Roll No.: 80
Date : 31/7/2024

Experiment 4:
Working with Custom controls

Aim:
To implement custom controls in C#.net.

Theory:
Controls which can be created and customised by the user are called custom controls. The user has full control over the display and behavior of controls. The following custom controls will be implemented in C# .net framework.

(1) Listview:
A listview control allows you to display a list of items with item text and optionally an icon to identify the type of item. For example, the windows Explorer list of files is similar in an appearance to a listview control. Listview supports single or multiple selection. The multiple selection feature gets the user select from a list of items in a way similar to listbox control.

Properties:
→ View
→ Label Edit

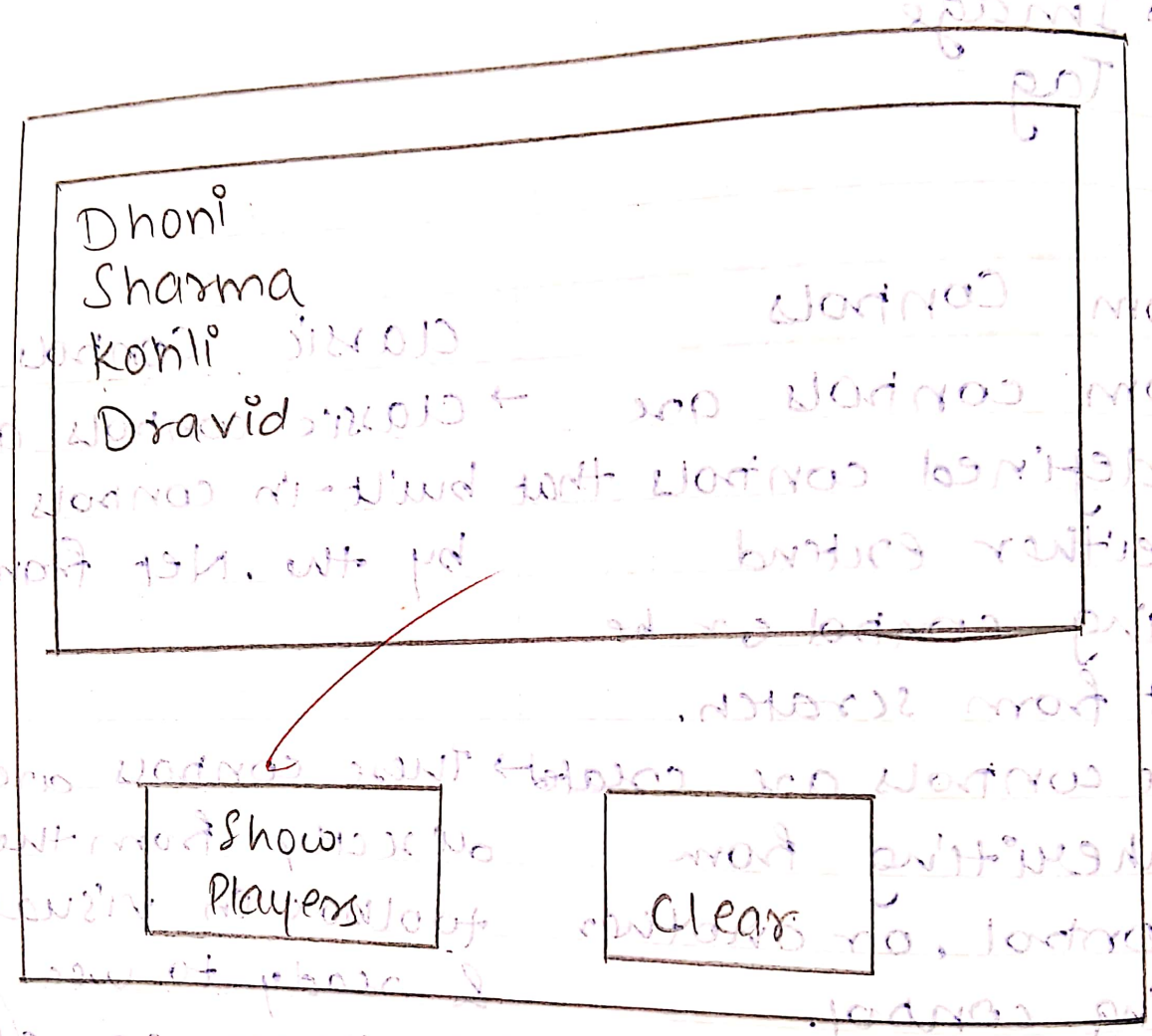
→ Allow Column Reader
→ checkboxes
→ Full row set
→ Gridlines
→ sorting.

