

Université d'Ottawa
Faculté de génie

École d'ingénierie et de
technologie de l'information



uOttawa

L'Université canadienne
Canada's university

University of Ottawa
Faculty of Engineering

School of Information
Technology and Engineering

Structures, techniques et normes du Web (CSI 3540)

EXAMEN FINAL

Instructeur: Marcel Turcotte

Avril 2010, durée: 3 h

Identification

Nom, prénom : _____

Numéro d'étudiant : _____ Signature : _____

Consignes

1. Livres fermés ;
2. Sans calculatrice ou toute autre forme d'aide ;
3. Répondez sur ce questionnaire, utilisez le verso des pages si nécessaire, mais vous ne pouvez remettre aucune page additionnelle ;
4. Écrivez lisiblement, votre note en dépend ;
5. Commentez vos réponses ;
6. Ne retirez pas l'agrafe.

Barème

Question	Maximum	Résultat
1	9	
2	12	
3	15	
4	12	
5	12	
6	12	
7	10	
8	18	
Total	100	

Question 1 : JavaScript

Suite à l'exécution des énoncés JavaScript (ECMAScript) qui suivent, quel sera le résultat affiché par l'appel à **window.alert**.

```
1  var i=0;
2  var j=1;
3  var k=2;
4
5  var o1 = new Object();
6  o1.data = "Data o1";
7
8  var o2 = o1;
9  o2.data += "Data o2";
10
11 var o3 = new Object();
12 o3.data = "Data o3";
13
14 var o4 = new Object();
15
16 function test( param1, param2, param3 ) {
17     var j;
18
19     param1=5;
20     j=7;
21     k=3;
22     param2.data = "Data param2";
23     param3.data = "Data param3";
24     param2 = param3;
25
26     return param2;
27 }
28
29 function TestObject1( value ) {
30     this.data = value;
31 }
32
33 function someMore() {
34     return this.constantString;
35 }
36
37 function TestObject2(value) {
38     var o = new Object();
39     o.constantString = "this must not change";
40     o.data = someMore();
41     return o;
42 }
```

```
43 o4 = test( i, o2, o3);
44 var o5 = new TestObject1( "o5" );
45 var o6 = TestObject2( "o6" );
46
47 window.alert( " i = " + i +
48               "\n j = " + j +
49               "\n k = " + k +
50               "\n o1.data = " + o1.data +
51               "\n o2.data = " + o2.data +
52               "\n o3.data = " + o3.data +
53               "\n o4.data = " + o4.data +
54               "\n o5.data = " + o5.data +
55               "\n o6.data = " + o6.data() );
```

Question 2 : Programmation côté client

Si vous deviez concevoir la partie client d'une application Web, quels sont les mécanismes auxquels vous auriez accès afin d'accéder aux éléments du document XHTML chargé en mémoire. Par exemple, imaginez un document tel que le premier fils de l'élément **body** soit une liste ordonnée, on souhaite accéder au troisième élément de cette liste afin de le rendre invisible. Illustrer chaque mécanisme, à l'aide d'énoncés JavaScript, d'éléments XHTML, etc.

Question 3 : Programmation côté serveur

Le modèle de conception MVC (Model View Controller) est utilisé fréquemment pour la conception des interfaces usager. Selon ce modèle, le contrôleur reçoit et traite les requêtes HTTP, s'occupe par exemple des initialisations, traces et contrôle les accès. La partie modèle représente les données de l'application, possiblement des données persistantes, et s'occupe des calculs. Finalement, la troisième partie de MVC, la vue, s'occupe de la présentation des données.

1. Pour chaque élément du modèle MVC, quelle technologie Web vue en classe est la plus appropriée ? Pourquoi ?
2. Décrivez, en vos propres mots, les interactions entre les parties de l'application et les mécanismes mis en place par les technologies vues en classe pour les échanges de données et le contrôle de flot.

