Debugging And Error Handling

Chapter 9

Learning Outcomes

- Differentiate the FOUR(4) main types of errors
- Use exception handling methods to handle errors
- Debug a program

Error

- To minimize the mistakes
  - We can identify the portions of code that are most prone to errors, to perform testing and exception handling
  - We can adhere to good coding practice that facilitate troubleshooting

Exception = Error

- The exceptions are any error condition or **unexpected behavior** that occurs during the execution of a program, and consequently disrupts the normal flow of execution.
- They can be because of user, logic or system errors.

4 Types of Errors

1. Parser Errors
2. Compilation Errors
3. Configuration Errors
4. Runtime /Logical Errors

Parser Error

Parse errors are only errors with the ***syntax*** *of* ***HTML****.*

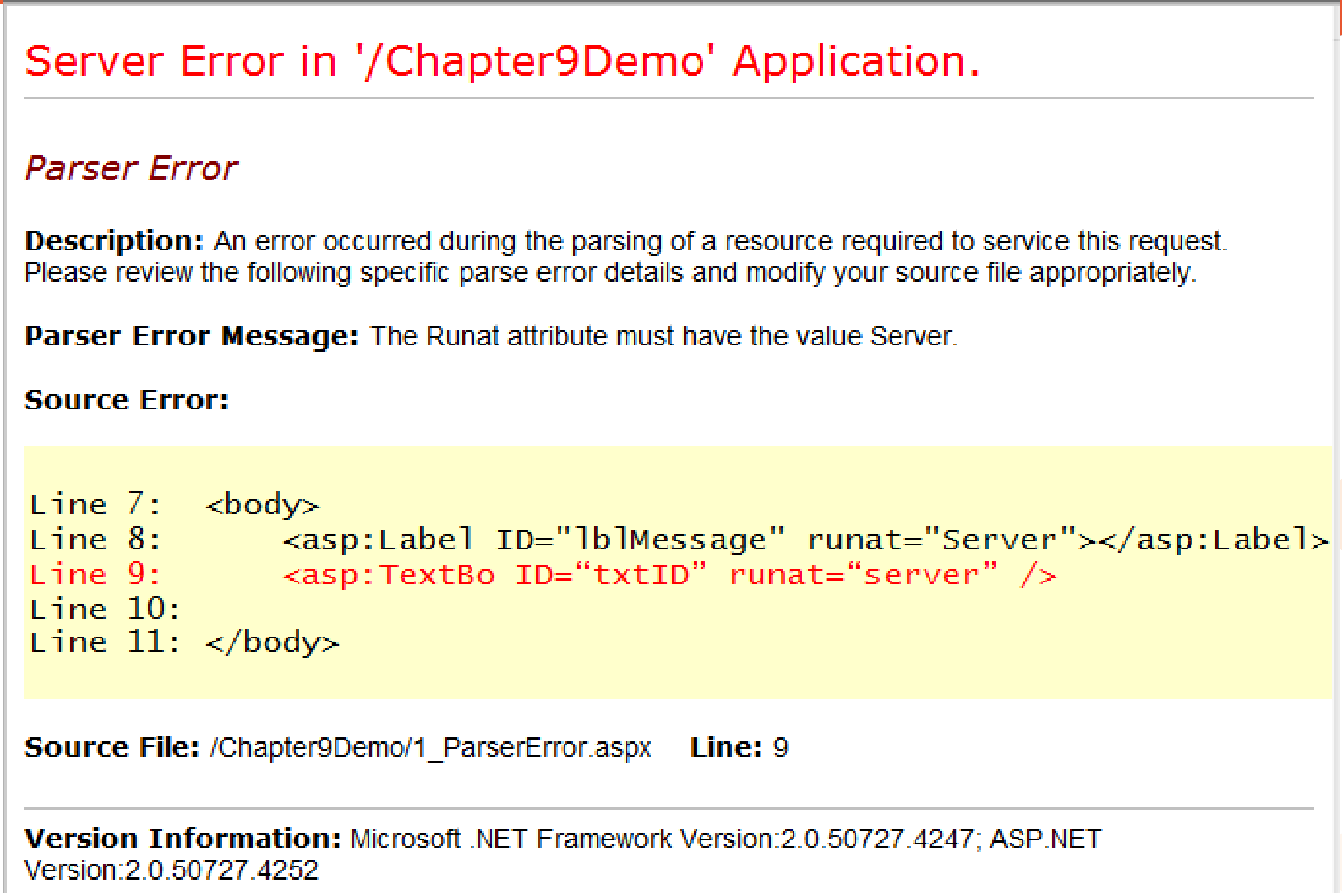
- Example:
  - <asp:TextBo ID="txtID" runat="server" />

- *Reference:*

  http://www.whatwg.org/specs/web-apps/current-work/multipage/syntax.html#parsing

TARUMT TUNKU ABDUL RAHMAN UNIVERSITY OF MANAGEMENT AND TECHNOLOGY

Parser Error



Server Error in '/Chapter9Demo' Application.

Parser Error

Description: An error occurred during the parsing of a resource required to service this request. Please review the following specific parse error details and modify your source file appropriately.

Parser Error Message: The Runat attribute must have the value Server.

Source Error:

```
Line 7:  <body>
Line 8:      <asp:Label ID="lblMessage" runat="Server"></asp:Label>
Line 9:      <asp:TextBo ID="txtID" runat="server" />
Line 10:
Line 11: </body>
```

Source File: /Chapter9Demo/1_ParserError.aspx **Line:** 9

Version Information: Microsoft .NET Framework Version:2.0.50727.4247; ASP.NET Version:2.0.50727.4252

