

# CISC474 - Spring 2005

Advanced Web Technologies - Midterm Exam 2 5/10/2005

Name (as on official roster): \_\_\_\_\_

Name you go by in class (if different): \_\_\_\_\_

- DO NOT WRITE YOUR NAME ON ANY PAGE EXCEPT THIS ONE!
- You have 75 minutes
- You are permitted to use one 8.5 x 11 sheet of notes, (both sides) which you must turn in with your exam. Put your name on it. You won't get it back, so make a copy before you come to the exam. It *is* permitted for your notes to be a laser print or photocopy.
- **Pace Yourself!!!!**

Pay attention to the point values. When there are 10 minutes left, skim through and be sure you have at least written *something* for the questions that are worth many points.

- Read *all* the directions *carefully* on each problem.
- In your answers be as precise as possible. Be sure to answer the question that is *asked*. Don't just do a brain dump of everything you know.

## Section 1. Short Answer

1. (20 pts) On p. 302, your book indicates that there are two kinds of comments that can be in a JSP, but there are actually three, as illustrated by the code listing below. This listing for the file `JSPComments.jsp` contains three different comments, labelled A, B, and C, each with a different syntax.

Write an explanation that distinguishes among these three different syntaxes, making specific reference to A, B, and C. Be very *precise* in your answer: describe the proper terminology for each style of comment, and how it is processed differently by the system from the other two.

```
1 <html>
2 <body>
3
4 <!-- Comment A -->
5
6 <%-- Comment B --%>
7
8 <% /* Comment C */ %>
9
10 </body>
11 </html>
```

Clues to writing a complete answer: think about how far each comment will propagate through the system. Which parts of the system pass the comment through, and which parts strip it out? Which of these comments, if any, will be visible at the client side when you do “view source”? Think too about the relationship between a JSP and a servlet.

2. In web applications, The technique known as “URL rewriting” is typically employed as a “backup strategy” in case a particular “preferred” technique fails.

- (10 pts) What is the “purpose” of URL rewriting, and what is the preferred technique that is generally tried first?

- (10 pts) In a servlet, URL rewriting is accomplished via `response.encodeURL()`; while in a JSP the `<c:URL>` tag can be used. However, this `<c:URL>` tag doesn't come from standard HTML. Where does it come from? And what special thing do you have to “do” or “enable” in your webapp to be able to use this tag?

3. Consider the memo from the textbook, p. 314 (reproduced on the following page to refresh your memory).
- (15 pts) The memo bans the use of Scriptlets, Expressions, and Declarations in JSPs. Give an example of each (in the context intended by this memo, i.e. as they would appear in a JSP.)

If you need more room, continue on the space next to the memo on the following page.

- (10 pts) Explain what would motivate a CTO to write such a memo, and what JSP designers ought to be using instead of Scriptlets, Expressions and Declarations.

If you need more room, continue on the space next to the memo on the following page.

Interoffice Memo from the CTO

---

URGENT

Effective immediately, anyone caught using scriptlets, expressions, or declarations in their JSP code will be suspended without pay until such time as it can be determined whether the programmer was fully responsible or simply trying to maintain some OTHER idiot's code.

If, in fact, the determination is made that the programmer is, in fact, responsible, the company will go ahead and, in fact, terminate the employee.

---

Rick Forester  
Chief Technology Officer

---

"Remember: there is no "I" in TEAM."

"Write your code as if the next guy\* to maintain it is a homicidal maniac who knows where you live."

[\*Note to HR: we use "guy" in its non-gender specific form.]

4. (20 pts) Albert Einstein is often quoted as having said: You do not really understand something unless you can explain it to your grandmother. This question is in that spirit.

Your friend “Chris” has asked for your help. Chris is a smart businessman, but is not particularly tech-savvy. Chris wants to hire someone to do some maintenance on the web site for his company. He knows that the current site is based on “JSPs” but he doesn’t really know what those are. He tried to hire you, but your work load is too much at the moment; you know you don’t really have time to help him out. However, you do agree to sit in on the interviews of the web developers who have applied for the job.