(2) Treeview:
With the windows forms Treeview control, you can display a hierarchy of nodes to users, like the way files or folders are displayed in the left pane of the Windows Explorer feature of the windows operating system. Each node in the tree view might contain other nodes called child nodes. You can display parent nodes, or nodes that contain child node as expanded or collapsed. Properties :-
→ Nodes
→ selected Node.

(3) Image List:
Image list is typically used by other controls, such as listview, Tree view or Toolbar. You can add bitmaps or icons to the image list and the other controls are able to use the images as they require. Imagelist uses a handle to manage the list of images. Properties:
→ Colordepth

→ Contain
→ Events
→ Handle
→ Image
→ Tag

Custom Controls	Classic Controls
→ Custom controls are user-defined controls that can either extend existing control or be built from scratch.	→ Classic controls are the built-in controls provided by the .Net framework
→ These controls are created by inheriting from UserControl, or another existing control.	→ These controls are used directly from the toolbox in visual studio & ready to use.
→ They offer extensive customization options, allowing developers to implement unique behaviour, appearance, etc.	→ While you can change properties like size, color, and font, the extent of customization is limited.
→ Once created custom controls can be used across different projects providing a way to standardize complex UI components.	→ These controls are easy to use and integrate because they come with predefined behavior and events.


Dhoni
Sharma
Kohli
Dravid
Show Players
Clear

Procedure:

Experiment 4A

```
public partial class Form1:form
{
    Listview listview1;
    public form1()
    {
     Initialize Component();
     listview1 = new listview();
     listview1.location = new system.Drawing.
                          Point(350, 50);
     listview1.size = newSystem.Drawing.
                      Size(500, 400);
     listview1.Backcolor = System.Drawing.
                           color.lightsalmon;
     listview1.Forecolor = System.Drawing.
                           Color.DarkSalmon;
     listview1.font = new Font("Georgia", 18);
     this.controls.Add(listview1);
}
 private void button1_click(object sender,
                            EventArgs e)
 {
  listviewItem.ListItem1 = new listview Item();
  ListItem1 = listview1.Item.Add("Dhoni");
  listviewItem.ListItem2 = new listview Item();
  ListItem2 = Listview1.Item.Add("Sharma");
```

```
listViewItem listItem3 = new listviewItem();
listview3 = listview1.Item.Add("Dhli");
listview Item listItem4 = new listviewItem();
listItem<1 = listview1.Item.Add("Dravid");
listview1.View = view.List;
}
}
```

Name:
John

Roll No.:
01.