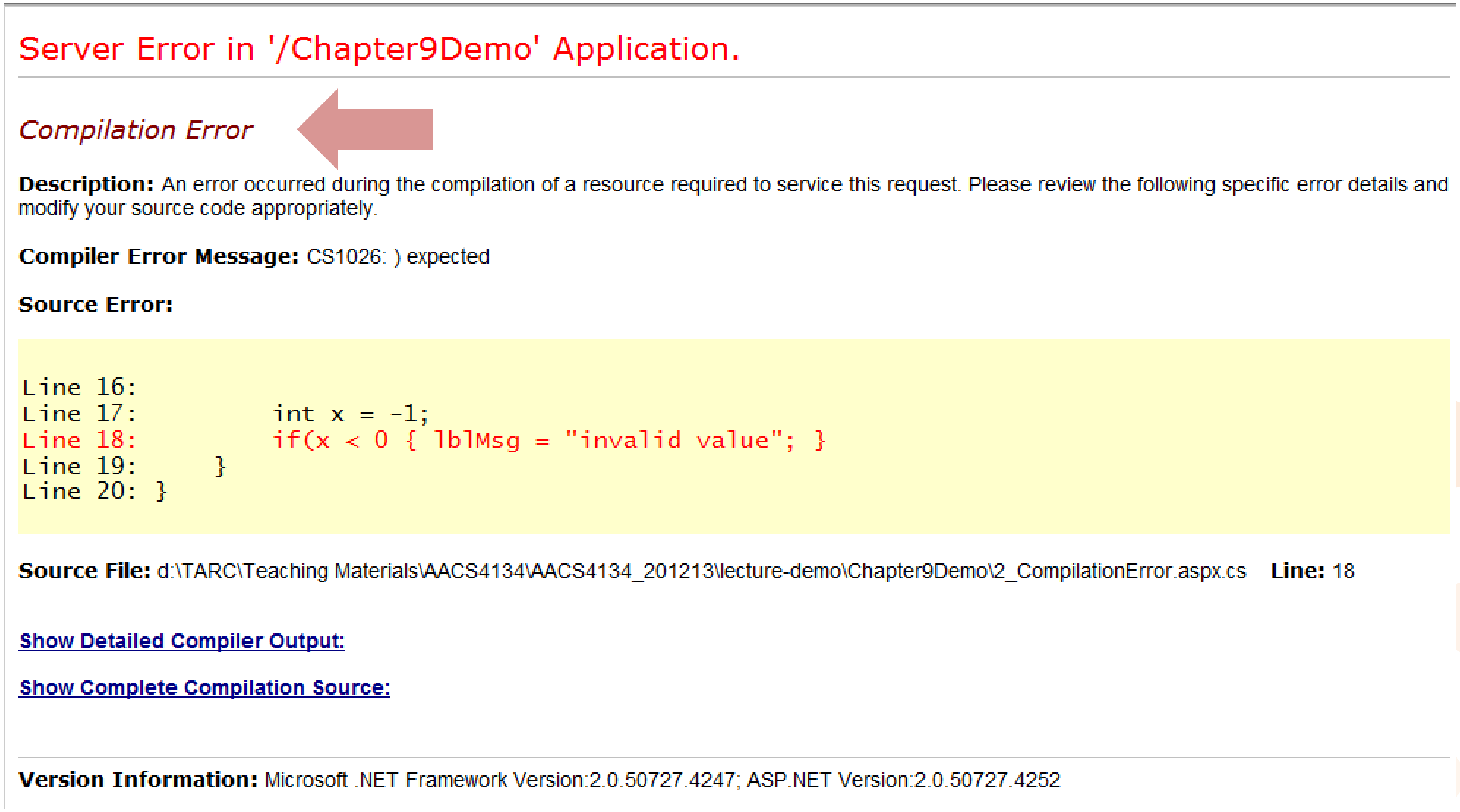
Compilation Error

Errors with the syntax of the statement, which cannot be recognized by the language compiler, such as C# or VB.NET

- Example
  if(x < 0 { lblMsg = “invalid value”; }

Can you identify the 2 syntax errors?

Compilation Error



Server Error in '/Chapter9Demo' Application.

Compilation Error

Description: An error occurred during the compilation of a resource required to service this request. Please review the following specific error details and modify your source code appropriately.

Compiler Error Message: CS1026:) expected

Source Error:

```
Line 16:
Line 17:            int x = -1;
Line 18:            if(x < 0 { lblMsg = "invalid value"; }
Line 19:        }
Line 20:    }
```

Source File: d:\TARC\Teaching Materials\AACS4134\AACS4134_201213\lecture-demo\Chapter9Demo\2_CompilationError.aspx.cs **Line:** 18

Show Detailed Compiler Output:

Show Complete Compilation Source:

Version Information: Microsoft .NET Framework Version:2.0.50727.4247; ASP.NET Version:2.0.50727.4252

Parser V.S. Compilation Errors

Parser Error	Compilation Error
• occurs when there is a syntax error in the HTML code • It is raised when the page is in the process of being parsed	• occurs when there is a syntax error in the C# code block • It is raised when the page is in the process of being compiled

Both involves Syntax Errors

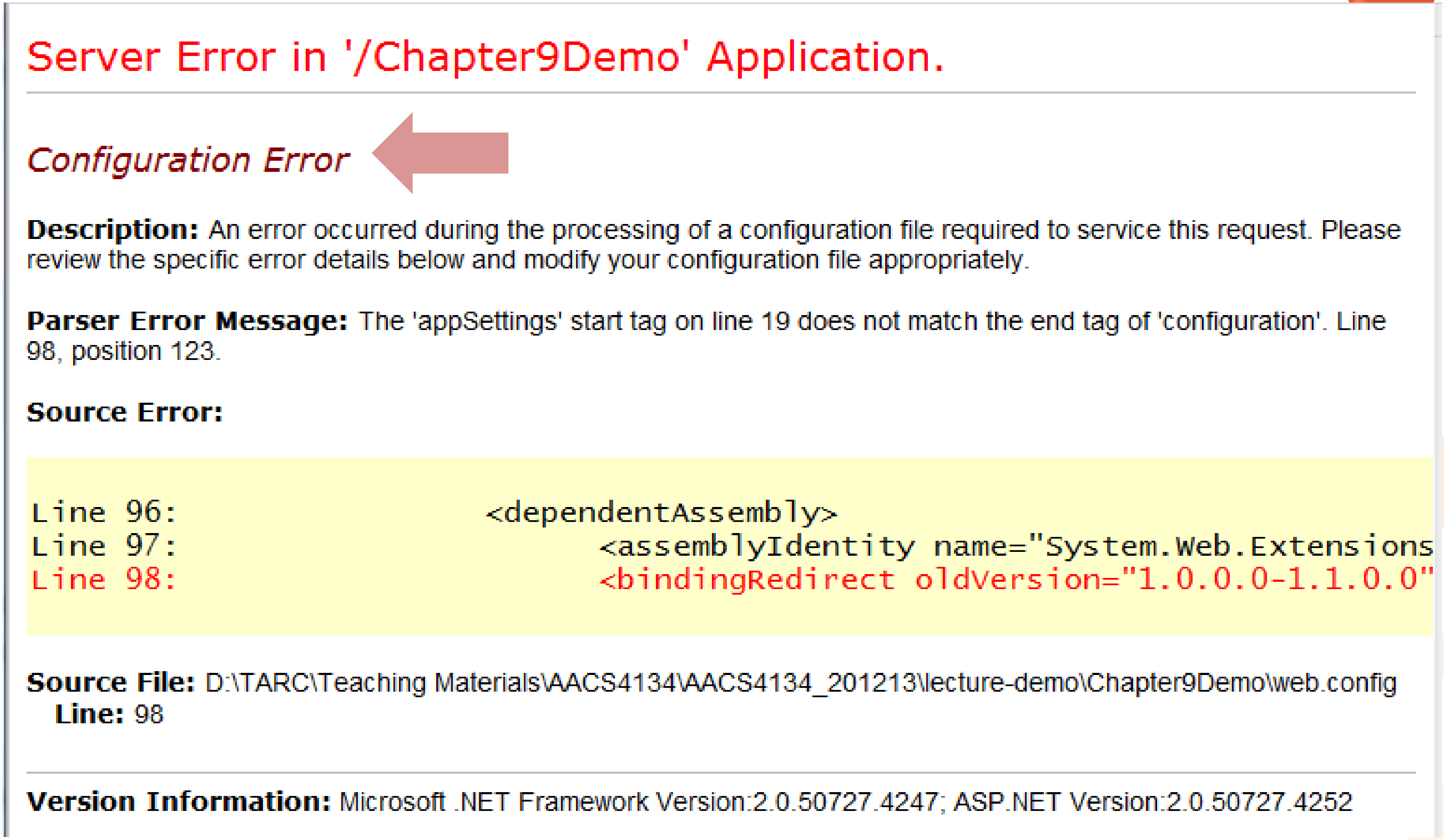
Question

- Discuss the similarity and a difference between Parser Error and Compilation Error.

Configuration Error

- Caused by problem in either the *Web.Config* or the *Machine.Config* file.
- Example of error:
  – Missing of </appSettings> in Web.config

Configuration Error



Server Error in '/Chapter9Demo' Application.

Configuration Error

Description: An error occurred during the processing of a configuration file required to service this request. Please review the specific error details below and modify your configuration file appropriately.

Parser Error Message: The 'appSettings' start tag on line 19 does not match the end tag of 'configuration'. Line 98, position 123.

Source Error:

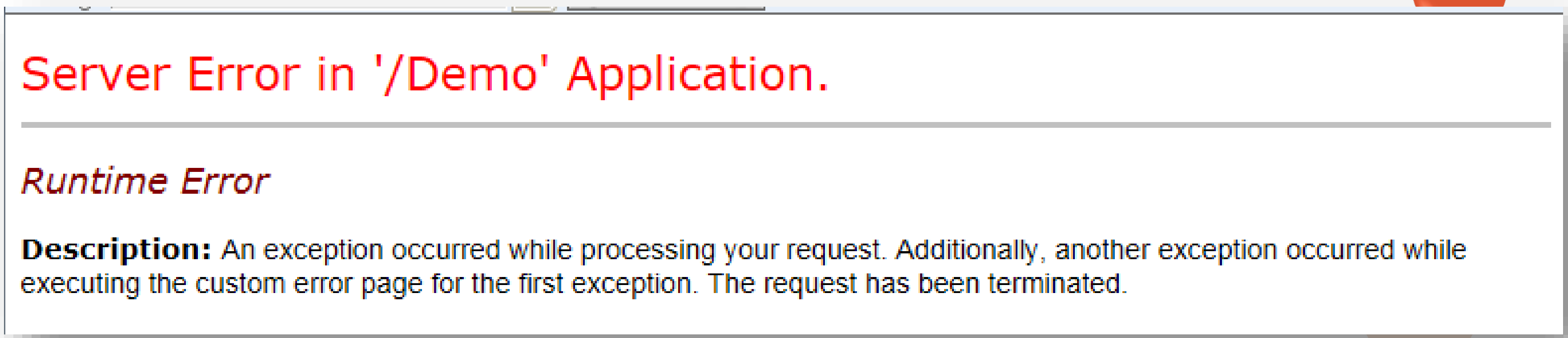
```
Line 96:                        <dependentAssembly>
Line 97:                              <assemblyIdentity name="System.Web.Extensions
Line 98:                              <bindingRedirect oldVersion="1.0.0.0-1.1.0.0"
```

