COLLEGE OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF INFORMATION SYSTEMS

Event Driven Programming

with C#

Course Contents

Chapter

1. Event Driven Fundamentals
2. Programming with Event Driven
3. The Elements of Event Driven Programming
4. Advanced Programming with Event Driven
5. Database Programming

Chapter 3: The Elements of event driven programs

Contents of the Chapter

1. Creating Menus in Your Programs
2. Enhancing Your Programs with Forms and Dialog Boxes
3. Handling Keyboard and Mouse Input in Your Programs
4. Working with Time and Timers
5. Adding Graphics to Your Programs
6. Writing Reusable Code with Subs and Functions
7. Saving and Retrieving Your Data with programs
8. Deploying your event driven Applications

The Elements of event driven programs…….. Events

- An **event** is an **action** that you can respond to, or "handle," in code.
- Events can be generated
  - ✓ by a user action, such as clicking the mouse or pressing a key,
  - ✓ by program code, or
  - ✓ by the system.
- Event-driven applications execute code in response to an event.

The Elements of event driven programs…….. Events

- Each form and control exposes a predefined set of events that you can program against.
- If one of these events occurs and there's code an associated event handler, that code is invoked.
- If a user clicks a form, code in the form's Click event handler is executed.

The Elements of event driven programs…….. Window Forms

- **Windows Forms**
  - is a smart client technology for the .NET Framework,
  - a set of managed libraries that simplify common application tasks such as reading and writing to the file system.
- When you use a development environment like Visual Studio, you can create **Windows Forms applications** that
  - display information,
  - request input from users, and
  - communicate with remote computers over a network.

The Elements of event driven programs…….. Window Forms

- A **form** is a visual surface on which you *display information to the user*.
- You ordinarily build Windows Forms applications by adding controls to forms and developing responses to user actions, such as mouse clicks or key presses.
- A **control** is a discrete user interface (UI) element that displays data or accepts data input.
- When a user does something to your form or one of its controls, the action generates an event.
- Your application reacts to these events by using code, and processes the events when they occur.

The Elements of event driven programs…….. Window Forms

- Windows Forms contains a variety of controls that you can add to forms: controls that display **text boxes**, **buttons**, **drop-down boxes**, **radio buttons**, and **even Web pages**.
- If an existing control does not meet your needs, Windows Forms also supports creating your own custom controls using the **UserControl** class.

The Elements of event driven programs…….. Window Forms

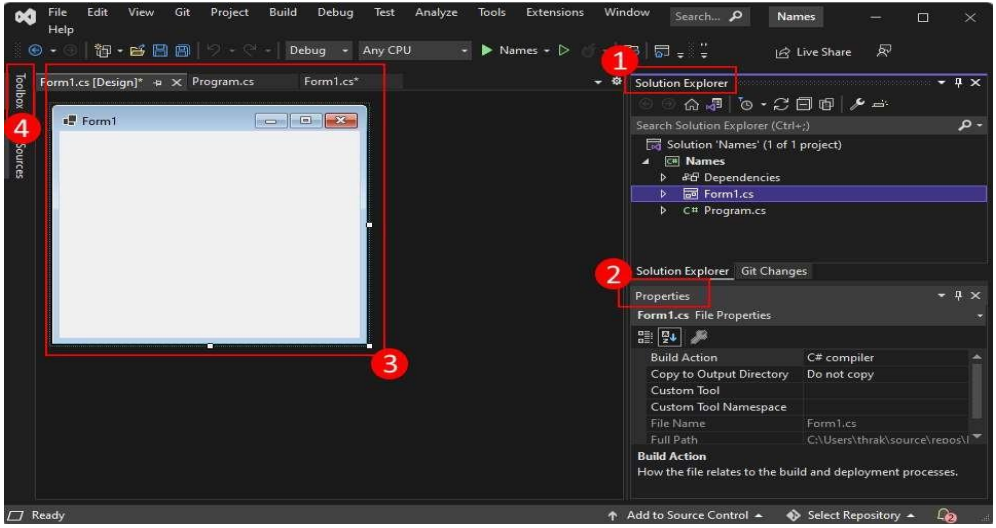
- With the drag-and-drop **Windows Forms Designer** in Visual Studio, you can easily create Windows Forms applications.
- Just **select** the controls with your cursor and **add** them **where you want on the form**.
- You can use the **FlowLayyoutPanel**, **TableLayyoutPanel** and **SplitContainer** controls to create advanced form layouts in less time.

The Elements of event driven programs…….. Creating a new Windows Forms

- Create Windows **Forms** applications,
- Add the **controls** you can use on them,
- Control **events** and handling events to provide App functionality and
- how to handle **input** from the user,

The Elements of event driven programs…….. Creating a new Windows Forms

- Important parts of Visual Studio to create A Windows Forms App.



