INTRODUCTION TO EXCEPTION ERRORS

n exception error occurs when a problem is encountered during the processing of a JSP page. When an exception error occurs, an object that stores information about the error is created. Error handling is achieved by accessing the properties of this object.

The two main types of exception errors are RuntimeException errors and Error exception errors. RuntimeException and Error are the names of the classes that create objects when one of these types of errors is encountered.

RuntimeException errors are the most common type of exception error. These errors can arise from a variety of problems ranging from simple mathematical errors, such as dividing a number by zero, to more complex programming errors, such as specifying an incorrect type when attempting to cast an object.

Error exception errors occur when a problem related to the processing environment arises. For example,

RUNTIMEEXCEPTION ERRORS

📓 Untitled - Nol _ | 8 | × http://127.0.0.1:808 <u>File Edit Search H</u>elp <u>File Edit View Favorites Tools Help</u> <html> 🖕 Back 🔹 🤿 🕝 🙆 🖓 🔞 Search 📾 Favorites 🕉 History 🖏 🖓 🖓 🖓 📃 🤗 <head> <title>Item Costs</title> Address 🖉 http://127.0.0.1:8080/examples/costs.jsp 💌 🧬 Go 🛛 Links 🏾 </head> bodu> Error: 500 <h1>Item Costs by Product in Stock</h1> int itemAQuantity = 45; Location: /examples/costs.jsp int itemAGrosscost = 945; int itemAltemcost = itemÁGrosscost/itemAQuantitu: int itemBQuantity = 0; Internal Servlet Error: int itemBGrosscost = 1025; int itemBItemcost = itemBGrosscost/itemBQuantity; org.apache.jasper.JasperException: / by zero at org.apache.jasper.runtime.JspServlet\$JspServletWrapper.load(JspSe-<h3>Product A</h3> at org.apache.jasper.runtime.JspServlet\$JspServletWrapper.loadIfNece: There are <%= itemAQuantity %> units of Product A in stock.
 at org.apache.jasper.runtime.JspServlet\$JspServletWrapper.service(Js The item cost of Product A is <%= itemAltemcost %>. at org.apache.jasper.runtime.JspServlet.serviceJspFile(JspServlet.ja at org.apache.jasper.runtime.JspServlet.service(JspServlet.java:369) at javax.servlet.http.HttpServlet.service(HttpServlet.java:853) (h3)Product B(/h3) at org.apache.tomcat.core.ServletWrapper.handleRequest(ServletWrappe choice are (%= itemBQuantity %> units of Product B in stock.

The item cost of Product B is <%= itemBItemcost %>. at org.apache.tomcat.core.ContextManager.service(ContextManager.java at org.apache.tomcat.service.http.HttpConnectionHandler.processConne K/D> at org.apache.tomcat.service.TcpConnectionThread.run(SimpleTcpEndpoi: at java.lang.Thread.run(Unknown Source) </body> </html> 🕅 🕅 Internet 🖉 Done

A RuntimeException error is typically generated by an error in the code for a JSP page.

In this example, the JSP page contains code that divides a number by zero.

When the JSP page is displayed in a Web browser. a message appears displaying information about the RuntimeException error.

a problem with the Java Virtual Machine or a problem with a supporting file that is required by a JSP page will generate an Error exception.

Encountering an exception error does not necessarily mean that the processing of a JSP page must stop. Some errors can be handled within the code for the page. For example, you can create a try block and a catch block to handle exception errors that could potentially arise when a section of code is processed. This allows your code to recover from an exception error. For information about creating a try block and catch block, see page 176.

Some exception errors cannot be recovered from. For example, an Error exception error generated by a problem with the Java Virtual Machine cannot be fixed within the code for a JSP page. In such cases, the object can be accessed to determine valuable information about the error and how it may have been caused.



Although exception errors can variety of situations, the situati into three general categories.

Logical Errors

Logical errors are the most con and usually result when a prog validated parameters or values an action. An example of a cor is dividing a number by zero.