Source File: D:\TARC\Teaching Materials\AACS4134\AACS4134_201213\lecture-demo\Chapter9Demo\web.config **Line:** 98

Version Information: Microsoft .NET Framework Version:2.0.50727.4247; ASP.NET Version:2.0.50727.4252

TARUMT TUNKU ABDUL RAHMAN UNIVERSITY OF MANAGEMENT AND TECHNOLOGY

Runtime / Logical Error


Server Error in '/Demo' Application.

Runtime Error

Description: An exception occurred while processing your request. Additionally, another exception occurred while executing the custom error page for the first exception. The request has been terminated.

Runtime error is error occurs during runtime, after a page is successfully compiled

Logical error occurs due to logical error of the code.

- Runtime/logical error cannot be detected by compiler

Runtime Error vs. Logic Error

runtime error

- Any error occurs during runtime, including logical error.
- Example:
  - invalid query string
  - attempt to open a database connection which was not closed previously
  - Server failed

logical error

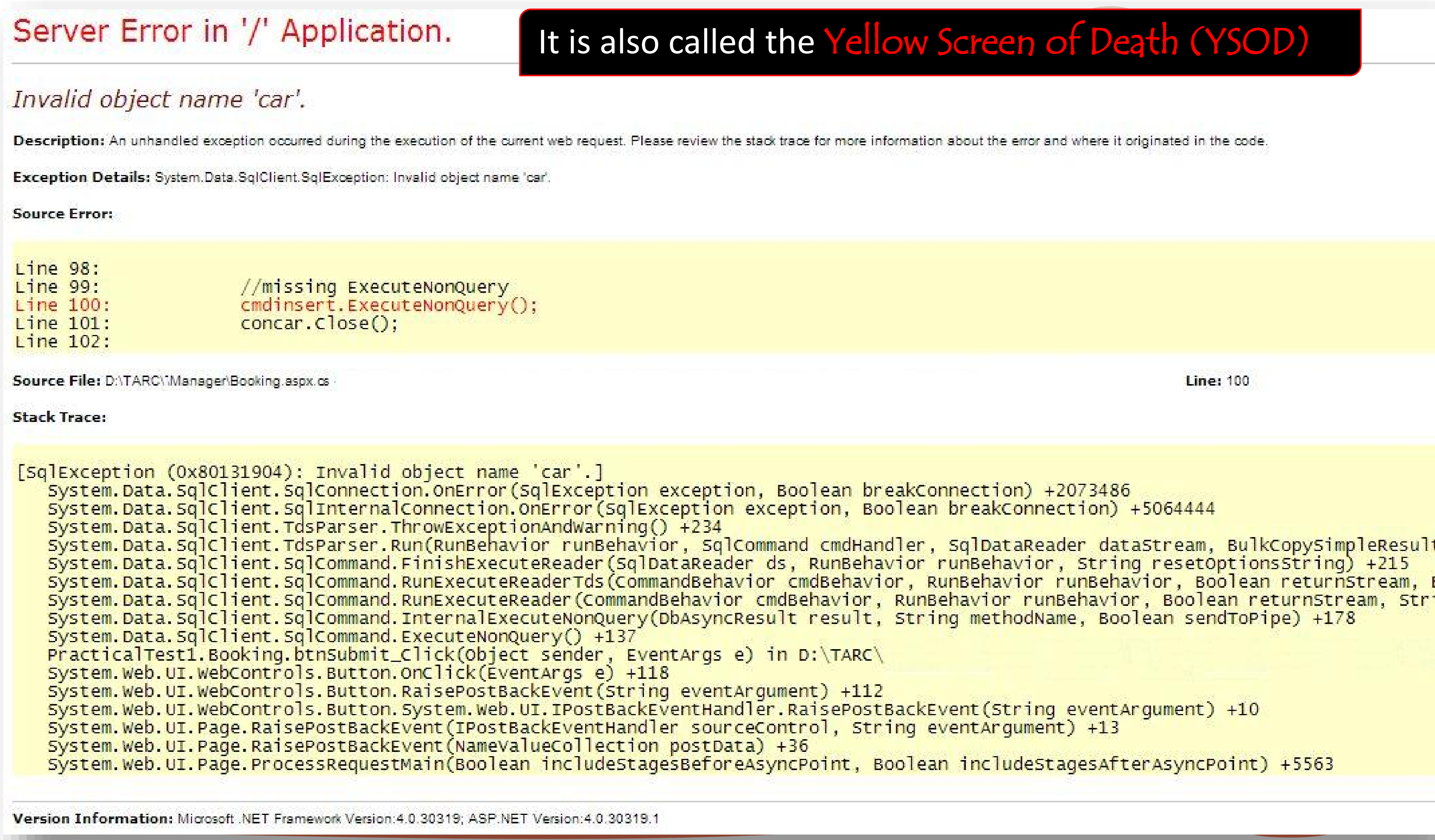
- Error occurs due to invalid logic.
- Example:
  - Attempt to convert a non-numeric value to integer
  - Attempt to divide a number by constant zero (e.g. 5/0)

Handle an exception

- If no mechanism is used to handle these anomalies, the .NET run time environment provide a default mechanism, which **terminates** the **program execution**.
- Exception handling is a built-in mechanism in .NET framework to detect and handle run time errors.

TUNKU ABDUL RAHMAN UNIVERSITY OF MANAGEMENT AND TECHNOLOGY

The .NET Default Exception Handling Mechanism



Server Error in '/' Application.

It is also called the Yellow Screen of Death (YSOD)

Invalid object name 'car'.

Description: An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated in the code.

Exception Details: System.Data.SqlClient.SqlException: Invalid object name 'car'.

Source Error:

```
Line 98:
Line 99:            //missing ExecuteNonQuery
Line 100:           cmdinsert.ExecuteNonQuery();
Line 101:           concar.Close();
Line 102:
```

Source File: D:\TARC\\Manager\Booking.aspx.cs   **Line:** 100

Stack Trace:

```
[SqlException (0x80131904): Invalid object name 'car'.]
   System.Data.SqlClient.SqlConnection.OnError(SqlException exception, Boolean breakConnection) +2073486
   System.Data.SqlClient.SqlInternalConnection.OnError(SqlException exception, Boolean breakConnection) +5064444
   System.Data.SqlClient.TdsParser.ThrowExceptionAndWarning() +234
   System.Data.SqlClient.TdsParser.Run(RunBehavior runBehavior, SqlCommand cmdHandler, SqlDataReader dataStream, BulkCopySimpleResult
   System.Data.SqlClient.SqlCommand.FinishExecuteReader(SqlDataReader ds, RunBehavior runBehavior, String resetOptionsString) +215
   System.Data.SqlClient.SqlCommand.RunExecuteReaderTds(CommandBehavior cmdBehavior, RunBehavior runBehavior, Boolean returnStream, B
   System.Data.SqlClient.SqlCommand.RunExecuteReader(CommandBehavior cmdBehavior, RunBehavior runBehavior, Boolean returnStream, Str
   System.Data.SqlClient.SqlCommand.InternalExecuteNonQuery(DbAsyncResult result, String methodName, Boolean sendToPipe) +178
   System.Data.SqlClient.SqlCommand.ExecuteNonQuery() +137
   PracticalTest1.Booking.btnSubmit_Click(Object sender, EventArgs e) in D:\TARC\
   System.Web.UI.WebControls.Button.OnClick(EventArgs e) +118
   System.Web.UI.WebControls.Button.RaisePostBackEvent(String eventArgument) +112
   System.Web.UI.WebControls.Button.System.Web.UI.IPostBackEventHandler.RaisePostBackEvent(String eventArgument) +10
   System.Web.UI.Page.RaisePostBackEvent(IPostBackEventHandler sourceControl, String eventArgument) +13
   System.Web.UI.Page.RaisePostBackEvent(NameValueCollection postData) +36
   System.Web.UI.Page.ProcessRequestMain(Boolean includeStagesBeforeAsyncPoint, Boolean includeStagesAfterAsyncPoint) +5563
```