The first person to apply has a really nice looking portfolio. The web sites he says he developed look great. Your friend is very impressed as he looks through the sites the candidate shows him.

So, your friend asks you if you have any questions for the candidate. You start with an easy one, just to break the ice: “Tell me, how did you learn HTML?”

He says, “well, to be honest, I never took a formal class. I just did a ‘view source’ on a bunch of web pages that I thought looked good, and sort of figured out how they worked, and picked up HTML that way. As you can see, my web pages look pretty snazzy, even though I never took a formal class. Here see, you can do ‘view source’ on any web page and read how it is coded.” At this the candidate shows a view source, and shows your friend how the tags line up with some of the elements on the rendered page. Your friend is getting more and more impressed.

You reply: “Very clever. Tell me though, my friend also needs someone who can maintain Java Server Pages... you know JSPs. You’ve shown us a bunch of JSP based sites as part of your portfolio. Did you create those sites too?”

The candidate replies, proudly: “Yes, I sure did”.

You respond: “How did you learn JSP?”

He replies: ” Exactly the same way... just found a bunch of sites that used JSPs, and then did “view source” and figured out they worked.”

You respond: “Really? You learned not just static HTML, but also *JSPs* by doing ‘view source’? Are you sure?”

He replies: “Yup, sure did. Took me a while to figure it out the JSP part, but eventually I got it.”

You reply: “Ah, very good... well I think that’s all we need to know.”

Your friend is all smiles, and you and he both thank the candidate for stopping by. Once the candidate leaves the room, you turn to your friend and say: “That candidate is a bald faced liar, and technically incompetent. Under no circumstances should you hire him.”

Your friend is taken aback. “That’s a pretty serious charge. How do you know this after only asking him four questions?”

What do you tell your friend?

**Note:** If you understand how JSPs work, it pretty simple to work out that the candidate is lying, and furthermore that he is not technically competent enough to realize that the lie is easy to spot.

However, for this question, a quick explanation in technical terms is NOT sufficient for full credit... having the technical facts correct will only earn you half the points.

For full credit, you'll be graded not only on the technical content of your question, but how well you explain it in terms that a non-technical person can understand. Keep in mind that your friend won't understand terms like "compiler", "container", "source code", etc. Your friend isn't stupid... he's a very successful businessman. But you won't communicate with him if you talk in "geek speak". And it is important that he fully understands... "trust me, I'm an expert" is not going to cut it here.

Let's assume, to make things easier, that your friend did take one computer class in programming in BASIC a zillion years ago, and vaguely remembers things like a "for loop" and "input" and "output". So you can use that as a point of reference. He's also seen the HTML code... the candidate did a view source, and showed him how the tags like <TABLE> correspond with the table on the page. He also understands that a computer has files on it, and that the files are stored on the hard disk. He sort of gets, too, that information on web pages comes from some place called a "server" out there on the network somehow, and makes its way to his computer, though he is very hazy on the details.

That's all you have to go on. Good luck!

There is extra space on the next page in case you need it.

Extra space in case you need it

5. A designer of an XML-based application might choose to use a DTD or to use XML Schema, and place a reference to one or the other in the XML document.

- (5 pts) Whether the reference is to a DTD or a Schema, the purpose is the same. What is the purpose of a DTD or Schema?

An accurate one-sentence explanation is enough. Don't write a book.

- (5 pts) List an advantage of using a DTD over an XML Schema.

An accurate one-sentence explanation is enough. Don't write a book.

- (5 pts) List an advantage of using an XML Schema over a DTD.

An accurate one-sentence explanation is enough. Don't write a book.

Total Points: 100

# CISC474 - Spring 2005

Advanced Web Technologies - Midterm Exam 2 5/10/2005

Name (as on official roster): \_\_\_\_\_

Name you go by in class (if different): \_\_\_\_\_

- DO NOT WRITE YOUR NAME ON ANY PAGE EXCEPT THIS ONE!
- You have 75 minutes
- You are permitted to use one 8.5 x 11 sheet of notes, (both sides) which you must turn in with your exam. Put your name on it. You won't get it back, so make a copy before you come to the exam. It *is* permitted for your notes to be a laser print or photocopy.
- **Pace Yourself!!!!**

Pay attention to the point values. When there are 10 minutes left, skim through and be sure you have at least written *something* for the questions that are worth many points.