Standard Errors

Most of the methods that mak library contain code that will ge certain situations arise. For exa result if you use a number whe or assign a value to an array ele exist.

Program Errors

Problems with the Java Virtual server that processes Java code occur. Applications that are not beta releases, are more likely t

ERROR EXCEPTION ERRORS

Russians a			The second second
夕 Untitled - Notepad File Edit Search Help	_ 8 ×		ahttp://12
Eme Eme Search Help			<u>File E</u> dit
<hr/> kead>			🗧 😓 Back 🤜
			Address 🖨
<jsp:usebean <br="" class="formjavabean.processBean" id="processFormBeanId">scope="session"/></jsp:usebean>			j
<pre><jsp:setproperty name="processFormBeanId" property="*"></jsp:setproperty></pre>			Erro
(++++			LALL
<title>processForm</title> 			
 solution in the solution of the solution			Locat
<h1>Welcome,</h1>		>	Internal
<jsp:getproperty name="processFormBeanId" property="userName"></jsp:getproperty> 			Internal
<pre></pre>			org.apa
<pre><jsp:getproperty name="processFormBeanId" property="location"></jsp:getproperty></pre>			org.apa
			4
	-		@]Done
	<u></u>		Me Doube

An Error exception error is typically generated by a problem with the environment that processes a JSP page.

In this example, the JSP page contains code that attempts to access a JavaBean that does not exist.

HANDLING ERRORS

8

	_
arise from a wide ons can be grouped	
nmon type of errors rammer has not before performing nmon logical error	
e up the Java class enerate errors when imple, an error will ere a string is expected ement that does not	
Machine or a Web e can cause errors to t yet stable, such as o generate errors.	

127.0.0.1:8080/examples/process.jsp - Microsoft Internet Explorer
· → · ② ② △ ③ Search ⓐ Favorites ③ History □ • ● ◎ · ■ ♀
🖹 http://127.0.0.1:8080/examples/process.jsp 📃 🔗 Go 🗍 Links 🤉
ror: 500 ation: /examples/process.jsp
l Servlet Error:
<pre>ache.jasper.JasperException: Unable to load class formjavabean.process at org.apache.jasper.compiler.BeanRepository.getBeanType(BeanReposito at org.apache.jasper.compiler.GetPropertyGeneratory.generate(GetProper at org.apache.jasper.compiler.JspParseEventListener.generateAll(3) at org.apache.jasper.compiler.JspParseEventListener.endPageProcessin at org.apache.jasper.compiler.Compiler.compile(Compiler.java:183) at org.apache.jasper.compiler.SpServletJaspEventListener.endPageProcessin at org.apache.jasper.compiler.Compiler.compile(Compiler.java:183) at org.apache.jasper.runtime.JspServletJ3pServletUsTyper.loadIfMece: at org.apache.jasper.runtime.JspServletJ3pServletUsTyper.service(Js at org.apache.jasper.runtime.JspServletJspServletUsTyper.service(Js at org.apache.jasper.runtime.JspServletJserViceJspFile(JspServlet.java:365) at org.apache.jasper.runtime.JspServlet.service(JspServlet.java:365) at org.apache.jasper.runtime.JspServlet.service(JspServlet.java:365)</pre>
internet 🔮

When the JSP page is displayed in a Web browser, a message appears displaying information about the Error exception error.

CREATE A TRY BLOCK AND CATCH BLOCK

f a section of code in a JSP page may generate an exception error, you can create a try block and a catch block to handle the error.

A try block detects if an exception error has occurred in a section of code. To create a try block, use the keyword try and surround the code that may cause an exception error in braces.

A catch block contains the code that is executed when the try block detects an error. The catch block must immediately follow the try block. To create a catch block, use the keyword catch and enclose the code you want to execute in braces. The catch keyword is followed by a parameter enclosed in parentheses. The parameter specifies the class of the exception error and a name for the object that is created when the error occurs.

A catch block can only catch the type of exception error specified by the parameter. If the try block generates a

different type of exception error, the code in the catch block will not be executed.