Version Information: Microsoft .NET Framework Version:4.0.30319; ASP.NET Version:4.0.30319.1

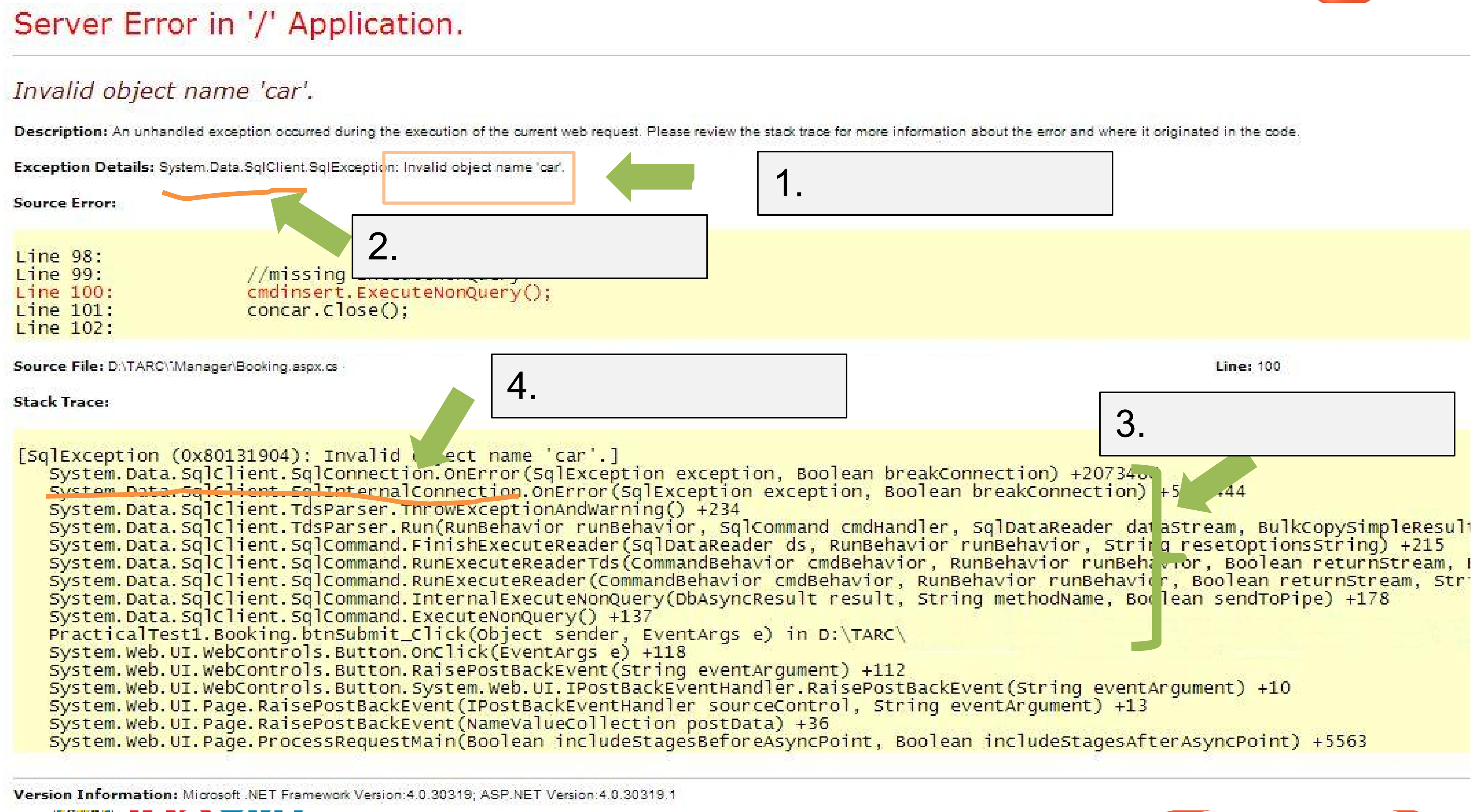
Standard Exceptions

- Exceptions are represented within the .NET framework with instances of the Exception class.
- The exception class consists of some standard properties that we always use to detect the exceptions.

Exception Class Properties

Properties	Description
Message	Returns a string that represents the error message.
Source	Returns a string representing the object or application that caused the error.
StackTrace	Returns a string that represents the methods called immediately before the error occurred.
TargetSite	Returns a MethodBase object that represents the method that caused the error

Exception Class Properties



Server Error in '/' Application.

Invalid object name 'car'.

Description: An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated in the code.

Exception Details: System.Data.SqlClient.SqlException: Invalid object name 'car'.

Source Error:

```
Line 98:
Line 99:          //missing
Line 100:         cmdinsert.ExecuteNonQuery();
Line 101:         concar.Close();
Line 102:
```

Source File: D:\TARC\Manager\Booking.aspx.cs   **Line:** 100

Stack Trace:

```
[SqlException (0x80131904): Invalid object name 'car'.]
   System.Data.SqlClient.SqlConnection.OnError(SqlException exception, Boolean breakConnection) +20734
   System.Data.SqlClient.SqlInternalConnection.OnError(SqlException exception, Boolean breakConnection) +5[illegible]44
   System.Data.SqlClient.TdsParser.ThrowExceptionAndWarning() +234
   System.Data.SqlClient.TdsParser.Run(RunBehavior runBehavior, SqlCommand cmdHandler, SqlDataReader dataStream, BulkCopySimpleResult
   System.Data.SqlClient.SqlCommand.FinishExecuteReader(SqlDataReader ds, RunBehavior runBehavior, String resetOptionsString) +215
   System.Data.SqlClient.SqlCommand.RunExecuteReaderTds(CommandBehavior cmdBehavior, RunBehavior runBehavior, Boolean returnStream, B
   System.Data.SqlClient.SqlCommand.RunExecuteReader(CommandBehavior cmdBehavior, RunBehavior runBehavior, Boolean returnStream, Str
   System.Data.SqlClient.SqlCommand.InternalExecuteNonQuery(DbAsyncResult result, String methodName, Boolean sendToPipe) +178
   System.Data.SqlClient.SqlCommand.ExecuteNonQuery() +137
   PracticalTest1.Booking.btnSubmit_Click(Object sender, EventArgs e) in D:\TARC\
   System.Web.UI.WebControls.Button.OnClick(EventArgs e) +118
   System.Web.UI.WebControls.Button.RaisePostBackEvent(String eventArgument) +112
   System.Web.UI.WebControls.Button.System.Web.UI.IPostBackEventHandler.RaisePostBackEvent(String eventArgument) +10
   System.Web.UI.Page.RaisePostBackEvent(IPostBackEventHandler sourceControl, String eventArgument) +13
   System.Web.UI.Page.RaisePostBackEvent(NameValueCollection postData) +36
   System.Web.UI.Page.ProcessRequestMain(Boolean includeStagesBeforeAsyncPoint, Boolean includeStagesAfterAsyncPoint) +5563
```

Version Information: Microsoft .NET Framework Version:4.0.30319; ASP.NET Version:4.0.30319.1

TUNKU ABDUL RAHMAN UNIVERSITY OF MANAGEMENT AND TECHNOLOGY

The Exception Details YSOD Includes Information About the Exception

Exception Handling Methods

- Page-Level Error Handling
  - try…catch… finally
  - Page_Error() method
- Application-Level Error Handling
  - Application_Error() method

Page Level Error Handling

- This method is used when we need to catch and handle any error that occurs in a page

use *try…catch…finally* to identify the **portions of code** that are more likely prone to errors

use *Page_Error()* method to handle **any errors** at the **page** level

Try…Catch…Finally

```
try
{    //Statement which might fail at runtime  }
catch (Exception e)
{    //Error handling block  }
finally
{
     //Statement is executed irrespective of the fact
     that an exception has been raised
}
```

try encloses the statements that might throw an exception

catch handles an exception if one exists

finally is optional. Use it if you like to FORCE the code enclosed to be executed

Page_Error() Method

- Page error event is raised whenever there is an **unhandled exception occur within a page**.
- You can catch and handle those unhandled errors that occur in a page by using the *Page_Error( )* event handler.