Question 4 : Servlet

Expliquez brièvement en quoi consiste le mécanisme de réécriture d'adresses URL des technologies Java Servlet (méthode `encodeURL` d'un objet `response` de type `HttpServletResponse`). Spécifiquement,

1. Suite à l'exécution du code Java suivant, quel sera le format de la chaîne `url` ?
2. À quoi sert ce mécanisme ?
3. Quand devrait-on l'utiliser ?
4. Quelles sont les difficultés associées à son usage ?

```
String url = "GetCount";  
url = response.encodeURL( url );
```

Question 5 : Processeurs XML

1. Pour chaque application ci-dessous, choisissez le processeur XML le mieux adapté, DOM ou SAX.
2. Donnez une brève justification.
3. Décrivez, en vos propres mots, les principales étapes de l'analyse et les principaux éléments à programmer.

Applications nécessitant l'utilisation d'un processeur XML :

- A. Un compteur de balises, qui prends en entrée un document XML bien formé et compte le nombre d'occurrences d'une balise donnée.
- B. Une application qui rajoute automatiquement en haut d'une page XHTML bien formée un lien vers toutes les sections et sous sections de cette page.

Question 6 : Transformation XSL

1. Donnez le résultat de l'application de cette transformation XSL au document XML qui suit.
sequence.xsl :

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
3
4     <xsl:template match="/sequence">
5         <list>
6             <xsl:apply-templates select="list/element"/>
7         </list>
8     </xsl:template>
9
10    <xsl:template match="element">
11        <tuple>
12            <first>
13                <xsl:value-of select="."/>
14            </first>
15            <second>
16                <xsl:value-of select="../name"/>
17            </second>
18        </tuple>
19    </xsl:template>
20
21 </xsl:stylesheet>
```

sequence.xml :

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <sequence>
3     <list>
4         <name>alpha</name>
5         <element>1.0</element>
6         <element>2.0</element>
7     </list>
8     <list>
9         <name>beta</name>
10        <element>3.0</element>
11        <element>4.0</element>
12        <element>5.0</element>
13    </list>
14 </sequence>
```

2. Donnez le résultat de l'application de cette transformation XSL au document XML qui suit.

sequence.xsl :

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
3
4     <xsl:template match="/sequence">
5         <list>
6             <xsl:apply-templates select="list/element"/>
7         </list>
8     </xsl:template>
9
10    <xsl:template match="element">
11        <tuple>
12            <first><xsl:value-of select="."/></first>
13            <xsl:apply-templates select="../name"/>
14        </tuple>
15    </xsl:template>
16
17    <xsl:template match="name">
18        <second><xsl:value-of select="."/></second>
19    </xsl:template>
20
21 </xsl:stylesheet>
```

sequence.xml :

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <sequence>
3     <list>
4         <name>alpha</name>
5         <element>1.0</element>
6         <element>2.0</element>
7     </list>
8     <list>
9         <name>beta</name>
10        <element>3.0</element>
11        <element>4.0</element>
12        <element>5.0</element>
13    </list>
14 </sequence>
```

Question 7 : Ajax

On dit que les applications Web utilisant la technologie Ajax sont plus réactives que celles qui ne l'utilisent pas.

1. Justifiez cette affirmation, soyez précis, quelle est la technologie clé qui permet cette agilité?
2. Décrivez brièvement point pas point les étapes et les techniques mises en oeuvre lors d'un échange de messages entre un client et le serveur avec Ajax.

Question 8 : Service Web

Depuis le milieu des années 1970s, il existe des protocoles permettant de faire des appels de procédures sur un ordinateur distant. Contrairement aux autres technologies, les services Web reposent sur un ensemble de protocoles XML ouverts. Pour chacune des technologies XML suivantes, XML Schema, SOAP et WSDL, vous devez :

1. Décrire cette technologie ;
2. Expliquer son rôle par rapport aux services Web ;
3. Préciser ses relations avec les deux autres technologies.

(page blanche)

(page blanche)