When an exception error occurs in a line of code, the line of code is said to throw an error. When a line of code in a try block throws an error, the processing of code in the try block stops immediately and any remaining statements in the try block are not executed. The catch block catches the error thrown by the try block and processing continues on the first line of code in the catch block.

The code in a catch block can display a customized error message to notify a user that an error has occurred. The customized error message should be specific to the error and easy to understand. When a try block and catch block are not used to handle errors, Java generates cryptic error messages that can be difficult to comprehend.

Extra | As with any Java code, there are strict rules governing the scope of variables used in try and catch blocks. Variables declared in a try block are not available for use in the catch block. In the following example, the locationMessage variable is not available in the catch block.

Example: try {

String locationMessage = "determining item cost"; int itemCost = itemGrossCost / itemOuantity; out.print("Each item costs " + itemCost);

catch (ArithmeticException e)

out.print("Error has occurred at " + locationMessage);

The above code can easily be rewritten to change the scope of the locationMessage variable so that it is available to both the try and catch blocks.

String locationMessage = ""; try

> locationMessage = "determining item cost"; int itemCost = itemGrossCost / itemQuantity; out.print("Each item costs " + itemCost);

catch (ArithmeticException e)

out.print("Error has occurred at " + locationMessage);

CREATE A TRY BLOCK AND CATCH BLOCK

🔊 Untitled - Notepad	🔊 Untitled - Notepad	🖉 Untitled - Notepad	Item Costs - Microsoft Internet Explorer	
<u>F</u> ile <u>E</u> dit <u>S</u> earch <u>H</u> elp	<u>File Edit S</u> earch <u>H</u> elp	Eile Edit Search Help		* <u>0</u>
<html></html>	<html></html>	<html></html>	🗍 😓 Back 🔹 ⇒ ∞ 🔕 🚮 🕼 🐼 Search 📷 Favorites 🔇 History	
<head> <title>Item Costs</title></head>	<head> <title>Item Costs</title></head>	<pre><head> <title>Item Costs</title></head></pre>	Address & http://127.0.0.1:8080/examples/tryandcatch.jsp	▼ 🖉 Go 🛛 Links ≫
				*
 	<body></body>	 	Item Costs	
<h2>Item Costs</h2>	<h2>Item Costs</h2>	<h2>Item Costs</h2>	Error has occured: possible division by zero.	
<% int itemQuantity = 0;	<% int itemQuantity = 0;	<pre><% int itemQuantity = 0;</pre>	Inter has occured, pessible division by Lete.	
int itemGrossCost = 945;	int itemGrossCost = 945;	int itemGrossCost = 945;		
try	tru	tru		
int itemCost = itemGrossCost / itemQuantitu;	<pre>{ int itemCost = itemGrossCost / itemQuantity; }</pre>	{ int itemCost = itemGrossCost / itemQuantity;		
out.print("Each item costs " + itemCost);	out.print("Each item costs " + itemCost);	out.print("Each item costs " + itemCost);		
	<pre>> catch(ArithmeticException e)</pre>	<pre>} catch(ArithmeticException e)</pre>		
		{ out.print("Error has occured: possible division by zero. ")		
۶ ^۰	6 ²	<pre>but.print(error mas occured. \u00fbpossible uroision by zero.()b/) }</pre>	·	
 	 	*>		
				X
			御 Done	👔 Internet
1 Type the code 2 To create a try block,	4 To create a catch block, type 6 Type a n	ne for 7 Type the code you	8 Save the page with	The Web browser
that will generate enclose the code that will	catch () on the line immediately the object th		the .jsp extension and	displays the result of
an exception error. generate an exception error	following the try block. be created v		then display the JSP	creating a try block
in braces.	overation or		page in a Web browser.	and catch block.
	Between the parentneses, type thrown		puge in a web browsei.	und catch block.
On the line directly above	the class of the exception error that			
the opening brace, type try.	will be thrown by the try block.			177

HANDLING ERRORS

8

CATCH MULTIPLE EXCEPTION ERRORS

Ithough a try block may be capable of throwing different types of exception errors, a catch block can catch only one specific type of exception error. You can create multiple catch blocks to catch different types of exception errors.