Page_Error() Method

```
void Button1_Click(object sender, EventArgs e)
{         int x = Convert.ToInt32("abc");    }

void Page_Error()
{
      Response.Write("<p><h1>Sorry, there was an error:<br />");
      Response.Write(Server.GetLastError().Message + "</h1></p>");
      Server.ClearError(); //comment this line to see the difference
}
```

This error is unhandled

The *GetLastError* method returns the last exception thrown

To display the *Message* of the last exception thrown.

The *ClearError* method clears the last exception thrown.

Application-Level Error Handling

We use *Application_Error()* Method to handle the errors **in any page**, regardless where they occur **within an application**

- For example, we can use this event handler to log the errors to a log file, notify an administrator using email, or store the information into a database for debugging later.

Application_Error() Method

- Write the *Application_Error()* event handler in the **Global.asax** file.

```
void Application_Error(object sender, EventArgs e)
{
        // Code that runs when an unhandled error occurs
        Application.Lock();
        Application["ErrorMsg"] += Server.GetLastError().Message;
        Application.UnLock();
}
```

```
d:\Demo\4_RuntimeError.aspx.cs(18): error CS0020: Division by constant zero
```

try-catch, Page_Error and Application_Error

try-catch

- handles error(s) in an **identified portion of code**

Page_Error()

- it is called whenever an unhandled exception (by the try-catch) is thrown **within a page**

Application_Error()

- it is called whenever an unhandled exception (by try-catch and Page_Error) is thrown **within an application**

TARUMT TUNKU ABDUL RAHMAN UNIVERSITY OF MANAGEMENT AND TECHNOLOGY

Question

- Why error handling is important? Suggest **TWO (2)** ways of handling runtime errors.

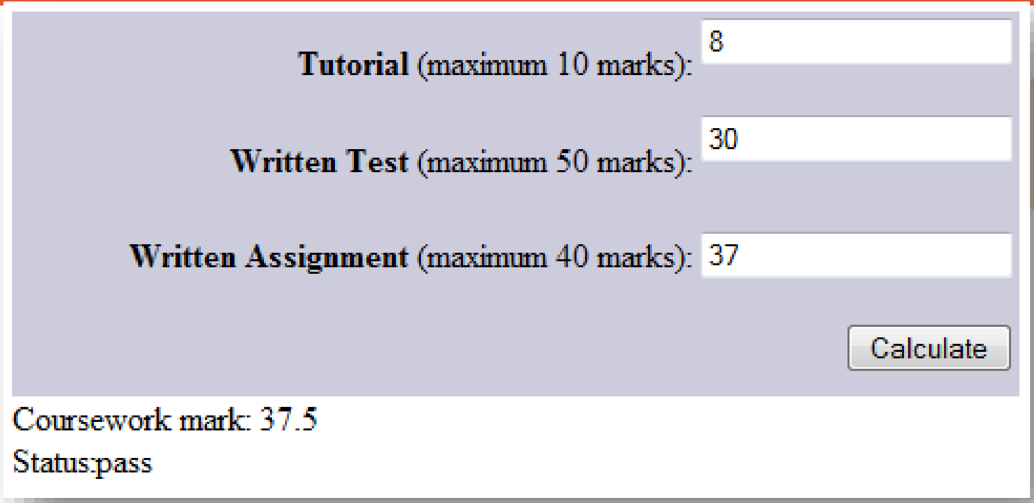
Questions

Refer to the following code, answer the following questions.

```
void btnCal_Click(object sender, EventArgs e)
{
  double Total = Convert.ToInt32(txtTotal.Text);
  double Amount = Convert.ToInt32(txtAmount.Text);
  lblAnswer.Text = Convert.ToString(Amount / Total);
  lblMsg.Text = “”;
}
```

1. Identify and explain TWO potential errors that could be raised by the highlighted code above. Then name the **type** of the potential errors that you have suggested.
2. Suggest which exception approach is better to catch the potential errors in Q1 above, and justify your answer.
3. Write C# code to demonstrate how to handle the potential errors raised by the highlighted code above by using *Page_Error()* method.

Questions



Tutorial (maximum 10 marks):	8
Written Test (maximum 50 marks):	30
Written Assignment (maximum 40 marks):	37

Coursework mark: 37.5
Status:pass

1. Assume that no validation is done on the form above. Predict and explain one runtime exception that can be caused by the user.
2. Discuss why would you use the Try…Catch block to handle the above error rather than the Page_Error() method.
3. Demonstrate in C# code, how to handle the potential exception as suggested in Q1 by using a try…catch code block.

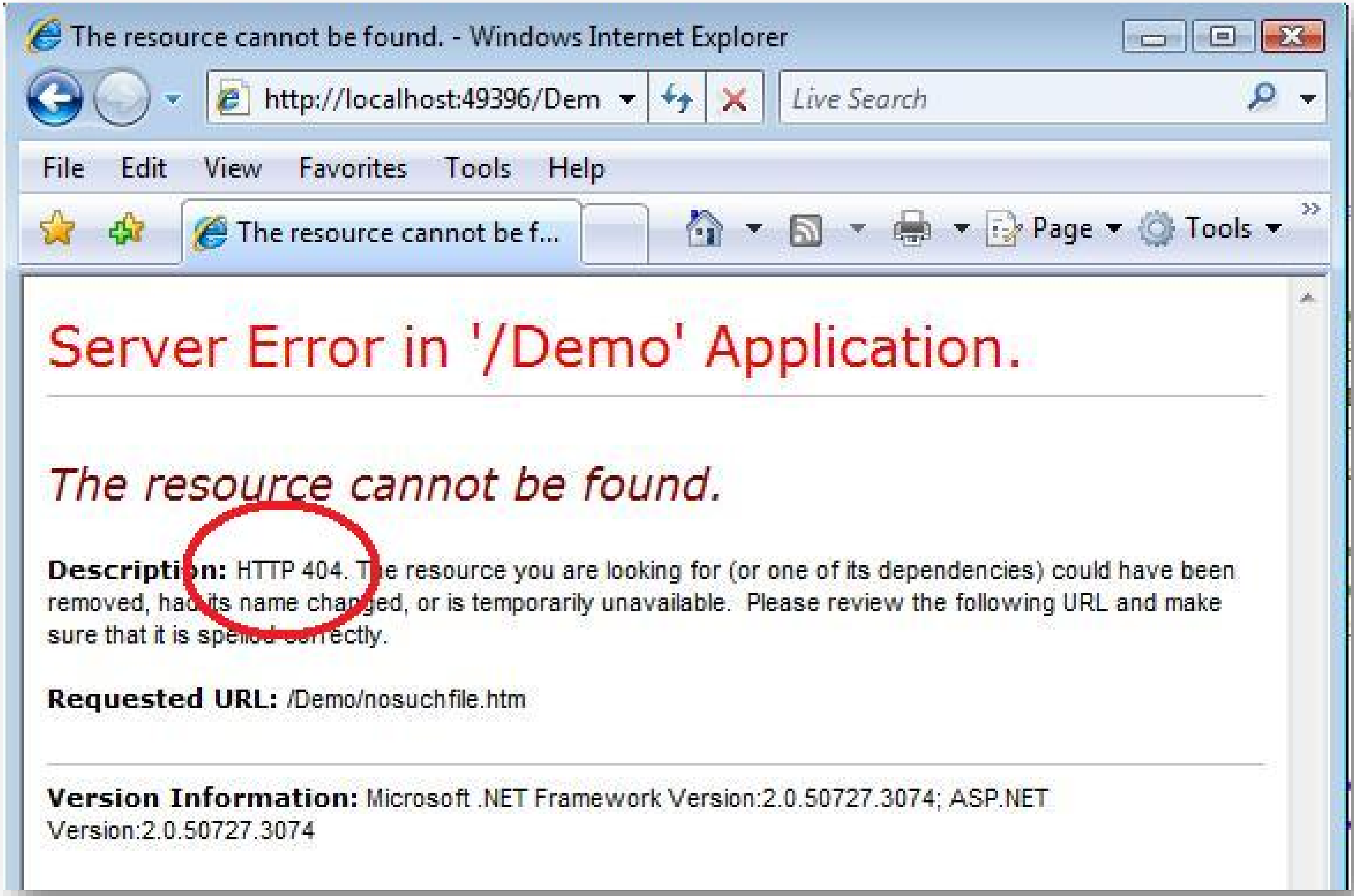
Custom Error Pages

We should avoid public users to see the technical details of exception

- We should provide them a more user-friendly custom-made error page

Custom Error Page allows us to map different types of errors to different page

- E.g. We would display different messages for page-not-found error and other errors


