請勿攜去
Not to be taken away

香 港 中 文 大 學
The Chinese University of Hong Kong

二0一八至一九年度上學期科目考試
Course Examination 1st Term, 2018-19

科目編號及名稱
Course Code & Title : CSCI2720 – Building Web Applications

時間 小時 分鐘
Time allowed : 2 hours 0 minutes

學號 座號
Student I.D. No. : Seat No. :

Instructions

- This is an open-book, open-notes examination
- No electronic device is allowed
- Write your answers clearly in the space provided in this exam script
- You need to return this exam script with all pages for grading
- Assume that the web browser in used is the almost-latest version of Google Chrome, and all libraries and frameworks are of the version used in the course
- Cross out all draft or rough work if they are not to be graded
- Please keep your answers legible, precise and concise

Markers' use only

Q1	Q2	Q3	Q4	Q5	Q6	Q7	**Total**
/10	/15	/20	/20	/20	/10	/5	/100

Question 1: HTML, CSS and JavaScript (10%)

Look at this HTML file, which is shown in a web browser.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Exam</title>
  <script>
    function b1() {
      document.getElementsByTagName("p")[0].innerText = 'Not a question.';
      document.getElementsByClassName("question")[0].style.color = 'red';
    }
    function b2() {
      document.querySelector("#question").style.display = 'none';
    }
    function b3() {
      b4();
      b5();
    }
    function b4() {
     alert("A");
      return new Promise((resolve, reject) => {
        setTimeout( () => {
          alert("B");
          resolve();
        }, 1000);
      });
    }
    async function b5() {
      alert("C");
      await b4();
      alert("D");
    }
  </script>
  </head>

  <body>
    <h1 id="question">The first question</h1>
    <p>To be or not to be? </p>

    <button onclick="javascript:b1()">Button 1</button><br>
    <button onclick="javascript:b2()">Button 2</button><br>
    <button onclick="javascript:b3()">Button 3</button><br>
  </body>
</html>
```

Briefly describe one by one what would happen when the 3 buttons are clicked on. Ignore anything shown in the JavaScript console.

Question 2: DOM Manipulation (15%)

You are given an HTML file with the following table as the only content.

Feature detection
DOM wrapped
XMLHttpRequest data retrieval
WebSocket
Server push data retrieval
Simple visual effects
Animation / advanced visual effects
Autocompletion tools
HTML generation tools
Widgets themeable / skinnable
Accessibility / graceful degradation
ARIA compliant
Developer tools, Visual design

Please write **JavaScript** code to automatically create an unordered list in the same page after the table, where each list item corresponds to the content of a table cell in this table. This should be done when the document is ready. Suppose that libraries like jQuery are already included.

Question 3: Security (20%)

a) In `http://www.foo.com/p1.html`, you (the server admin) realize that the Ajax code in the page is not able retrieve `http://www2.foo.com/data.txt` due to some error. What could be the reason? Please briefly explain, and suggest two possible solutions with some elaboration.
Suppose that both addresses are publicly accessible when typing directly into the URL bar of a web browser.

b) You are now asked to implement a login feature for this website. Describe one possible implementation so that after logging in, a user can stay logged in for 7 days or until the user logs out. Discuss how your implementation can avoid users from editing local files to pretend to be another user.

Question 4: MongoDB and Mongoose (20%)

For a web site serving data of movies, you need to store the following data:

- Information of ***movies***, which should include the **movie name**, the **year**, the **origin country**, the **director** (one person), a group of **actors** (many persons), and a **ranking** (e.g. 1.6)
- Information of ***persons***, which should include the **name** (**first name** and **last name**), and the **age**

In this database, all the **names** are very important and cannot be missing.

a) Please design efficient schemas for these data to be used in Mongoose. Put your assumptions in code comments if necessary.

```
let mongoose = require('mongoose');
let Schema = mongoose.Schema;









let Movie = mongoose.model('Movie', movieSchema);
let Person = mongoose.model('Person', personSchema);
```

b) Assume you are using the same JS file from part a). Simply using the function `CB()` as the callback, without error handling, please write down the queries for the following tasks.

i. List the movies of top 5 ranking in the year 2018, showing only the movie name.

ii. List the details for persons who has a last name "Chan" and an age of less than 50.

iii. List the movies of origin "USA" showing the movie name, and the director's age.

Question 5: RESTful API and Express (20%)

a) Suppose you are building a web service to allow users to manage books in a library. The books are uniquely identified by their ISBN (International Standard Book Number). You need to provide a URL scheme for creating, retrieving, updating, and deleting the book records. Please explain briefly why a RESTful API is helpful, and list out the basic RESTful URLs with the appropriate HTTP method you would provide.

b) According to your URL scheme, implement the corresponding route paths using Express with Node.js. You can simply call these given functions in the route paths for the relevant actions:

- `addBook(isbn, bookObj)`
- `updateBook(isbn, bookObj)`
- `getBook(isbn)`
- `delBook(isbn)`

where `isbn` should be the book identifier, and `bookObj` should contain all property/value pairs for details of a book. You are *NOT* required to prepare the server response. You may write down assumptions as code comments.

```
let express = require('express');
let app = express();
let bodyParser = require('body-parser');
app.use(bodyParser.urlencoded({ extended: true }));









app.listen(3000);
```

Question 6: Angular (10%)

In an *Angular 7* web app, this is the complete content for the file `src/app/app.component.html`:

```
<ol>
  <li *ngFor="let i of ['Mon', 'Tue', 'Wed', 'Thu']">Meal for {{i}}</li>
</ol>
```

a) Please write down the rendered text/symbols that would be shown on the browser screen using the `<app-root></app-root>` directive.

b) By changing ONLY **ONE** of the following four lines, detect the mouse click anywhere on the list to change between *showing* and *not showing* "Friday". No change in any other file is allowed.

```
<ol>

  <li *ngFor="let i of ['Mon', 'Tue', 'Wed', 'Thu']">Meal for {{i}}</li>

  <li *ngIf="friday">Friday</li>

</ol>
```

Question 7: Response Web Design (5%)

What is the importance of responsive web design? Please briefly discuss how responsive web design can be implemented with 3 examples, and how they are useful.

~ END OF THE EXAM ~