When a catch block is created, the exception error class the block can handle is specified. If the try block throws an exception error of a different class, the code in the catch block will not be executed. A try block that contains a complex section of code may throw different types of exception errors. Creating multiple catch blocks allows a section of code to be executed for each type of exception error the try block throws.

The first catch block must immediately follow the try block and each subsequent catch block must be placed one right after the other. There cannot be any lines of code

between the try block and the first catch block. You also cannot place lines of code between any of the subsequent catch blocks.

When using multiple catch blocks, the order of the catch blocks is important. For example, since the Exception class is a superclass of the RuntimeException class, a catch block that uses the Exception class will catch most of the exception errors thrown by a try block. If you place a catch block that uses the Exception class before other catch blocks in your code, the code in the other catch blocks may never be executed. As a rule, you should place catch blocks that handle exception error subclasses before catch blocks that have a broader scope.

Extra

When using multiple catch blocks, you may want to add a finally block to your code. A finally block executes a section of code regardless of which catch block is processed. The finally block must immediately follow the last catch block.

Example:

for(int x = 0; x < itemQuanity.length; x++)</pre>

try

itemCost[x] = itemGrossCost[x] / itemQuanity[x];

catch(ArithmeticException e)

out.print("
An ArithmeticException error has occurred.");

catch(ArrayIndexOutOfBoundsException e)

out.print(" has occurred.");

finally

out.print("<hr>");

CATCH MULTIPLE EXCEPTION FRRORS

	ION LIKKORD				
🖉 Untitled - Notepad		🖉 Untitled - Notepad	📓 Untitled - Notepad 📃 🖪	🛛 🖾 Catch Multiple Exception Errors - Microsoft Internet Explorer	_ Ð ×
<u>File Edit Search H</u> elp		<u>File Edit Search H</u> elp	<u>Eile Edit S</u> earch <u>H</u> elp	<u>Eile E</u> dit <u>V</u> iew F <u>a</u> vorites <u>T</u> ools <u>H</u> elp	
<html></html>	<u> </u>	<pre><head></head></pre>	 kody>	▲ Arck + → + ③ ④ 🖓 ◎ Search 📷 Favorites 🎯 Histor	
<pre><head> <title>Catch Multiple Exception Errors</pre></td><td>(/title></td><td><title>Catch Multiple Exception Errors</title> </head></pre>	<pre>Item Costs</pre>	Address @ http://127.0.0.1:8080/examples/catcherrors.jsp	▼ @Go Links »		
		 body>			
<pre>\$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$</pre>	<pre>x++) i1 / itemOuanitu[x]:</pre>	<pre>Item Costs <% int[] item@rossCost = {280, 42, 44, 156}; int[] item@anity = {20, 0, 4, 12}; int[] itemQost = new int[3]; for(int x = 0; x < item@uanity.length; x++) { try { itemCost[x] = item@rossCost[x] / item@uanity[x]; out.print(" vr> { out.print(" br>ltem ' + x + " costs " + itemCost[x]);</pre>	<pre>(% int[] itemGrossCost = {280, 42, 44, 156}; int[] itemGuanity = {20, 0, 4, 12}; int[] itemCost = new int[3]; for(int x = 0; x < itemQuanity.length; x++) { try { itemCost[x] = itemGrossCost[x] / itemQuanity[x]; out.print("Chr>Item " + x + " costs " + itemCost[x]); } catch(ArithmeticException e) { out.print("Chr>An ArithmeticException error has occurred."); } catch(ArrayIndexOutOfBoundsException e) { out.print("Chr>An ArrayIndexOutOfBoundsException error"); out.print(" has occurred."); } }</pre>	Item Costs Item 0 costs 14 An ArithmeticException error has occurred. Item 2 costs 11 An ArrayIndexOutOfBoundsException error has occurred. * *	
1 Type the code that will generate multiple exception errors.	2 Type the code that creates a try block.	3 Type the code that creates a catch block to handle exception errors of the ArithmeticException class.	4 Type the code that creates a catch block to handle exception errors of the ArrayIndexOutOfBounds Exception class.	5 Save the page with the .jsp extension and then display the JSP page in a Web browser.	The Web browser displays the result of catching multiple exception errors.