The resource cannot be found. - Windows Internet Explorer
http://localhost:49396/Dem
Live Search
File
Edit
View
Favorites
Tools
Help
The resource cannot be f...
Page
Tools

Server Error in '/Demo' Application.

The resource cannot be found.

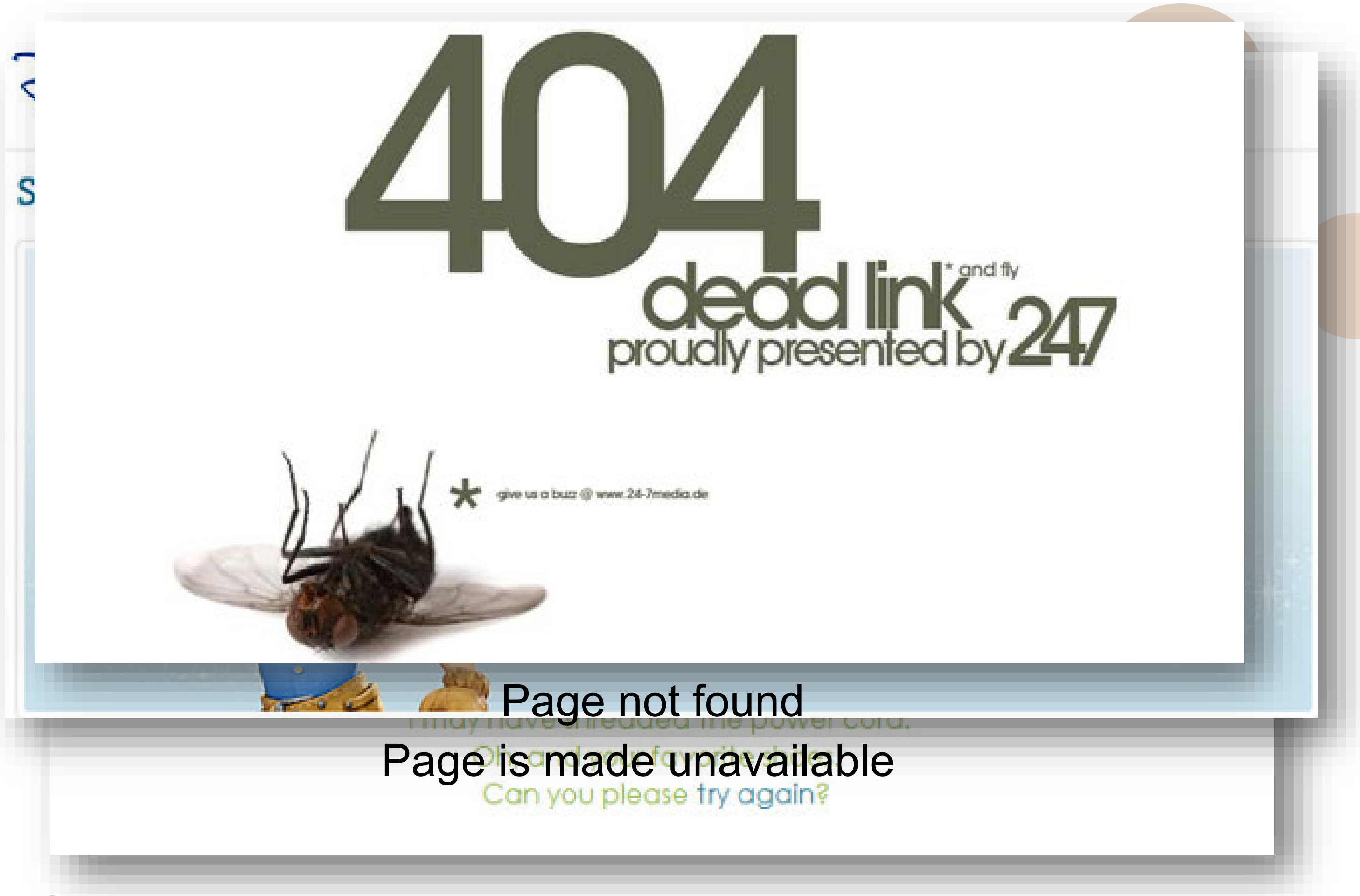
Description: HTTP 404. The resource you are looking for (or one of its dependencies) could have been removed, had its name changed, or is temporarily unavailable. Please review the following URL and make sure that it is spelled correctly.

Requested URL: /Demo/nosuchfile.htm

Version Information: Microsoft .NET Framework Version:2.0.50727.3074; ASP.NET Version:2.0.50727.3074

the YSOD that shows the HTTP status code

Examples of custom error page


404
dead link
* and fly
proudly presented by 247
* give us a buzz @ www.24-7media.de
Can you please try again?

Page not found

Page is made unavailable

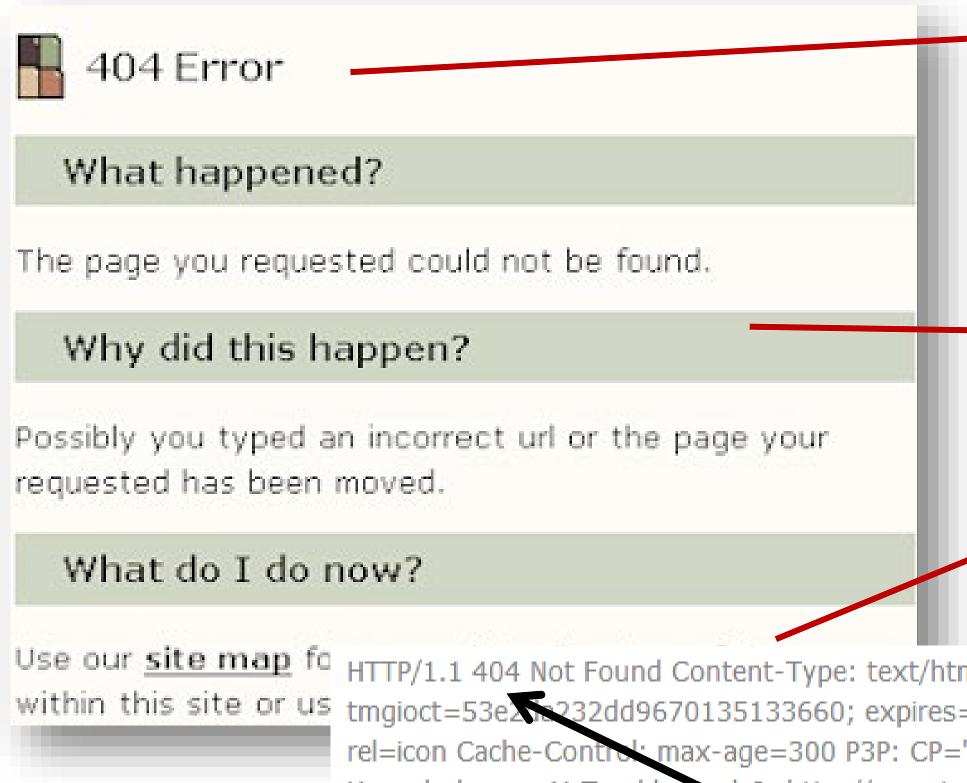
Server error

Question

- Discuss the benefits of using custom error pages in an application.

