UNIVERSITY OF TORONTO
Faculty of Arts and Science
Midterm Exam – October 2005
CSC 309 H1 F
Instructor – Eyal de Lara
Duration – 1 hours
Examination Aids: One single-sided hand-written 8.5”x11” page containing notes

NAME _____

STUDENT NUMBER _____

PART I (Short Answers): _____ /40
PART II (Programming): _____/60
Total: _____/100

Part I of this examination is a series of short-answer questions.

Part II of this examination is a series of programming exercises.

Answer all questions in the spaces provided on this examination paper. There is no need to use more space than is provided. If you must, then use the back of the examination paper and so indicate in your answer.

General Advice:

- Skim through the entire exam before beginning your detailed work, to get a sense of where best to spend your time; if you get stuck on one question, go on to another and return to the difficult question later.
- Show your work, not just the final answer. Partial credit will be granted in cases that demonstrate correct reasoning, even if an error leads to an incorrect final result.
- For the programming exercise, if unsure of API details or language syntax details, try your best and include a comment indicating what you are attempting to achieve.

Good luck!

Part I – Short Questions

1.- [10 points] The key feature of the World Wide Web is that it makes possible the standardized exchange of information. Name the key protocol and key 3 standards that made the Web possible (beyond what was already provided by the Internet).

2.- [10 points] Give two examples of constrains that cannot be expressed using a DTD.

3.- [10 points] It is often the case that functionality can be implemented using either JavaScrip or a Java Applet. Give one example of functionality that:

- Cannot be implemented (or would be extremely cumbersome to implement) in JavaScript, but can be easily implemented with a Java Applet.

- Cannot be implemented with a Java Applet, but can be implemented with JavaScript.

4.- [10 points] Give 3 examples of limitations that browsers typically place over both JavaScript and Java Applets for security reasons.

Part II - Programming Question

5.- [60 points] This is a complex programming question. Read the following instructions **carefully** before you start to answer. A breakdown of the grade (based on implemented functionality) is provided at the bottom of the next page.

Develop a Web-based application for accesing an online music collection. Your application should use two frames as show in Figure 1. The left frame should show an index of available music albums . Users select an album by clicking on the respective link in the index. A list of available songs is then displayed on the left frame. To play a song, a user just clicks of the link. The site then uses a provided Java Applet to download and play the song.

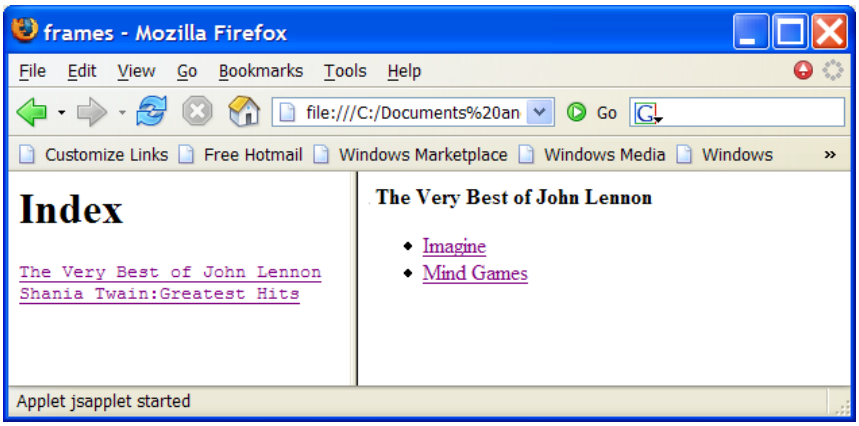

Figure 1 Music Album Catalog

Assume that information about available music albums is stored in a XML file called mymusic.xml, which has the following content:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE Catalog SYSTEM "music.dtd">
<Catalog>
  <Album id="cd1">
    <Title>The Very Best of John Lennon</Title>
    <Tracks>
      <Track src="http://songs.com/d1song1.mp3">Imagine</Track>
      <Track src="http://songs.com/d1song2.mp3">Mind Games</Track>
    </Tracks>
  </Album>
  <Album id="cd2">
    <Title>Shania Twain:Greatest Hits</Title>
    <Tracks>
      <Track src="http://songs.com/d2song1.mp3">Forever and For Always</Track>
      <Track src="http://songs.com/d2song2.mp3">I'm Gonna Getcha Good</Track>
          ……. <!— more tracks -- >
    </Tracks>
  </Album>
          ……. <!—more albums -- >
</Catalog>
```

Figure 2 mymusic.xml

Further, assume that Figures 2 and 3 show the code for the main page (the frameset) and the page that is loaded on the rigth frame, respectively.

```
<html>
  <head>
    <title>frames</title>
  </head>
  <frameset cols="30%, 70%">
      <frame src="catalogindex.html"/>
      <frame name="songs" src="album.html" />
  </frameset>
</html>
```

Figure 3 Frameset

```
<html>
<head>
  <script type="text/javascript">
     function play(songURL)  {
       var ap = document.getElementById("jsapplet");
       ap.play(songURL);
       return false;
     }
  </script>
</head>
<body>
  <applet id="jsapplet" code="MP3Player.class" height="1" width="1" />
  <strong id="albumtitle"> <!--  Dynamic: album title goes here  -->  </strong>
  <ul id="albumtracks">
    <!--  Dynamic: album track information goes here  -->
  </ul>
</body>
</html>
```

Figure 4 album.html

Finally, to play a song, use the JavaScript function playSong defined in album.html, which takes as its single argument the URL of the song.

Your have 2 tasks for this questions:

1) [30 points] Develop the index page (the left frame in Figure 1).
2) [30 points] Dynamically generate the album-specific information that appears in the rigth frame of Figure 1. For this purpose, use DOM commands to add content to the two locations in Figure 2 that are flaged with comments that start with <!—Dynamic -->.

Hint:

When a user clicks on a link, the event *onclick* is triggered. If the onclick event executes JavaScript code that returns *false*, then the browser does not follow the hyperlink and does not load the page.

Space for answer to question 5

Space for answer to question 5