The Elements of event driven programs…….. Creating a new Windows Forms

1. Solution Explorer

- All of your project files, code, forms, resources, will appear in this pane.

2. Properties

- This pane shows property settings you can configure based on the item selected.

3. Form Designer

- This is the designer for the form. It's interactive and you can drag-and-drop objects from the Toolbox.
- By selecting and moving items in the designer, you can visually compose the user interface (UI) for your app.

4. Toolbox

- The toolbox contains all of the controls you can add to a form.
- To add a control to the current form, double-click a control or drag-and-drop the control.

The Elements of event driven programs…….. Window Forms

- An **event** is an **action** which you can respond to, or "handle," in code.
- **Events** can be generated by a user action, by program, or by the system.
- An **event handler** is a procedure in your code that determines what actions are performed when an event occurs, such as when the user clicks a button or a message queue receives a message.
- An event handler is a method that is bound to an event.
- When an event is raised, the event handler or handlers that receive the event are executed.

The Elements of event driven programs…….. Window Forms

```
private void button1_Click(object sender, EventArgs e)
    {
    / /  Add event handler code here.
    }
```

- Each event handler provides two parameters that allow you to handle the event properly

The Elements of event driven programs…….. Controls

- Windows Forms controls are reusable components that encapsulate user interface functionality and are used in client-side, Windows-based applications.
- Not only does Windows Forms provide many ready-to-use controls, it also provides the infrastructure for developing your own controls.
- You can combine existing controls, extend existing controls, or author your own custom controls

The Elements of event driven programs……..

Position and layout of controls

- **Control placement** in Windows Forms is determined not only by the control, but also by the parent of the control.
- The **position** a control appears on a parent is determined by the value of the **Location property** relative to the top-left of the parent surface.
- The top-left position coordinate in the parent is (x0,y0) .
- The size of the control is determined by the Size property and represents the width and height of the control

The Elements of event driven programs……..

Position and layout of controls

- When a control is added to a parent that enforces automatic placement, the position and size of the control is changed.
- In this case, the position and size of the control may not be manually adjusted, depending on the type of parent.
- The MaximumSize and MinimumSize properties help set the minimum and maximum space a control can use.

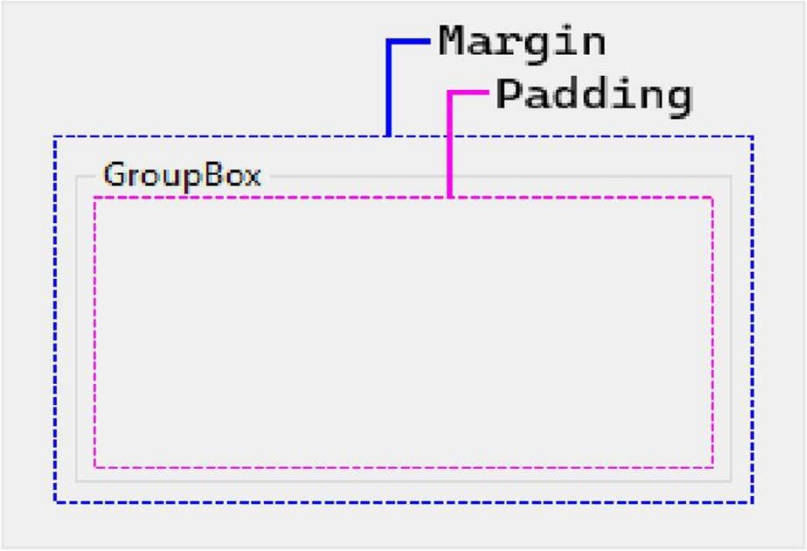
The Elements of event driven programs……..

Position and layout of controls

- There are two control properties that help with precise placement of controls: **Margin** and **Padding**.
- The **Margin property** defines the space around the control that keeps other controls a specified distance from the control's borders.
- The **Padding property** defines the space in the interior of a control that keeps the control's content (for example, the value of its Text property) a specified distance from the control's borders.
- The following figure shows the Margin and Padding properties on a control.

The Elements of event driven programs……...

Position and layout of controls



The Elements of event driven programs……..

Automatic placement and size

- Controls can be automatically placed within their parent.
  - Some parent containers force placement while others respect control settings that guide placement.
- There are two properties on a control that help automatic placement and size within a parent: **Dock** and **Anchor**.
- Drawing order can affect automatic placement.
  - The order in which a control is drawn determined by the control's index in the parent's Controls collection.
- This index is known as the Z-order. Each control is drawn in the reverse order they appear in the collection.
  - Meaning, the collection is a first-in-last-drawn and last-in-first-drawn collection.