Designing a Custom Error Page

Include in the body (content) area:



404 Error

What happened?

The page you requested could not be found.

Why did this happen?

Possibly you typed an incorrect url or the page your requested has been moved.

What do I do now?

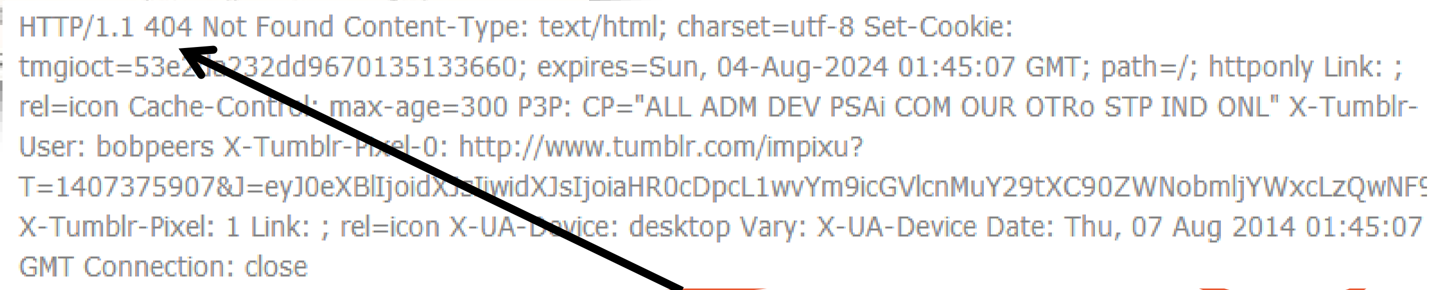
Use our **site map** fo

within this site or us

The proper error code *headers* (it is for search engines and your server's analytic software)

a precise description of what has happened, written in plain English, not a technical explanation.

a link to the website sitemap

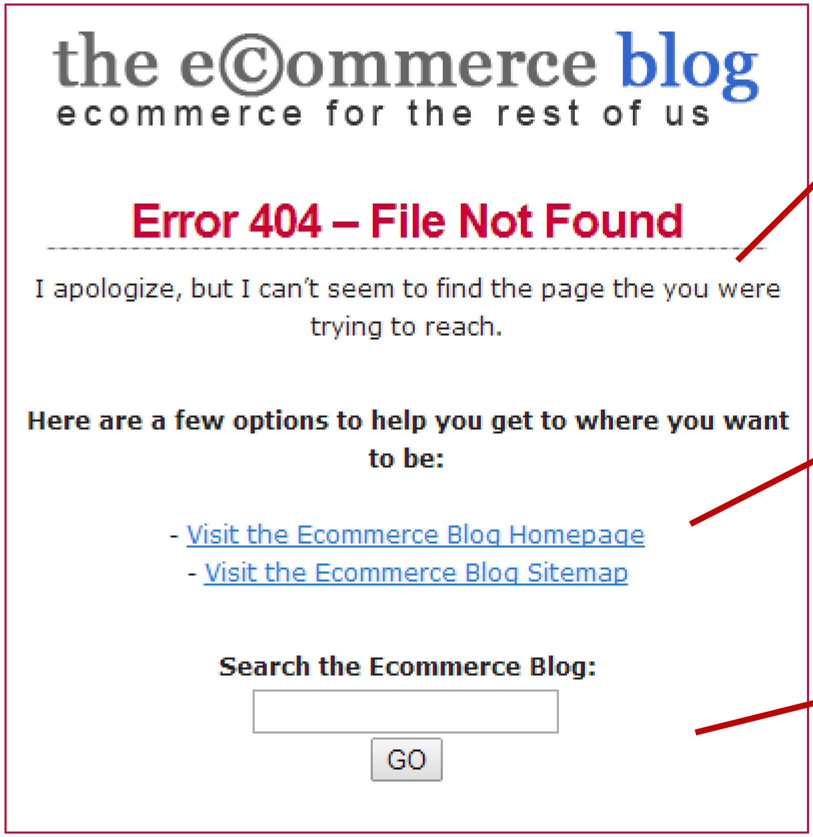


```
HTTP/1.1 404 Not Found Content-Type: text/html; charset=utf-8 Set-Cookie:
tmgioct=53e2[illegible]232dd9670135133660; expires=Sun, 04-Aug-2024 01:45:07 GMT; path=/; httponly Link: ;
rel=icon Cache-Control: max-age=300 P3P: CP="ALL ADM DEV PSAi COM OUR OTRo STP IND ONL" X-Tumblr-
User: bobpeers X-Tumblr-Pixel-0: http://www.tumblr.com/impixu?
T=1407375907&J=eyJ0eXBlIjoidXJsIiwidXJsIjoiaHR0cDpcL1wvYm9icGVlcnMuY29tXC90ZWNobmljYWxcLzQwNF
X-Tumblr-Pixel: 1 Link: ; rel=icon X-UA-Device: desktop Vary: X-UA-Device Date: Thu, 07 Aug 2014 01:45:07
GMT Connection: close
```

Response Code Header

Designing a Custom Error Page

Include in the body (content) area:



the e©ommerce blog
ecommerce for the rest of us

Error 404 – File Not Found

I apologize, but I can't seem to find the page the you were trying to reach.

Here are a few options to help you get to where you want to be:

- Visit the Ecommerce Blog Homepage
- Visit the Ecommerce Blog Sitemap

Search the Ecommerce Blog:

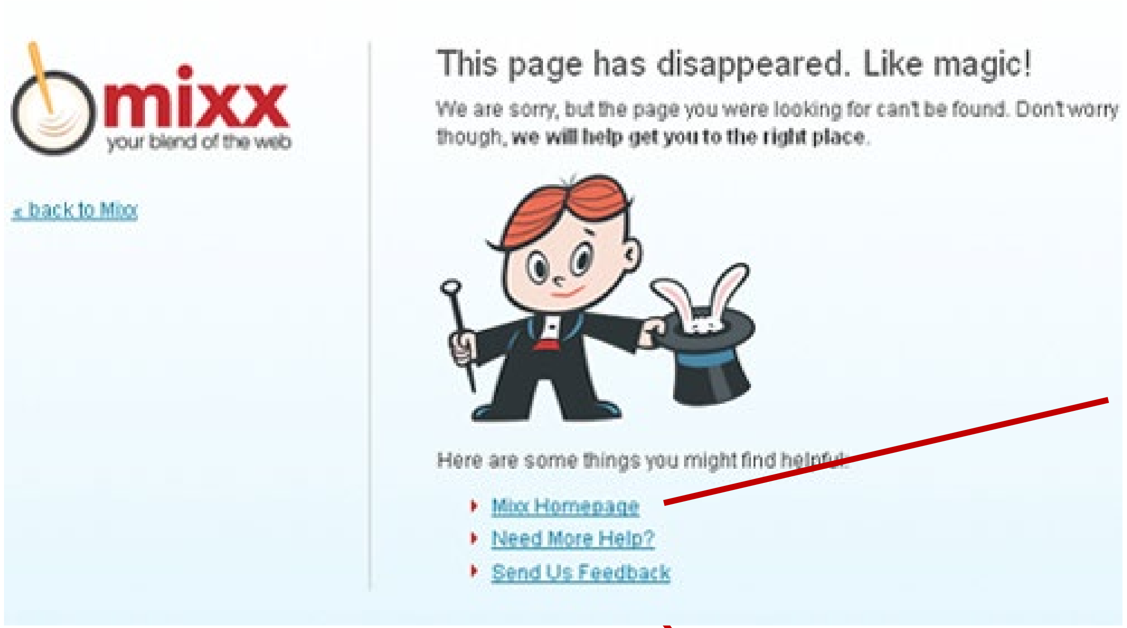
An Apology for the error

a list of possibly related links

a search box if you have website search enabled on the website.

Designing a Custom Error Page

Include in the body (content) area:



mixx your blend of the web

« back to Mixx

This page has disappeared. Like magic!

