**PRACTICAL # 13**

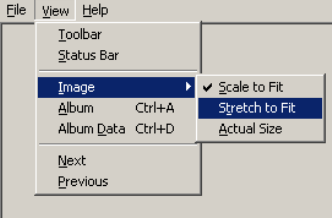
**OBJECT:**

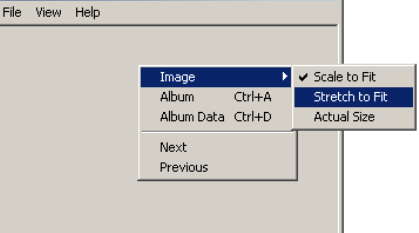
Working with Menus in Windows Forms application

**THEORY:**

Menus provide a convenient and consistent way to group related commands in one place. Standard menus such as File, Edit, and Help appear in their applications. Clicking a menu on the menu bar displays a dropdown list of commands. The consistent appearance of menus and menu items provide a user friendly user interface.

Prior to menus, users had to cope with a wide array of interfaces offered by desktop applications. The function keys at the top of computer keyboards were developed in part as a standard way to access common functions in an application.

The traditional menu bar is a set of menus shown horizontally across the top of most applications (shown next). The menus in a typical menu bar display a dropdown list of commands when they are activated with the mouse or keyboard.

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. Typically, a context menu contains a set of commands or menus related to a specific graphical element of the application. Such menus appear with the right click of mouse (Shown next). For instance, right-click the Windows desktop, any program icon on your screen, and a context menu will pop up with a set of commands related to the desktop display, or the program respectively. Context menus are

typically associated with a specific control, the contents of which

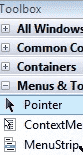
may change to reflect the condition of the control or type of item

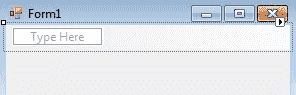
selected within the control.

MENU CLASS

All menus in .NET derive from the Menu class. This class provides the core capabilities required by all menus, such as access to the parent menu, if any, and the collection of submenu items for the menu.

**Program:**

Creating menus in Windows Forms application is made easier with Visual Studio’s drag and drop feature. Simply select and drag menustrip control from the toolbox in Visual Studio and drop it to on the Form, which creates a menu.





After drag the Menustrip on your form you can directly create the menu items. Type a name into the "Type Here" box on the menubar part of your form as shown in the figure next.

To add items in the menustrip, just click on new item and name the menu item. Such menu items will be dummy if you do not add an onClick handler. To perform an action on menu item click, double click the menu item to open the onClick handler code.

**Common menu functions:**

File → Open

Double click Open item and write the following code in the onClick handler to open file dialog.

//onClick handler generated code

*private void exitToolStripMenuItem\_Click(object sender, EventArgs e)*

*{*

***OpenFileDialog dlg = new OpenFileDialog();***

***dlg.ShowDialog();***

***if (dlg.ShowDialog() == DialogResult.OK)***

***{***

***string fileName = dlg.FileName;***

***}***

*}*

File → Exit

Double click Exit item and write the following code in the onClick handler to exit the application:

//onClick handler generated code

*private void exitToolStripMenuItem\_Click(object sender, EventArgs e)*

*{*

*//user code*

***Application.Exit();***

*}*

Help → About

Double click the About item in help menu to show information about the application in a message box. Write the following code to show such information:

//onClick handler generated code

*private void aboutToolStripMenuItem\_Click(object sender, EventArgs e)*

*{*

*MessageBox.Show(“This application demonstrates the use of Menu”);*

*}*

**ACTIVITIES**

**Activity 1**

Write a simple application that loads images in a pictureBox, with open, exit and about application menu options.

**REVIEW QUESTIONS**

1. Which component is required to add the application menu?
2. Which class does menustrip derive from?
3. How do you add menu items in the main menu?
4. How is a menu item click handler code inserted for a menu item?
5. How do you implement Open file menu action?