Technical Errors

The following are technical errors in code or text which may impede understanding of the topics concerned. Typographical errors are listed later.

PAGE	ERROR	CORRECTION	COMMENTS
100	The second and sixth lines of the first paragraph contain the interface name `HttpRequest.`	The interface name should be `HttpServletRequest`.	
104	The third paragraph reads: "setting the content type or encoding only has an effect before the `getWriter()` method is first called."	`getWriter()` depends only on the character encoding, set directly via `setContentEncoding`, or indirectly as part of the content type or locale, for example in `setContentType("text/html;charset=UTF-8")` or via `setLocale`. The MIME content type (e.g. "text/html") may still be changed successfully after invoking `getWriter()`, but before the response is committed.	This paragraph will be re-worded in future editions. The key idea is that only the character encoding affects the `PrintWriter` obtained from `getWriter()`.
106	The `setStatus(int code)` method description contains the interface name `HttpResponse`.	The interface name should be `HttpServletResponse`.	
121	Q.6 Exhibit line 14 reads: `sendRedirect("/servlet2");`	It should read: `resp.sendRedirect("/servlet2");`	
130	The description of the `forward` method states: "This method should be called before the response has been committed, and before any calls to `getOutputStream()` or `getWriter()` or an `IllegalStateException` is thrown".	This would better read: "This method must be invoked before the response has been committed or an `IllegalStateException` is thrown. Recommended practise is to invoke this method before any calls to `getOutputStream()` or `getWriter()` as only one of these methods may ever be invoked during a single request or an `IllegalStateException` is thrown (see page 101). The resource being forwarded to may in general use one or other of these methods, but which one is unknown to the caller, so to use either in arbitrary calling code would be an unwise practise!"	The word 'should' indicates the advice regarding `getOutputStream()` and `getWriter()` is a *recommendation* and not authoratitive (as per the Servlet specification).
132	The paragraph before the note reads: "If we'd omitted the explicit `return` statement from above, the subsequent header modification would be executed and an `IllegalStateException` would be thrown since the response is committed".	As emphasised on page 105, attempting to change headers is ignored after the response has been committed; no exceptions are thrown. Therefore the `return` statement on page 132 is not necessary and the header modification in that code does nothing.	

169	`if(outBytes != null) {`
	`    return outWriter;`
	`}`
		`if(outBytes != null) {`
		`    return outBytes;`
		`}`
			The download bundle contains the correct code.

175	Exhibit for Q.10, line 10 reads: `if(email != null) {`	It should read: `if(param != null) {`	
176	Exhibit for Q.11, line 5 contains: `extends HttpResponseWrapper`	It should read: `extends HttpServletResponseWrapper`	
185	The first line of "The Session API" section reads "object of the `javax.servlet.http.HttpSession` class".	This should read "object of the `javax.servlet.http.HttpSession` interface".	
188	The heading at the top of the page is "The HttpActivationListener Interface"	Should read: "The HttpSessionActivationListener Interface"	
186-187	Description of `setMaxInactiveInterval` states: "If the `int` parameter is negative or zero, the timeout is infinite (i.e. The session will never timeout)".	This statement is only correct for negative integers. The case `setMaxInactiveInterval(0)` means the session will be invalidated at the end of the current request-response cycle in a manner equivalent to a 'time out'. `invalidate()` results in the session being invalidated immediately at the point in code where it is invoked.	It's a slightly subtle distinction and one to watch in practise (but very unlikely to be asked on the SCWCD exam).
191	Question 3 states "choose two".	It should state "choose three".	This follows from p.198 correction.
192	Question 6 states "choose three".	It should state "choose two".	This follows from p.198 correction.
198	Answer to Q3: option D incorrectly states that invoking `setMaxInactiveInterval(0)` means 'never time out'.	`setMaxInactiveInterval(0)` causes the session to be invalidated at the end of the current request-response cycle. It cannot be used to invalidate the request immediately like `invalidate()` does, but will still ultimately cause the session to be invalidated. Therefore the correct answers are B,D,E.	As stated in the book in parentheses, use of a negative value to 'never time out' is correct.
198	Answer to Q6: option C has no explanation.	Option C is in fact incorrect since `setMaxInactiveInterval(0)` causes the session to be invalidated at the end of the current request-response cycle; it cannot be used to configure a 'never time out' situation.	

