###### **Cairo University**

**Faculty of Computers and Information Information Systems Department Database Systems 1**

**Lab 2 (C# Tutorial 2)**

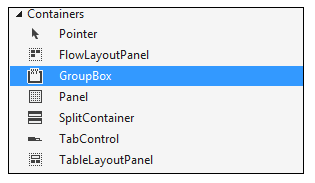
**Creating checkboxes and radio buttons:**

Checkboxes and radio buttons are ways to offer your users choices. Checkboxes allow a user to select multiple options, whereas radio buttons allow only one.

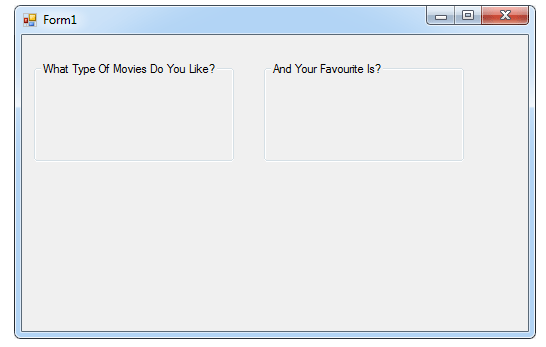
1. Start a new project.
2. It is better to place checkboxes and radio buttons on a GroupBox.

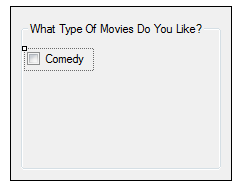
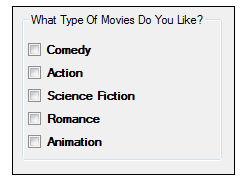
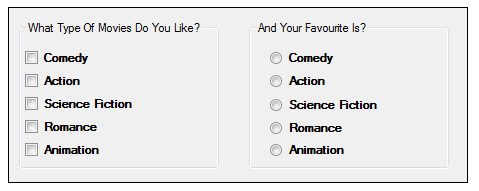
Windows Forms **GroupBox** controls are used to provide an identifiable grouping for other controls. Grouping all options in a **GroupBox** gives the user a logical visual cue, and at design time all the controls can be moved easily. When you move the single **GroupBox** control, all its contained controls move, too.

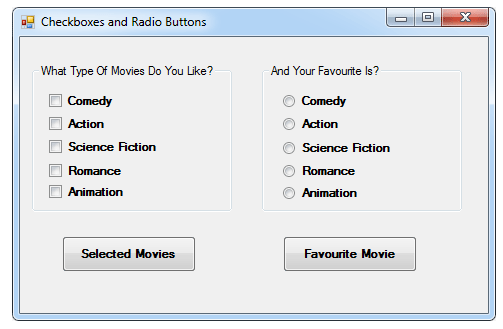
1. Locate the GroupBox control in the Toolbox on the left, under Containers. It looks like this:



1. Draw one out on your form. Locate the **Text** property in the properties window on the right. Change the Text property to **What Type of Movies Do You Like?**.
2. Add a second GroupBox along side of the first one, and set the **Text** property as **And Your Favourite Is?**.
3. Your form will then look like this:



1. We'll place some checkboxes on the first GroupBox, and some radio buttons on the second one.
2. Locate the **Checkbox** control on the toolbox, under **Common Controls**. Draw one out on your first Groupbox.
3. In the properties area on the right, notice that the default Name property is **checkBox1**. Leave it on this, but locate the **Text** property and change it to **Comedy**, it will look like this:
4. Draw four more checkboxes on the GroupBox, and set the Text properties as follows: Action, Science Fiction, Romance, Animation. Your GroupBox should look like this:
5. You add Radio Buttons in the same way. So add five radio buttons to the second GroupBox. Leave the Name property on the defaults. But change the Text to the same as for the Checkboxes. Your form should look like this:
6. Now add two buttons, one below each group box. Set the Text for the first one as Selected Movies. Set the Text for the second one as Favourite Movie. Here's what your form should look like now:



1. Run your form and test it out. What you should find is that you can select as many checkboxes as you like, but only one of the radio buttons.
2. What we'll do now is to write code to get at which selections a user made. First, the Checkboxes.
3. Double click your Selected Movies button to open up the code window. We will make use of the Checked property of Checkboxes. This is either true or false. It will be true if the user places a check in the box, and false if there is no check.
4. We can build up a message, if an option was selected:

string movies = "";

if (checkBox1.Checked)  
{

movies = movies + checkBox1.Text;

}

MessageBox.Show(movies);

1. Inside the if statement, we are building up the string variable we've called movies. We're placing the Text from the checkbox into this variable.
2. Add a second if statement to your code:

string movies = "";

if (checkBox1.Checked)  
{

movies = movies + checkBox1.Text;

}

if (checkBox2.Checked)  
{

movies = movies + checkBox2.Text;

}

MessageBox.Show(movies);

1. The second if statement is the same as the first, except it refers to checkBox 2 instead of checkBox1.
2. Test out your code so far. Run your program and check both boxes. Click your button and the message box should display the following:



1. As you can see, they are both on the same line, with no spacing.
2. To get the choices on separate lines, you can use the new line character, like this:

movies = movies + checkBox1.Text **+** "\n";

1. Return to your code, and add three more if statements.
2. To get at which radio button was chosen, the code is the same as for checkboxes. The only difference is that you need else if, instead of 5 separate if statements:

string ChosenMovie = "";

if (radioButton1.Checked)   
{

ChosenMovie = radioButton1.Text;

}  
else if (radioButton2.Checked)   
{

ChosenMovie = radioButton2.Text;

}

MessageBox.Show(ChosenMovie);

1. Modify the message as follows:

MessageBox.Show("Your Favorite Movie is " + ChosenMovie);

1. To send a value from one form to another:
2. Add two forms to your solution and name them as "Form1.cs" and "Form2.cs" respectively.
3. We will send Form1 [TextBox](http://stepcoder.com/search/articles/textbox) value to Form2 Label Control.
4. So, add one TextBox and a [Button](http://stepcoder.com/search/articles/button) on Form1 and a [Label](http://stepcoder.com/search/articles/label) on Form2.
5. Open Form2.Designer.cs File.
6. Replace this line  "private System.Windows.Forms.Label label1;" with "public System.Windows.Forms.Label label1;"
7. This will make Label1 Public and we can use it in any Form.
8. Open Form1.cs and Double click on [Button](http://stepcoder.com/search/articles/button).
9. Write the below code on click event of the button.

Form2 f = new Form2();

f.label1.Text = this.textBox1.Text;

f.Show();