- Read *all* the directions *carefully* on each problem.
- In your answers be as precise as possible. Be sure to answer the question that is *asked*. Don't just do a brain dump of everything you know.

\*\*\*\*\*Answer Key\*\*\*\*\*

Section 1. Short Answer

1. (20 pts) On p. 302, your book indicates that there are two kinds of comments that can be in a JSP, but there are actually three, as illustrated by the code listing below. This listing for the file `JSPComments.jsp` contains three different comments, labelled A, B, and C, each with a different syntax.

Write an explanation that distinguishes among these three different syntaxes, making specific reference to A, B, and C. Be very *precise* in your answer: describe the proper terminology for each style of comment, and how it is processed differently by the system from the other two.

```
1 <html>
2 <body>
3
4 <!-- Comment A -->
5
6 <%-- Comment B --%>
7
8 <% /* Comment C */ %>
9
10 </body>
11 </html>
```

Clues to writing a complete answer: think about how far each comment will propagate through the system. Which parts of the system pass the comment through, and which parts strip it out? Which of these comments, if any, will be visible at the client side when you do “view source”? Think too about the relationship between a JSP and a servlet.

- Answer:**
- (a) A is an HTML style comment. It will be passed through the JSP system just like any other HTML markup. It is the only one that will be visible in the “view source” on the client side. The web browser is the part of the system that strips it out.
  - (b) B is a JSP style comment. It is stripped out as the code is turned into Java servlet, and is never seen again. It is visible only to someone reading the JSP source.
  - (c) C is a Java style comment inside a scriptlet. It is copied into the Java source code that is generated from the scriptlet. It is stripped out by the Java compiler. It will not be seen at the client side, just like the Java code is not seen at the client side.

2. In web applications, The technique known as “URL rewriting” is typically employed as a “backup strategy” in case a particular “preferred” technique fails.

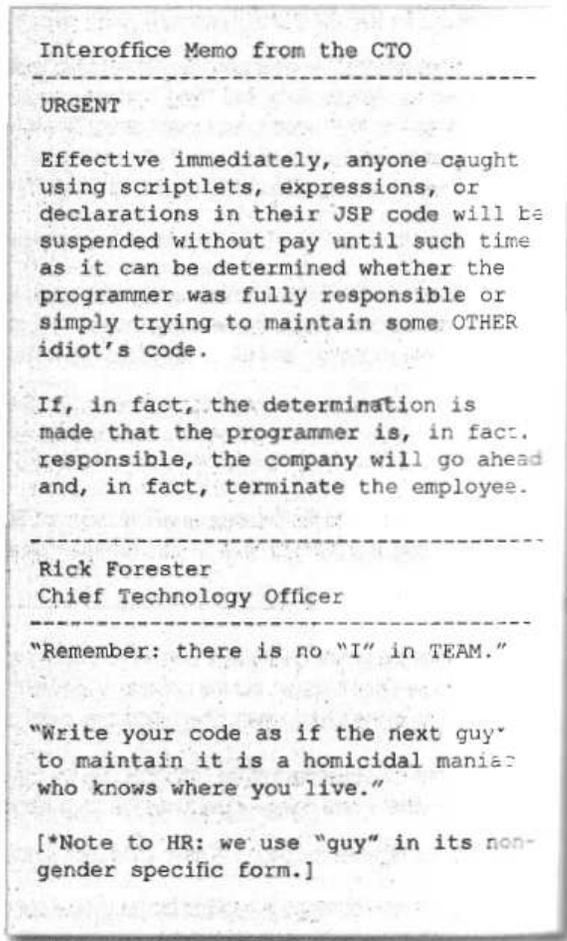
- (10 pts) What is the “purpose” of URL rewriting, and what is the preferred technique that is generally tried first?
- (10 pts) In a servlet, URL rewriting is accomplished via `response.encodeURL()`; while in a JSP the `<c:URL>` tag can be used. However, this `<c:URL>` tag doesn’t come from standard HTML. Where does it come from? And what special thing do you have to “do” or “enable” in your webapp to be able to use this tag?