We are sorry, but the page you were looking for can't be found. Don't worry though, **we will help get you to the right place.**

Here are some things you might find helpful:

- Mixx Homepage
- Need More Help?
- Send Us Feedback

A link to your homepage.

a contact form (or link to) so the visitor can notify you of the error.

see another example

Setting Custom Error Pages

- In Web.config

```
<configuration>
<system.web>
   <customErrors mode="On" defaultRedirect="errors/GenericErrorPage.htm">
      <error statusCode="403" redirect="errors/NoAccess.htm"/>
      <error statusCode="404" redirect="errors/FileNotFound.htm"/>
   </customErrors>
</system.web>
</configuration>
```

`mode` attribute determines whether a visitor gets to see a detailed error page or not.

On - custom error page is visible to both client (remote) and server (local).

Off – client (visitor) will see the .NET technical error page (YSOD)

RemoteOnly – client see the custom-error page; but the YSOD will be displayed at the server side.

HTTP status code

TUNKU ABDUL RAHMAN UNIVERSITY OF MANAGEMENT AND TECHNOLOGY

Question

Refer to the statements below, write the necessary code in Web.config in order to show the specific custom-made error pages to the users.

- Display "**`Unauthorized.htm`**" page (error **`401`**) when an authorized access occurs.
- Display "**`NoSuchFile.htm`**" page (error **`404`**) when a requested file is not found.
- Display "**`ContactAdmin.htm`**" page when any other errors occur.

Basics of Debugging

- Process of finding and fixing bugs in your code.
- Working with bugs:
  - Setting breakpoints and Debug
  - Debugging Windows
  - Debugging JavaScript
  - Tracing

Setting Breakpoints

- We set a breakpoint by pressing F9 on the line of code where we want to execution to halt.
- When a breakpoint is hit during the debugging mode (F5), execution is stopped so that you can look at the code and gives us access to variables, controls, methods and much more.

Moving around in Debugged Code

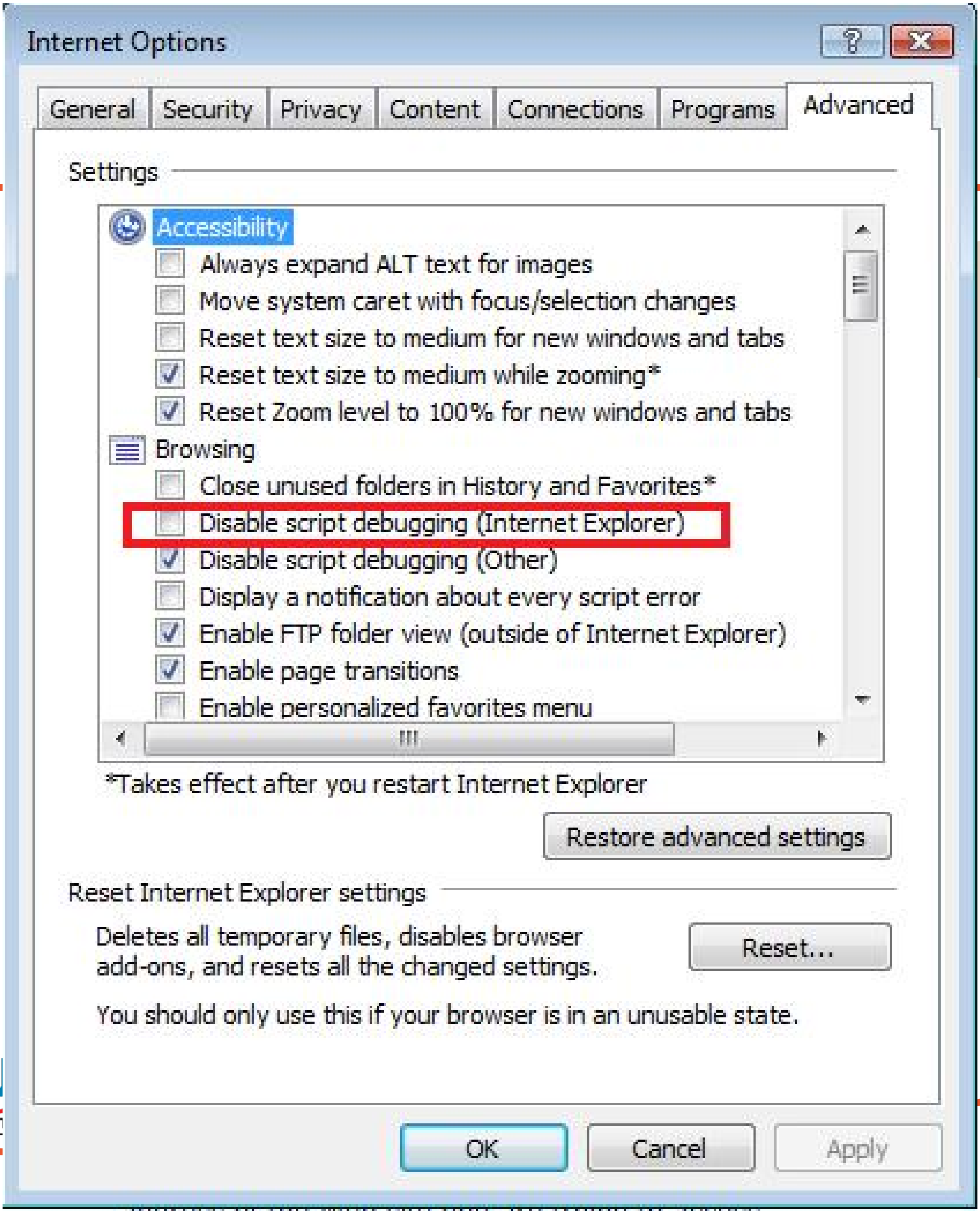
F5	Start Debugging
F11	Execute the current line and step into a method being called
F10	Execute the current line without stepping into the code that is being called
Shift + F11	Complete the code in the current method and return to the code that initially call it
Shift + F5	Stop debugging. Browser will be closed
Ctrl + Shift + F5	Restart the debugging process

Debugging Windows

- Watch Window – watch all variables
- Locals Window – watch local variables
- Breakpoints Window – overview of all breakpoints set
- Call Stack Window – order in which code has been executed or called
- Immediate Window – let you to execute code as if you had written it in a page. Use it to test expressions, see what functions return, etc.

Debugging JavaScript

- Debugging JavaScript with VS2008 requires you to use IE (won’t work on other browsers)
- Works on external js file and embedded JavaScript in the page.
- Breakpoints can be placed on the JavaScript code.



Internet Options

Settings

Accessibility
- [ ] Always expand ALT text for images
- [ ] Move system caret with focus/selection changes
- [ ] Reset text size to medium for new windows and tabs
- [✓] Reset text size to medium while zooming*
- [✓] Reset Zoom level to 100% for new windows and tabs

Browsing
- [ ] Close unused folders in History and Favorites*
- [ ] Disable script debugging (Internet Explorer)
- [✓] Disable script debugging (Other)
- [ ] Display a notification about every script error
- [✓] Enable FTP folder view (outside of Internet Explorer)
- [✓] Enable page transitions
- [ ] Enable personalized favorites menu

*Takes effect after you restart Internet Explorer

Reset Internet Explorer settings

Deletes all temporary files, disables browser add-ons, and resets all the changed settings.

You should only use this if your browser is in an unusable state.

Tracing

- Old day approach – create labels to display/trace the output one by one.
  - Cumbersome - need to create many labels
  - Ugly – may forget to remove the labels at the end
  - Performance affected by "extra" labels
- Tracing allows your pages, controls and code to write information to a central location called "Trace", which can then be shown in the browser.

Tracing (cont)

- To enable tracing, we need to set the Trace attribute in the Page directive to True:

  <%@ Page … Trace = “True” %>

- A long list of details will be shown at the bottom of page when you run a trace-enabled page.

Debugging Tips

Never Do This

- leave debug="true" in the Web.Config file.
  - Set it to false to ensure solid operation of your site
- Swallowing exceptions in a Catch block – leave the catch block empty.
  - It makes debugging difficult, can't prevent error

Debugging Tips (cont)

Do not assume your users know how to enter the right thing!

- Avoid exception handling if possible
  - Validation could be done so that can make exception handling unnecessary
  - Educate them, show them examples!

Next

CONFIGURATION AND OPTIMIZATION