Lifetime

- The service is installed on the system and loaded into the Windows Services Control Manager.
- The service is started.
  - At this point, the service can be:
    - Running
    - Paused
    - Stopped
- You can pause, stop, or resume a service
  - By using the Windows Services Control Manager
  - By using Server Explorer
  - By calling methods in code

LESSON 5.2

98-361 Software Development Fundamentals

No Student Lab 5.2

Microsoft

LESSON 5

98-361 Software Development Fundamentals

Complete the QUIA Test

MTA Software Fundamentals 5 Test