HANDLING ERRORS



out.print("
ltem " + x + " costs " + itemCost[x]);

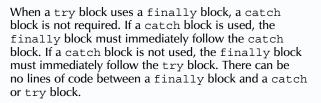
out.print("
An ArrayIndexOutOfBoundsException error");

hen a try block throws an exception error, the processing of code in the try block stops and any remaining statements in the try block are not executed. This can cause problems if the try block contains code that is important to the execution of your JSP page. To ensure important code is executed regardless of whether an exception error is thrown, you can place the code in a finally block.

To create a finally block, use the keyword finally and enclose the code you want to execute in braces. A finally block is useful for performing tasks that 'tidy up' a JSP page. For example, it is common for a finally block to contain code that closes a connection to a database or finishes writing data to a file.

There are strict rules governing the scope of variables used in try, catch and finally blocks. Variables declared in a try or catch block are not available for use in a finally block.

CREATE A FINALLY BLOCI



When a JSP page containing a finally block is processed, the code in the try block is executed first. If an error is thrown, the code in the appropriate catch block is then executed. The code in the finally block is executed last. The finally block is executed whether or not an exception error occurs and regardless of the type of exception error thrown.

Apply **L** Although the main purpose of a try block is to identify code that may generate an exception error, a try block can also be used with a finally block to save you time when typing code. For example, if a series of if statements will all have the same result, you can place the if statements in a try block and the result in a finally block. This saves you from having to type the same result for each if statement.

TYPE THIS:
try
{
if (winningScore > 10) return 10;
if (winningScore > 20)
return 20;
if (winningScore > 30)
return 30;
finally
{
out.print("The winning number
]
\mathbf{v}
RESULT:
The winning number has been dete

📓 Untitled - Notepad	💶 🗗 🔟 Untitled - Notepad		💐 Untitled - Notepad	_ 8 ×	Item Costs - Microsoft Internet Explorer	
<u>File Edit S</u> earch <u>H</u> elp	<u>File E</u> dit <u>S</u> earch <u>H</u> elp		<u>Eile E</u> dit <u>S</u> earch <u>H</u> elp			1 Alexandre
<html> <head> <title>Item Costs</title> </head></html>	<pre></pre>	-	<html> <head> <tile>Ite> Costs </tile></head></html>	<u> </u>	↓ → Back ▼ → ▼ ② ② △ ↓ ③ Search → Favorites ↓ Agdress ② http://127.0.0.1:8080/examples/finallyblock.jsp	History
<body> <h2>Item Costs</h2></body>	<body> <h2>Item Costs</h2></body>		<body> <h2>Item Costs</h2></body>		Item Costs	×.
<pre>(% String locationMessage = "Creating Variables"; int itemQuantity = 0; int itemFrosscost = 945; int itemCost;</pre>	<pre>{% String locationMessage = "Creating Va int itemQuantity = 0; int itemCrosscost = 945; int itemCost;</pre>	riables";	<pre><% String locationMessage = "Creating Variables"; int item@vascost = 945; int item@rosscost = 945; int item@cost;</pre>		An Error has occurred determining the cost. Items in stock = 0	
try { { itemCost = itemGrosscost/itemQuantity; }	try { itemCost = itemGrosscost/itemQua } catch(ArithmeticException e)	ntity;	try { itemCost = itemGrosscost/itemQuantity; } catch(ArithmeticException e)			
\$> 	out.print("An Error has occurred Finally	determining the cost. ");	<pre>{ out.print("An Error has occurred determining the cost.); finally { out.print("Items in stock = " + itemQuantitu); } }</pre>			
	<pre>%></pre>		<pre>> Complete read in Scone Pronouncesy, %> </pre>	-	환] Done	v 🖉
1 Type the code that will generate an exception error.2 Type the code that creates a try block.	at 3 Type the code that creates a catch block.	4 To create a finally block, type finally on the line immediately following the catch block.	5 Type the code you want to execute when the finally block is processed. Enclose the code in braces.		6 Save the page with the .jsp extension and then display the JSP page in a Web browser.	The Web browser displays the result of using a finally block.