199	Answer to Q10 says “throw an IllegalArgumentException if the session...”	This should read: “throw an IllegalStateException if the session...”	
206	Figure 9.6 has an OR between the `url-pattern` and `servlet-pattern` deployment descriptor elements.	Since only one or the other of the two elements may be declared, this would be better as an exclusive OR (XOR).	This is a future enhancement for clarity, and not an error in the book.
208	Figure 9.8 has an OR between the `servlet-class` and `jsp-file` deployment descriptor elements.	Since only one or the other of the two elements may be declared, this would be better as an exclusive OR (XOR).	This is a future enhancement for clarity, and not an error in the book.
209	In the second paragraph from the bottom: “The `<init-param>` element has exactly the same syntax as the `<filter>` element”	For clarity, this would better read: “The `<init-param>` element here has exactly the same syntax as the `<init-param>` for the `<filter>` element”	
213	Figure 9.13 has an OR between the `error-code` and `exception-type` deployment descriptor elements.	Since only one or the other of the two elements may be declared, this would be better as an exclusive OR (XOR).	This is a future enhancement for clarity, and not an error in the book.

219	Question 4 option B states:
	`<filter>`
	`  <filter-name>Auth Filter</filter-name>`
	`  <filter-class>MyAuthenticationFilter</name>`
	`</filter>`
		The closing tag `</name>` should in fact be `</filter-class>`, so that option B reads:
		`<filter>`
		`  <filter-name>Auth Filter</filter-name>`
		`  <filter-class>MyAuthenticationFilter</filter-class>`
		`</filter>`	

223	Question 13 option C uses a closing tag of the form: `</exception-type-type>`	This closing tag should be: `</exception-type>`	The answer supplied becomes valid once this change is made
243	The code references the method called 'int' on Random which doesn't exist.	This code should reference the method called nextInt, making the expression: `<%= new Random().nextInt(20) %>`	
251	The current list of JSP implicit objects omits “exception”.	For completeness, add to this list “exception (class java.lang.Throwable)” with the explanation: 'The exception object thrown by another page. Only available when this JSP declares the page directive isErrorPage=“true”'.	
260	In question 6, options D, E and F use invalid instantiation syntax; they are missing the `()` for the default constructor.	`new java.util.Date` should be `new java.util.Date()`	

272	`<jsp:scriptlet>request.getRemoteUser()</jsp:scriptlet>`	The semi-colon was omitted from the end of the scriptlet line. This line should have been: `<jsp:scriptlet>request.getRemoteUser();</jsp:scriptlet>`	This example is more instructive in practise if you use an expression rather than scriptlet, in which case it is correct to omit the semi-colon: `<jsp:expression>request.getRemoteUser()</jsp:expression>` Note that the software examples in the download bundle use an expression rather than a scriptlet.
272	The JSP namespace in the example is given as: `http://java.sun.com/jsp/Page`	The namespace should be: `http://java.sun.com/JSP/Page`	This is minor typographical mistake introduced correctly on p.271, and will not be tested on the exam.
273	`<jsp:scriptlet>request.getRemoteUser()</jsp:scriptlet>`	The semi-colon was omitted from the end of the scriptlet line. This line should have been: `<jsp:scriptlet>request.getRemoteUser();</jsp:scriptlet>`	This example is more instructive in practise if you use an expression rather than scriptlet, in which case it is correct to omit the semi-colon: `<jsp:expression>request.getRemoteUser()</jsp:expression>` Note that the software examples in the download bundle use an expression rather than a scriptlet.

274	The `<jsp:root>` in the example doesn't declare the mandatory 'version' attribute.	The opening tag should take the form:
		`<jsp:root xmlns:jsp="http://java.sun.com/JSP/Page"`
		`version="2.0"`
		`xmlns:c="http://java.sun.com/jsp/jstl/core">`
			This will not be tested in the exam. The 'version' attribute is only required on `<jsp:root>` and takes a default value if this root element is omitted.

275	`<jsp:expression>`*legal Java code*`</jsp:expression>`	This would be better written: `<jsp:expression>`*Java code as would be passed as the argument to System.out.print*`</jsp:expression>`	This makes the reasons for omitting the end-of-line semi-colon more obvious.

