

Calling the base constructor in C#

```
public class MyExceptionClass : Exception
{
    public MyExceptionClass(string message, string extrainfo) : base(message)
    {
        //other stuff here
    }
}
```

Method Overloading with Different Return Type

```
class A
{
    public int add(int a)
    {
        return 1;
    }
    public string add(int a)
    {
        return "rr";
    }
}
```

We will not get error while Writing the Methode

```
A a = new A();
int gh = a.add(1);
```

```
int A.add(int a)
```

Error:
The call is ambiguous between the following methods or properties: 'ConsoleApplication1.A.add(int)' and 'ConsoleApplication1.A.add(int)'

We will get Error While Calling The Methode

what are all the page level events that get fired when a button is clicked?

1 - Page.Init + Control.Init for every control on the Web Form The first stage in the page life cycle is initialization. After the page's control tree is populated with all the statically declared controls in the .aspx source the Init event is fired. First, the Init event for the Page object occurs, then Init event occurs for each control on the Page. Viewstate information is not available at this stage.

2 - Page.LoadViewState After initialization, ASP.NET loads the view state for the page. ViewState contains the state of the controls the last time the page was processed on the server.

3 - Page.ProcessPostData Post Data gets read from the request and control values are applied to control initialized in stage 1.

4 - Page.Load + Control.Load for each control on the Page If this is the first time the page is being processed (Page.IsPostBack property), initial data binding is performed here.

5 - "Change" events are fired for controls (TextChanged, SelectedIndexChanged, and similar) The current value (from Post Data) is compared to the original value located in the ViewState. If there is a difference "Changed" events are raised.

6 - Server-side events are fired for any validation controls

7 - Button.Click + Button.Command The Click and Command events are fired for the button that caused the postback

8 - Page.PreRender + Control.PreRender

9 - Page.SaveViewState New values for all the controls are saved to the view state for another round-trip to the server.

10 - Page.Render

How to add dynamic n checkbox columns to gridview ASP.NET

This is my grid aspx code

```
<asp:GridView ID="gv" runat="server" AutoGenerateColumns="false">
    <Columns>
    </Columns>
</asp:GridView>
```

First of all add num number of template fields to your gridview

```
protected void Page_PreInit(object sender, EventArgs e)
{
    int num = Request.QueryString["num"];

    for (int i = 0; i < num; i++)
    {
        TemplateField tf = new TemplateField();
        tf.HeaderText = "Status";
        gv.Columns.Add(tf);
    }
}
```

After adding template feilds, now we will add checkboxes to the gridview. We write a function to add checkboxes. Below is the code

```
private void AddCheckBox()
{
    int num = Request.QueryString["num"];
    for (int i = 0; i < num; i++)
    {
        foreach (GridViewRow row in gv.Rows)
        {
```

```

        if (row.RowType == DataControlRowType.DataRow)
        {
            CheckBox cb = new CheckBox();
            cb.Checked = true;
            row.Cells[i].Controls.Add(cb);
        }
    }
}

```

Now place this function in your grid databound event.

```

protected void gv_DataBound(object sender, EventArgs e)
{
    AddCheckBox();
}

```

At the end also call the function in the page load event so first time when grid loads it shows checkboxes are checked

```

protected void Page_Load(object sender, EventArgs e)
{
    if (!IsPostBack)
    {
        AddCheckBox();
    }
}

```

To

Stored Procedures - Output Parameters & Return Values

```

CREATE PROC sales_for_type @type VARCHAR(55), @total_sales INT OUTPUT
AS
SELECT SUM(qty) FROM sales a, titles b
WHERE
a.title_id = b.title_id
and
b.type = @type

```

This procedure can be executed as follows:

```

DECLARE @total_sales_business int
EXEC sales_for_type business, @total_sales=@total_sales_business OUTPUT

```