Mail Id:
xyz@gmail.com

Course:
IT

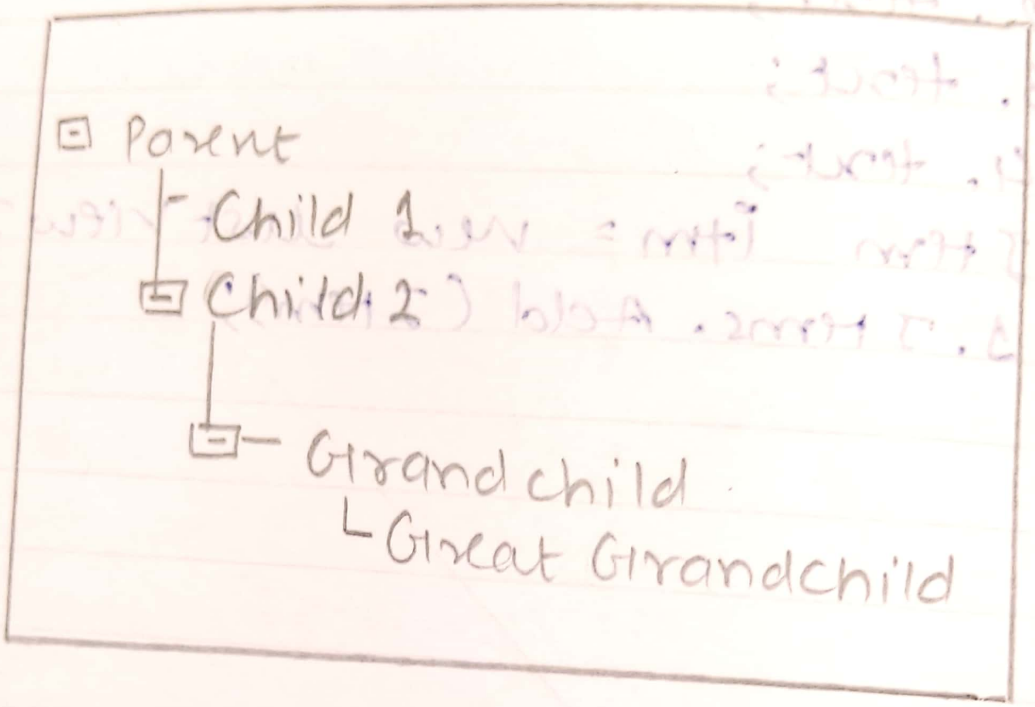
Show

Name	Roll No.	Mail	Course
John	01	xyz@gmail.com	I T.

Experiment 4B

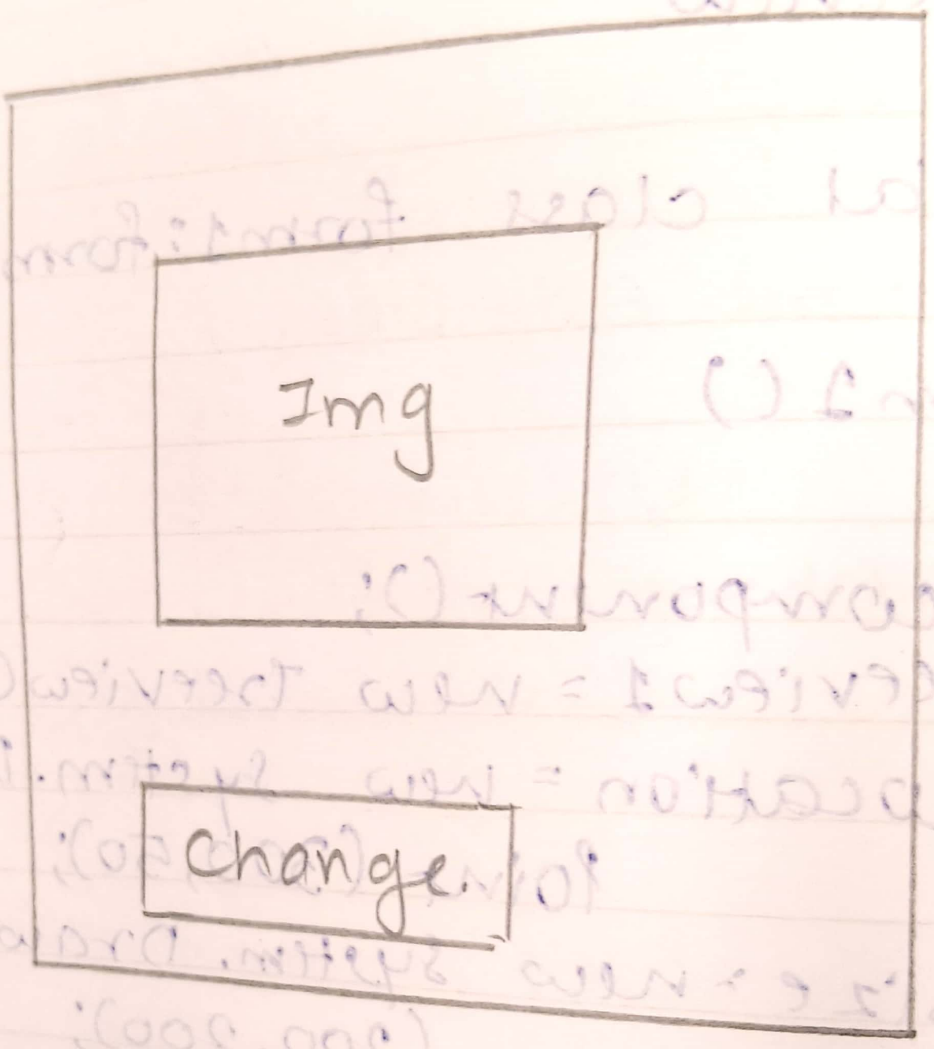
```
public partial class Form1 : form {
    listview listview1;
    public form1 (){
    listview1 = new listview ();
    listview1.location = new System.Drawing.
                         Point (300,50);
    listview1.size = new System.Drawing.
                     Size (450, 200);
    listview1.BackColor = System.Drawing.Color.
                          LightSalmon;
    listview1.Forecolor = System.Drawing.Color.
                          AliceBlue;
    listview1.Font = new Font("Georgia", 18);
    this.Controls.Add (listview1);
    listview1.view = view.Details;
    listview1.columns.Add ("Name", 70,
                  HOrizontal Allignment, Center);
    listview1.columns.Add ("Roll No.", HO 70,
                  Horizontal Alignment, Center);
    listview1.colums.Add ("Mail", 70, Horizontal
                  Alignment, Center);
    listview1.columns.Add ("course", 70,
                  Horizontal Alignment.Center);
    }
private void button1-Click (object sender,
                  Event Args e) {
    String [] arr = new string [] {
```

```
textbox1.text;
textbox2.text;
textbox3.text;
textbox4.text;
ListViewItem item = new ListViewItem(arr);
listView1.Items.Add(Item);
}
}
```