277	Question 3 contains `;` characters in the import statements. This causes a compilation error.	The `;` characters should be removed/ignored.	
314	Question 19 says “choose two” and option B is missing a closing brace `}`.	It should say “choose one” and option B should be: `${ pageContext.request.header[“Accept-Language”] }`	
319	Question 19 answer says both B and D are correct	The correct answer is D only (B is incorrect). Amend the explanation to read: “Option B is incorrect because `pageContext.request` is an object of type `ServletRequest` (and usually `HttpServletRequest`), but there is no zero-argument `getHeader()` method on that interface or class. Hence it cannot be searched using the [] operator.”	
340	Line 3 contains the variable called “rel_location”	This should be “rel_locn” as on p.339	
344	Question 5 has a <jsp:useBean> tag containing a beanName attribute but no type attribute.	There should also be a type attribute, so the entire line is: `<jsp:useBean id=“user” beanName=“org.mysite.Person” type=“org.mysite.Person” />`	This requirement is explained on p.328.
345	Question 8's first tag is missing the `type` attribute.	The answer given on p.351 is still correct, with the added explanation that the `type` attribute is required when using `beanName` and that it is missing (which would still cause a translation error).	The type attribute in this case should have the value `com.myshop.Item` or any superclass or implemented interface of that class.

348	Question 18 options D and E declare an invalid "path" attribute.	"path" should in fact be "page", making option D:
		`<jsp:include page="<jsp:expression>request.getAttribute("item")</jsp:expression>" />`
		and option E should read:
		`<jsp:include page="${dispatch}" />`
			Due to this error, the question currently has no correct answer, but the supplied answer is valid when these corrections are made.

349	Exhibit for Q4 has an illegal space in: `new com.mybeans. ConcreteConnection());`	The space is an error; this should read: `new com.mybeans.ConcreteConnection());`	
350	Question 3's explanation begins with “List is a superclass of ArrayList”.	It should read “ArrayList implements List”, since the latter is an interface not a class.	

363	The book states: "This sets the `itemTotal` property to `12` by calling `getItemTotal(12)`".	`getItemTotal(12)` should read `setItemTotal(12)`.	
363	`${ oldCountervalue + 1}`	Should read: `${ oldCounterValue + 1}`	
374	The example of using `<c:url />` to get a URL from appA to appB rewrites the URL into "something similar to:" `/appB/includes/banner.jsp;jsessionid=5F622D6`	According to the Glassfish v3 and Tomcat 5 source code, the `jsessionid` is not written into a foreign URL. It should read: `/appB/includes/banner.jsp`	This behaviour isn't specified in the Servlet or JSTL specifications; the correction is from common source code.
383	Exhibit for Q.4 contains the line: `<% 100/(i-1); %>`	This line should read: `<%= 100/(i-1) %>`	
392	Figure 15.1: The `setPageContext()` method is shown as being invoked after `setParent()`.	The figure should show `setPageContext()` being invoked *before* `setParent()`.	Both methods are setters whose implementations really should have no side effects – any actions other than setting the respective properties of the tag would be best placed in `doStartTag()`. So really it should not matter in which order these methods are invoked. Indeed, implementing either `TagSupport` or `BodyTagSupport` makes the order purely academic as you only usually override `doXxxTag()` methods.
396	Figure 15.2: The `setPageContext()` method is shown as being invoked after `setParent()`.	The figure should show `setPageContext()` being invoked *before* `setParent()`.	
400	Figure 15.3: The `setPageContext()` method is shown as being invoked after `setParent()`.	The figure should show `setPageContext()` being invoked *before* `setParent()`.	
403	Figure 15.4: The `setJspContext()` method is shown as being invoked after `setParent()`.	It is unclear from the JSP specifications or API documentation which order these are invoked in, so the figure may be correct or the order of execution of these methods may be inverted; it really isn't too important (see comments).	
418	The final code on the page contains a semi-colon (`;`) at the end of the scripting expression.	This semi-colon should not be present. The end should read: `.getName() %>`	As explained on p.249, this semi-colon would cause a fatal translation error.
432	The answer to Q.4 begins setParent(), setPageContext()	It should begin setPageContext(), setParent()	This follows from the error on p.396.
432	The answer to Q.5 begins: Instantiate, setParent(), setJspContext()	Either the answer is correct, or it should begin: Instantiate, setJspContext(), setParent()	This follows from the possible error on p.403.
437	In point 3 in the list at the bottom of the page, setParent() is said to be invoked before setJspContext().	Either this is correct, or the order should be inverted.	This follows from the possible error on p.403.
455	The first bullet item on the page contains "imports the `java.utill.Calendar` class"	Should read: "imports the `java.util.Calendar` class".	