HANDLING ERRORS

8



rmined.

REDIRECT TO AN ERROR PAGE

here are many types of exception errors that can be generated by the JSP pages in a Web site. Instead of trying to catch each specific type of exception error that could occur, you can configure the JSP pages to redirect to another page when an error occurs. The error page can be a JSP page or other type of Web document, such as an HTML document.

When an exception error occurs in a JSP page, the Web server stops processing the page and sends an error message to the Web browser to notify the client about the error. The type of exception error that occurs determines the information displayed in the error message. While the error information generated by the server may be useful to someone troubleshooting the JSP page, the information is usually not helpful to clients. Creating an error page allows you to determine the information that a client sees when an exception error occurs. For example, you may want to display a user-friendly page that provides clients with helpful instructions.

To redirect a JSP page to another page in the event of an exception error, you use the errorPage attribute of the page directive. For more information about the page directive, see page 74. The errorPage attribute takes the URL of the error page, enclosed in quotation marks, as its value. The URL of the error page must be a relative URL. This means that the JSP page and error page must be stored on the same Web server.

Multiple JSP pages can use the same error page. You must include the redirection instructions on each JSP page you want to use the error page.

Extra

On most Web servers, the default value for the autoFlush attribute of the page directive is true, which means that the buffer is set to automatically flush when it is full. When the buffer is flushed, information in the buffer is sent to a client's Web browser. If the buffer is flushed before a JSP page is redirected to the error page, an additional error will be generated. To avoid this, you can set the value of the autoFlush attribute to false when using the errorPage attribute.

Example

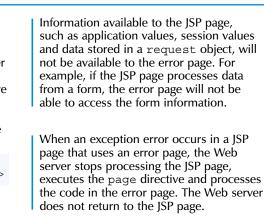
<@ page autoFlush="false" errorPage="error.jsp" %>

REDIRECT TO AN ERROR PAGE

🚭 Untitled - Notepad	🔊 Untitled - Notepad	💐 Untitled - Notepad	🖉 HTTP 404 Not Found - Microsoft Internet Explorer	_ 5 ×
<u>File Edit S</u> earch <u>H</u> elp	<u>File Edit Search Help</u>	<u>File Edit S</u> earch <u>H</u> elp	Eileditiew F_avoritesoolselp	
<pre></pre>	<pre></pre> <pre><</pre>	<pre></pre>	│ 🗢 Back 🔹 ⇒ 🚽 🙆 👔 🖓 │ 📿 Search 👔 Favorites 🔇 Hist	
<head></head>			Address 🖉 http://127.0.0.1:8080/examples/redirect.jsp	✓ interval → Go
<title>Error Message</title>	<html></html>	<html></html>		*
	<head> <title>Welcome</title></head>	<hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>	The JSP page you were viewing has generated an error.	_
<pre>vuluy></pre>				
The JSP page you were viewing has generated an error.	 	 		
(/bodu)	<pre>- </pre> <h1>Welcome to my Web site. </h1>	<pre></pre>		
 	(III) welcome to my web site. (/III)	(IT/WEICOME CO My WED SILE.(/IT/		
		<%		
	 	- int x = 1/0;		
	() IICHI)			
				-
			🙋 Done	S Internet
CREATE AN ERROR PAGE 2 Save the page on	REDIRECT A JSP PAGE 2 Between the quotation	n Type the code that	4 Save the page with	The Web browser
			the ice extension and	displays the result of
1 In a text editor, create the web server.		will generate an error.	the .jsp extension and	
the page you want to	1 On the first line of code the error page.		then display the JSP	redirecting to an error
display when an error	in a JSP page you want to		page in a Web browser.	page.
occurs.	redirect to an error page, type			
	<%@ page errorPage="" %>.			