**Answer:** • URL rewriting is a backup to cookies for session management.

- The `<c:URL>` tag comes from the JSTL. In Tomcat, you have to copy the `jstl.jar` and `standard.jar` into your `WEB-INF/lib` directory to be able to use such tags.

3. Consider the memo from the textbook, p. 314 (reproduced on the following page to refresh your memory).
- (15 pts) The memo bans the use of Scriptlets, Expressions, and Declarations in JSPs. Give an example of each (in the context intended by this memo, i.e. as they would appear in a JSP.)  
If you need more room, continue on the space next to the memo on the following page.
  - (10 pts) Explain what would motivate a CTO to write such a memo, and what JSP designers ought to be using instead of Scriptlets, Expressions and Declarations.

If you need more room, continue on the space next to the memo on the following page.



- Answer:**
- Scriptlets, Expressions and JSPs:
    - Scriptlets are bits of Java embedded in a JSP. They appear between `<%` and `%>` tags.
    - Expressions are Java Expressions embedded in a JSP that get passed into `out.println` and placed in the response stream. They appear between `<%=` and `%>` tags.
    - Declarations are declarations of instance variables for the Servlet class that is created from the JSP by the Container. They appear between `<%!` and `%>` tags.
  - A CTO would write such a memo because Java code inside a JSP is difficult to test and maintain. There may be legacy JSPs that contain scriptlets, etc., but new code development should use the Expression Language (EL) and tag libraries (e.g. the JSTL). This keeps Java code in Java files, and keeps web files containing only things that look like tags. The EL and tag libraries can easily be tied to Java classes, and are much more easily understood by Web designers that are used to HTML like syntax, not program syntax.

4. (20 pts) Albert Einstein is often quoted as having said: You do not really understand something unless you can explain it to your grandmother. This question is in that spirit.

Your friend “Chris” has asked for your help. Chris is a smart businessman, but is not particularly tech-savvy. Chris wants to hire someone to do some maintenance on the web site for his company. He knows that the current site is based on “JSPs” but he doesn’t really know what those are. He tried to hire you, but your work load is too much at the moment; you know you don’t really have time to help him out. However, you do agree to sit in on the interviews of the web developers who have applied for the job.

The first person to apply has a really nice looking portfolio. The web sites he says he developed look great. Your friend is very impressed as he looks through the sites the candidate shows him.

So, your friend asks you if you have any questions for the candidate. You start with an easy one, just to break the ice: “Tell me, how did you learn HTML?”

He says, “well, to be honest, I never took a formal class. I just did a ‘view source’ on a bunch of web pages that I thought looked good, and sort of figured out how they worked, and picked up HTML that way. As you can see, my web pages look pretty snazzy, even though I never took a formal class. Here see, you can do ‘view source’ on any web page and read how it is coded.” At this the candidate shows a view source, and shows your friend how the tags line up with some of the elements on the rendered page. Your friend is getting more and more impressed.

You reply: “Very clever. Tell me though, my friend also needs someone who can maintain Java Server Pages... you know JSPs. You’ve shown us a bunch of JSP based sites as part of your portfolio. Did you create those sites too?”

The candidate replies, proudly: “Yes, I sure did”.

You respond: “How did you learn JSP?”

He replies: ” Exactly the same way... just found a bunch of sites that used JSPs, and then did “view source” and figured out they worked.”

You respond: “Really? You learned not just static HTML, but also *JSPs* by doing ‘view source’? Are you sure?”

He replies: “Yup, sure did. Took me a while to figure it out the JSP part, but eventually I got it.”

You reply: “Ah, very good... well I think that’s all we need to know.”

Your friend is all smiles, and you and he both thank the candidate for stopping by. Once the candidate leaves the room, you turn to your friend and say: “That candidate is a bald faced liar, and technically incompetent. Under no circumstances should you hire him.”