474	The last paragraph contains “would be located at `com/mydomain/taglibs/MyClass.class`”	Should read: “ would be located at `com/mydomain/taglibs/MyTag.class`”	
523	One of the method signatures is: `Principle getUserPrinciple()`	This signature should be: `Principal getUserPrincipal()`	
527	Question 5 option D contains: `<auth-method>BASIC</method-name>`	The closing tag should be `</auth-method>` so the line reads: `<auth-method>BASIC</auth-method>`	

Typographical Errors

The following are minor typographical errors (spelling mistakes, misplaced punctuation etc.)

PAGE	ERROR	CORRECTION	COMMENTS
23	The answer to Q.7 is given as C	The correct answer should be E.	The explanation of answer E is correct.
108	Figure 5.4 has the constructor of the `ServletRequestAttributeEvent` class as `ServletRequestEvent`.	The constructor should be called `ServletRequestAttributeEvent`.	
123	The answer to Q.5 has the boxes misaligned.	The Line boxes on the right should be moved down one position relative to the options on the left. Hence Line A maps to the third option on th e left, Line B to the last option and Line C to the first option.	
124	The answer to Q.11 says “flushing the butter on line 23”.	Should read: “flushing the buffer on line 23”.	
139	The signature for the `HttpServletResponseWrapper` says it implements `HttpServletReponse`.	It should implement `HttpServletResponse`.	
140	The very last words on the page are “this header of not”.	Should read: “this header or not”.	
148	Question 14 option C, missing 'r': `javax.servlet.ServletResponseWrappe`	Should read: `javax.servlet.ServletResponseWrapper`	

193	Question 11 option A.	This would better read: “The request is forwarded to the other.jsp resource, and upon return the orderid session-scoped attribute is set to '1234'.”	
206	"requests they are serve are referred to"	Should read: "requests they serve are referred to"	
250	The first (second) line of the final paragraph ends (starts) with “should a they also”	Should read: “should they also”	
251	Each item in the list currently has the base datatype specified (e.g. “PageContext”)	For clarity these could be better as full class names also prepended by the type “class” or “interface”. E.g. “class javax.servlet.jsp.PageContext”).	
277	Question 5 option B has `%&lt;` at the end of the line.	This should be `%&gt;` at the end of the line.	This correction doesn't make the answer or the explanation given incorrect.
315	Option C for question 22 contains an * alongside the first line of the answer.	The * is a typographical mistake and should be ignored.	
371	Under the "Integer getEnd()" bullet the word 'end' is shown in bold.	The word 'end' should appear in `monospaced (code)` font.	
378	Question 5 reads “What is the effect of the following?”	It would be better as: “What is the effect of deploying and executing the exhibited JSP page?”	
394	The paragraph after the first list begins “We've use the”	Should begin with: “We've used the”	
401	The NOTE contains “amount of code requried”	Should read: “amount of code required”	
408	The third line on the page starts “Attributes, regardless of the which”	Should read: “Attribute, regardless of which”	
417	The last paragraph begins “This discrepency is”	Should begin with: “This discrepancy is”	
421	The first textual paragraph on the page ends “truely nested.”	Should read: “truly nested.”	
422	The second line from the bottom contains “tag file.. The”	Should read: “tag file. The” with only one full stop/period.	
429	Questions 13 and 16 contain “tag hander class” in the first line of the question.	Should read: “tag handler class”.	
432	Question 7 explanation contains “it is simply acts”	Should read: “it simply acts”.	

436	The third line of the second paragraph of the “Basic Tag Syntax” section contains “ `SimpleTag` hander class”	Should read: “`SimpleTag` handler class”	
439	The description for body-content contains “the contents should be evaulated”	Should read: “the contents should be evaluated”	
443	The first full paragraph on the page contains three occurrences of “sychronisation”	Should read “synchronisation”	
483	The second bullet item on the page begins “If it a relative”.	Should read: “If it is a relative”	
519	Line 4 contains “which it itself covered”	Should read “which is itself covered”	