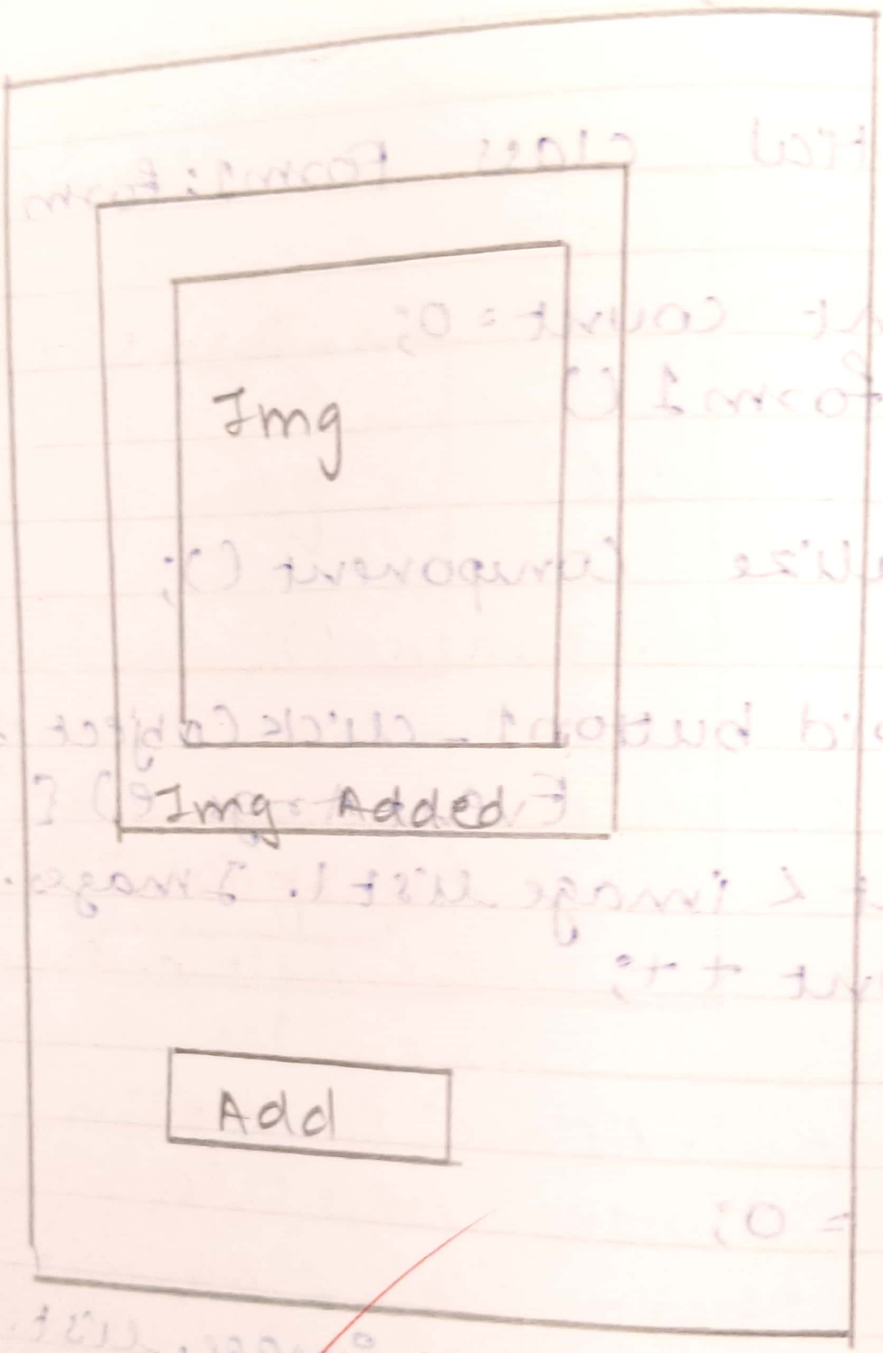
Experiment 4C

```
namespace treeview
{
public partial class form1:form
{
public form1()
{
Initialize component();
Tree View treeview1 = new Treeview();
Treeview1.Location = new System.Drawing.
                     Point(300,50);
Treeview1.size = new system.Drawing.
                 Size (200,200);
this.conmol.Add("Parent");
Treeview.Nodes.Add("Parent");
treeview.Nodes[0].Nodes.Add("Child 1");
Tree view.Nodes[0].Nodes.Add("child 2");
Treeview.Nodes[0].Nodes[1].Nodes.
                 Add("Grandchild");
Treeview.Nodes[0].Nodes[1].Nodes[0].
          Nodes.Add("Great
                 Grandchild");
}
}
}
```



Experiment 4D

```
public partial class Form1 : Form
{
  static int count = 0;
  public Form1()
  {
    Initialize Component();
  }
private void button1_click(object sender,
                           EventArgs e) {
  if (count < imageList1.Images.count - 1) {
      count++;
  }
  else {
    count = 0;
  }
pictureBox1.Image = imageList1.Images
                                [count];
}
private void Form1_load(object sender,
                        EventArgs e)
  {
pictureBox1.Image = imageList1.
                    Images[count];
  }
}
```


Img
Img Added
Add

Experiment 4E

```
public partial class Form1 : Form {
  Static int count = 0;
   public form1 () {
     Initialize component();
  }
 private void button1_click (object sender,
                        EventArgs e)
  {
  if (count < ImageList1. Images. count - 1) {
     count++;
  }
  else {
   count = 0;
  }
  listview1. Items. Add (new listViewItem
                  ("added an image", count)
 }
 private void form1_load (object sender,
                        EventArgs e)
  {
  listview1. largeImageList = ImageList;
  listview1. Items. Add (new listViewItem
                  ("added an Image", 0);
  }
}
```

Conclusion:

We have implemented custome controls to create flexible and reusable interfaces for improved user experience.