

Week 3 Lab 1

In this lab you will learn to create a simple word processor MDI application.

1. Create a New Project called WordProcessor in Solution Week3Lab
2. Change the name of the Form1 file to MainForm.cs
3. Change the text displayed in MainForm's title bar to Word Processor
4. Set its isMdiContainer property to True
5. Set its WindowState property to Maximized
6. Add a MenuStrip control to MainForm and insert the standard menu items
7. Add a new Windows form to the project named ChildForm
8. Place a RichTextBox control in your ChildForm and Dock the RichTextBox in the parent container
9. Add the event handler method in MainForm for the File Menu's New menu item that instantiates and shows a new ChildForm

```
ChildForm myChild = new ChildForm();  
myChild.MdiParent = this;  
myChild.Show();
```

10. Compile and test your program
11. Add to your MenuStrip1 a Window menu named windowToolStripMenuItem and place it in-between the Tools and the Help menu
12. Add menu items to Cascade, Tile Horizontally, Tile Vertically, Separator line, Close (Ctrl+F4), and Close All items
13. Include access keys to the menu items
14. To create a MDI child window list in the Window menu, set MenuStrip1's MdiWindowListItem property to windowToolStripMenuItem.
15. Add event handlers to the Window menu items to Cascade, Tile Horizontally, Tile Vertically, Close, and Close All (Use code given below)

```
this.LayoutMdi(MdiLayout.TileVertical);  
this.ActiveMdiChild.Close();  
foreach (ChildForm window in MdiChildren)  
    window.Close();
```

16. To add an event handler method, double-click on the menu item in the Design view
17. Test your application
18. Add a MenuStrip to the ChildForm with two Menus: Edit and Font. Note: to add the Edit Menu, Select "Insert the Standard Items" and then remove the File, Tools, and Help menus
19. Use the Properties window to set the ChildForm's menuStrip1 Visible property to false (The menu strip should not display on the child window)
20. On the ChildForm's standard Edit menu
21. Set its MergeAction to Replace
22. Add event handlers for Undo, Redo, Cut, Copy, Paste, and Select_All using the richTextBox1's built-in methods to the same name

```
private void CutText (object sender, EventArgs e)
{
    richTextBox1.Cut();
}
```

23. On the Font menu, Set its MergeAction to Insert and the MergeIndex to 3

24. Use a FontDialog to allow the user to specify a font style and size for the selected text

```
private void fontStyleToolStripMenuItem1_Click (object
sender, EventArgs e)
{
    fontDialog1.Font = richTextBox1.SelectionFont;
    DialogResult result = fontDialog1.ShowDialog();
    if (result == DialogResult.OK)
    {
        richTextBox1.SelectionFont = fontDialog1.Font;
    }
}
```

25. Use a ColorDialog to allow the user to specify the font color for the selected text

26. Test your application

27. If there is still time remaining during lab, implement the File Open and Save As actions, using *.rtf as the file extension.