HANDLING ERRORS

8



CREATE A DETAILED ERROR PAGE

v ou can create an error page that accesses detailed information about an exception error that has occurred in a JSP page. Accessing detailed information can help you troubleshoot the page. You can choose to simply display the detailed information about an exception error or you can log the information in a file or database.

When a JSP page generates an exception error of the Exception class, an exception object is created. The object holds information about the exception error. You can access the exception object in an error page to find detailed information about the error that occurred.

To make the exception object available to an error page, you use the isErrorPage attribute of the page directive. For more information about the page directive, see page 74. The isErrorPage attribute can have a value of either true or false. A value of true will make the exception object available to an error page. False is the default value of the isErrorPage attribute.

The getMessage method of the exception object can be called to access an error message that describes the type of error that has occurred. You can use an expression to display the information returned by the getMessage method. Some exception errors do not have an error message associated with them. In this case, the getMessage method will return a null value. For more information about the methods of the exception object, refer to the Java SDK documentation.

To redirect a JSP page to a detailed error page in the event of an error, you use the errorPage attribute of the page directive. The errorPage attribute takes the URL of the detailed error page, enclosed in quotation marks, as its value. The URL of the detailed error page must be a relative URL.

Web server with the .jsp

extension.

Extra |

The exception object is only available to the detailed error page. However, there are techniques you can use to make the information in the exception object available to other JSP pages. For example, in the detailed error page, you can use the setAttribute method of the session object to store the error message as a session value.

Example:

<% session.setAttribute("errorMessage", exception.getMessage()); %>

You may be able to use the getLocalizedMessage method of the exception object to access even more detailed information about an exception error. However, in most cases, the getLocalizedMessage method returns the same information as the getMessage method.

Example:

<%= exception.getLocalizedMessage() %>

CREATE A DETAILED ERROR PAGE

🔊 Untitled - Notepad	🖉 Untitled - Notepad
<u>Eile E</u> dit <u>S</u> earch <u>H</u> elp	<u>File Edit Search H</u> elp
<%@ page isErrorPage="true" %>	<%@ page isErrorPage="true" %>
<html></html>	<html></html>
<head></head>	<pre><head></head></pre>
<title>A Detailed Error Page</title> 	<pre><title>A Detailed Error Page</title> </pre>
<body></body>	
(body)	(body)
The JSP page you were viewing generated an error.	The JSP page you were viewing generated an error.
The type of error appears to be:	The type of error appears to be:
	<pre></pre>
	<pre><<= exception.getmessage() %></pre>
1 On the first line of	2 Type exception.getMessage() 3 Type the code that
code in the error page, type	where you want to access an displays the error message
<%@ page isErrorPage="true" %>	error message. on the detailed error pag
to access the exception object.	4 Save the page on the
- ,	1 Save the nage on the

🔊 Untitled - Notepad		🖉 A Detailed Error Page - Microsoft Internet Explorer	_ 8 ×
File Edit Search Help			# 1
<pre></pre>		← Back → → → ② ② ④ 🖓 ◎ Q Search 📷 Favorites ③	History 🖏 🛃 📃 🤶
<pre><html></html></pre>		Address 🖉 http://127.0.0.1:8080/examples/redirect.jsp	▼ 🖓 Go 🛛 Links ≫
<pre><head> <title>A Page with an Error</title> </head> </pre>	(The JSP page you were viewing generated an error.	×
<body> <h2>Welcome!</h2></body>		The type of error appears to be: / by zero	
<pre>(% int x = 1/0; %> </pre>		後] Done	S Internet
	2 Type the code that vill generate an error.	3 Save the page with the .jsp extension and then display the JSP page in a Web browser.	The Web browser displays the result of redirecting to a detailed error page.

HANDLING ERRORS

8



The toString method of the exception object can be used to display the class name of an exception error. The result of the toString method may also contain the information returned by the getMessage method.

Example:

<%= exception.toString() %>

Returns: java.lang.ArithmeticException: / by zero