The Elements of event driven programs…….. Docker

- The Dock property sets which border of the control is **aligned** to the corresponding side of the parent, and how the control is **resized** within the parent.
- When a control is docked, the container determines the **space** it should occupy and **resizes** and **places** the control.
- The width and height of the control are still respected based on the docking style.
  - For example, if the control is docked to the top, the Height of the control is respected but the Width is automatically adjusted.
  - If a control is docked to the left, the Width of the control is respected but the Height is automatically adjusted.
- The Location of the control can't be manually set as docking a control automatically controls its position.
- The Z-order of the control does affect docking. As docked controls are laid out, they use what space is available to them.
  - For example, if a control is drawn first and docked to the top, it will take up the entire width of the container.
    - If a next control is docked to the left, it has less vertical space available to it.

The Elements of event driven programs…….. Anchor

- Anchoring a control allows you to tie the control to one or more sides of the parent container. As the container changes in size, any child control will maintain its distance to the anchored side.
- A control can be anchored to one or more sides, without restriction. The anchor is set with the Anchor property.

The Elements of event driven programs…….. Container

FORM

- The **Form** is the main object of Windows Forms.
- **A Windows Forms application will usually have a form displayed at all times.**
- Forms contain controls and respect the Location and Size properties of the control for manual placement.
- Forms also respond to the **Dock** property for automatic placement.
- Most of the time a form will have grips on the edges that allow the user to resize the form.
- The **Anchor** property of a control will let the control grow and shrink as the form is resized.

The Elements of event driven programs…….. Container

PANEL

- The Panel control is similar to a form in that it simply groups controls together.
- It supports the same manual and automatic placement styles that a form does.
- A panel blends in seamlessly with the parent, and it **does cut off any area of a control that falls out of bounds of the panel.**
- If a control falls outside the bounds of the panel and AutoScroll is set to true , scroll bars appear and the user can scroll the panel.
- Unlike the group box control, a panel doesn't have a caption and border.

The Elements of event driven programs…….. Container

GROUPBOX

- The GroupBox control provides an **identifiable grouping** for other controls.
- Typically, you use a group box to subdivide a form by function.
  - For example, you may have a form representing personal information and the fields related to an address would be grouped together.
- At design time, it's easy to move the group box around along with its contained controls.
- The group box supports the same manual and automatic placement styles that a form does.
- A group box also cuts off any portion of a control that falls out of bounds of the panel.
- Unlike the panel control, a group box doesn't have the capability to scroll content and display scroll bars.

The Elements of event driven programs…….. Container

FLOWLAYOUT PANEL

- The FlowLayout Panel control arranges its contents in a horizontal or vertical flow direction.
- You can wrap the control's contents from one row to the next, or from one column to the next. Alternately, you can clip instead of wrap its contents.
- You can specify the flow direction by setting the value of the **FlowDirection** property.
- The FlowLayoutPanel control correctly reverses its flow direction in **Right-to-Left (RTL)** layouts.
- You can also specify whether the FlowLayoutPanel control's contents are
- wrapped or clipped by setting the value of the **WrapContents** property.
- The FlowLayoutPanel control automatically sizes to its contents when you set the **AutoSize** property to true.

The Elements of event driven programs…….. Controls

TABLELAYOU TPANEL

- The TableLayoutPanel control arranges its contents in a **grid**.
  - Because the layout is done both at design time and run time, it can change dynamically as the application environment changes.
- This gives the controls in the panel the ability to **resize proportionally**, so they can respond to changes such as the parent control resizing or text length changing because of localization.
- Any Windows Forms control can be a child of the TableLayoutPanel control, including other instances of TableLayoutPanel.
- This allows you to construct **sophisticated layouts** that adapt to changes at run

The Elements of event driven programs…….. Controls

TABLELAYOU TPANEL…….

- You can also **control the direction of expansion** (horizontal or vertical) after the TableLayoutPanel control is full of child controls.
- By default, the TableLayoutPanel control expands **downward** by **adding rows**.
- You can control the size and style of the rows and columns by using the **RowStyles** and **ColumnStyles** properties. You can set the properties of rows or columns individually.
- The TableLayoutPanel control adds the following properties to its child controls: Cell , Column , Row , ColumnSpan , and RowSpan .