Your friend is taken aback. “That’s a pretty serious charge. How do you know this after only asking him four questions?”

What do you tell your friend?

**Note:** If you understand how JSPs work, it pretty simple to work out that the candidate is lying, and furthermore that he is not technically competent enough to realize that the lie is easy to spot.

However, for this question, a quick explanation in technical terms is NOT sufficient for full credit... having the technical facts correct will only earn you half the points.

For full credit, you'll be graded not only on the technical content of your question, but how well you explain it in terms that a non-technical person can understand. Keep in mind that your friend won't understand terms like "compiler", "container", "source code", etc. Your friend isn't stupid... he's a very successful businessman. But you won't communicate with him if you talk in "geek speak". And it is important that he fully understands... "trust me, I'm an expert" is not going to cut it here.

Let's assume, to make things easier, that your friend did take one computer class in programming in BASIC a zillion years ago, and vaguely remembers things like a "for loop" and "input" and "output". So you can use that as a point of reference. He's also seen the HTML code... the candidate did a view source, and showed him how the tags like <TABLE> correspond with the table on the page. He also understands that a computer has files on it, and that the files are stored on the hard disk. He sort of gets, too, that information on web pages comes from some place called a "server" out there on the network somehow, and makes its way to his computer, though he is very hazy on the details.

That's all you have to go on. Good luck!

There is extra space on the next page in case you need it.

## Extra space in case you need it

**Answer:** The technical explanation is that when you do a “view source” on a JSP, you aren’t seeing the source code of the JSP, but rather the HTML that is “computed” by that JSP. So it is impossible to learn how to code a JSP from doing “view source”... you never actually get to see the JSP code, only the computed HTML. That technical explanation is worth half credit.

A non-technical explanation might proceed as follows:

There are two types of web pages. Some web pages are “static”—every time you look at them, they are the same, while other web pages are dynamic—every time you look at them they are different.

The static ones have HTML codes that just sit there on the hard disk on the server. When you ask for the page, the server gives you that HTML code, exactly. When you do a view source, you see exactly what was on the server. So, it is reasonable that a really clever person MIGHT be able to learn static HTML coding from doing lots of view source.

Dynamic pages, though, come from a computer program that is written in a computer language other than HTML. It is called JSP, and it is actually a mixture of a couple of different languages.. HTML plus a language called Java. Java is a lot like the BASIC programs you wrote back in school. Every time you go to the dynamic page, the program runs—on the server, not on your local machine—and spits out some HTML. That HTML is sent to your computer and displayed. You can do a view source on your local machine, but if you do, you are only seeing what the program spit out. You are NOT seeing the instructions in the program itself.. the JSP instructions. Those are on the server, and are hidden from view. There is no way to get access to them over the network unless you actually break into the server. But your candidate never claimed he has access to the server... he said that he learned JSP from doing “view source”. You definitely CAN’T see them with a “view source”.

So this programmer claims to have learned JSPs by doing view source. But that is impossible. So he is lying. Furthermore, he doesn’t even realize how easy his lie is to spot.. anyone who did understand how JSPs work would realize how stupid it is to claim that you can learn JSP coding from doing a view source. That’s why I claim that he is technically incompetent.

So sorry. Hope your next candidate is better!”

5. A designer of an XML-based application might choose to use a DTD or to use XML Schema, and place a reference to one or the other in the XML document.

- (5 pts) Whether the reference is to a DTD or a Schema, the purpose is the same. What is the purpose of a DTD or Schema?

An accurate one-sentence explanation is enough. Don't write a book.

- (5 pts) List an advantage of using a DTD over an XML Schema.

An accurate one-sentence explanation is enough. Don't write a book.

- (5 pts) List an advantage of using an XML Schema over a DTD.

An accurate one-sentence explanation is enough. Don't write a book.

**Answer:** • The purpose is to provide a means to specify the structure of the XML document, and validate the XML document against that structure.

- DTD syntax is more compact for specifying simple structures.
- XML Schema provides a richer syntax for restricting the datatypes of XML elements.

Total Points: 100