The Elements of event driven programs…….. Container

SPLITCONTAINER

- The Windows Forms **SplitContainer** control can be thought of as a composite control; it's **two panels** separated by a **movable bar**.
- When the mouse pointer is over the bar, the pointer changes shape to show that the bar is movable.
- With the SplitContainer control, you can create complex user interfaces; often, a selection in one panel determines what objects are shown in the other panel.
- This arrangement is effective for **displaying** and **browsing** information.
- Having two panels lets you aggregate information in areas, and the bar, or "splitter," makes it easy for users to resize the panels.

The Elements of event driven programs…….. Container

TABCONTROL

- The TabControl displays multiple tabs, like dividers in a notebook or labels in a set of folders in a filing cabinet.
- The tabs can contain pictures and other controls.
- Use the tab control to produce the kind of multiple-page dialog box that appears many places in the Windows operating system, such as the Control Panel and Display Properties.
- Additionally, the TabControl can be used to create property **pages**, which are used to set a group of related properties.
- The most important property of the TabControl is **TabPages**, which contains the individual tabs. Each individual tab is a TabPage object

The Elements of event driven programs…….. Label Control

- Windows Forms Label controls are used to display text that cannot be edited by the user.
- They're used to identify objects on a form and to provide a description of what a certain control represents or does.
  - For example, you can use labels to add descriptive captions to text boxes, list boxes, combo boxes, and so on.
- You can also write code that changes the text displayed by a label in response to events at run time.
- The caption displayed in the label is contained in the **Text** property.
- The **TextAlign** property allows you to set the alignment of the text within the label.

The Elements of event driven programs…….. Controls Events

- Controls provide events that are raised when the user interacts with the control or when the state of the control changes.
- Common events shared by most controls, events raised by user interaction, and events unique to specific controls.

The Elements of event driven programs…….. Controls Events

- **Common Events**
  - **Controls provides more than 60 events through the base class Control**.
    - These include the **Paint** event, which causes a control to be drawn, events related to displaying a window, such as the Resize and Layout events, and low-level mouse and keyboard events.
    - Some low-level events are synthesized by Control into semantic events such as **Click** and **DoubleClick**.
    - Most shared events fall under these categories:
      - ✓ Mouse events
      - ✓ Keyboard events
      - ✓ Property changed events
      - ✓ Other events

The Elements of event driven programs…….. Controls Events

- Mouse Events
  - Considering Windows Forms is a User Interface (UI) technology, mouse input is the primary way users interact with a Windows Forms application. All controls provide basic mouse-related events:
    - ✓ MouseClick
    - ✓ MouseDoubleClick
    - ✓ MouseDown
    - ✓ MouseEnter
    - ✓ MouseHover
    - ✓ MouseLeave
    - ✓ MouseMove
    - ✓ MouseUp
    - ✓ MouseWheel
    - ✓ Click

The Elements of event driven programs…….. Controls Events

- Keyboard Events
  - If the control responds to user input, such as a TextBox or Button control, the appropriate input event is raised for the control.
  - The control must be **focused** to receive keyboard events.
  - Some controls, such as the Label control, can't be focused and can’t receive keyboard events.
  - The following is a list of keyboard events:
    - KeyDown
    - KeyPress
    - KeyUp

The Elements of event driven programs:

Creating Menus in Your Programs

- Menus provide a convenient way to group similar or related commands in one place.
- Most users are familiar with the **menu bar** concept and expect standard menus such as File, Edit, and Help to appear in their applications.
- Clicking a menu on the menu bar displays a dropdown list of commands.
- While graphical elements like **menus**, **toolbars**, and other constructs make applications much more friendly.

The Elements of event driven programs:

Creating Menus in Your Programs

- The different kinds of menu structures and the classes that support them in the .NET Framework.
- The traditional **menu bar**, sometimes called the **main menu** or an **anchored menu**, is a set of menus shown horizontally across the top of most applications.
- The menus in a typical menu bar display a **dropdown list of commands** when they are activated with the mouse or by a keyboard accelerator.
- Another type of menu is a **context menu**, also called a **popup menu** or **shortcut menu**.
- A context menu is a menu that appears in a particular situation, or context.

The Elements of event driven programs:

Creating Menus in Your Programs

- A context menu is a group of commands or menu items that can be accessed by **right-clicking** on the control surface.
  - It usually contains some frequently used commands for example Cut, Copy and Paste in a text editor.
- In Windows Forms, a context menu is created using the **MenuStrip** control and its command or menu items are **ToolStrip** objects.
- When you use the ToolStrip and MenuStrip controls, you can create toolbars and menus that contain text and images, display submenus, and host other controls such as text boxes and combo boxes.

The Elements of event driven programs:

Adding the following Elements

1. Working with Time and Timers
2. Adding Graphics to Your Programs
3. Adding Help to Your Programs
4. Making Programming Easier with add-Ins
5. Deploying your event driven Applications

Thank you