University of Toronto Scarborough

CSCC09

Spring 2018

Final Exam

First Name: _____

Last Name: _____

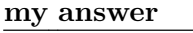Student Number: _____

_____ @mail.utoronto.ca

Do not turn this page until you have received the signal to start.

In the meantime, fill out the identification section above and read the instructions below carefully.

This 3 hours exam contains 22 pages (including this cover page) and 6 parts. Check to see if any pages are missing. Enter all requested information on the top of this page, and put your initials on the top of every page, in case the pages become separated.

You may **not** use notes, or any calculator on this exam.

There are several types of questions identified throughout the exam by their logos:

✓₁	select **the best answer** only among multiple choices.
✓★	select **all correct answers** (and those ones only) among multiple choices. At least one applies.
<u>my answer</u>	fill the blanks with a **short answer** (less than 5 words).
●—●	connect **each item from the left side** to one (and one only) item from the right side. A right side item might be connected to several left side items. The right side and the left side might not necessarily have the same number of items.

For each question, the following rules apply:

- **There is no partial credit**. This means that, for a given question, you either receive the full mark allocated if all answers are correct or 0 if there is any mistake.
- **Your answer should be clear**. Your answer to a question might be considered as incorrect if there is:
  - any part of the answer that cannot be read
  - any part of the answer that cannot be associated with its corresponding question part

Do not write anything in the table below.

Part:	1	2	3	4	5	6	Total
Points:	15	20	20	20	25	5	105
Score:							

1. **Part 1**

   (1.1) 2 points - ●—●

   Match each technology with its concept:

HTML	●	●	Content
CSS	●	●	Processing
Javascript	●	●	Presentation

   (1.2) 2 points - ✓★

   CSS can be used to define how the page should be rendered:

   ☐ on a screen

   ☐ on a printed page

   ☐ on a screenreader (that reads the page out loud)

   (1.3) 1 point - ✓$_1$

   The Document Object Model is

   ◯ an API for storing structured data locally (in the browser)

   ◯ a way to structure NoSQL databases

   ◯ the tree structure of a webpage

   ◯ an API to access the browser configuration

   (1.4) 2 points - **my answer**

   - The attribute _____ identifies a unique element in the DOM
   - The attribute _____ identifies several elements in the DOM
   that share the same characteristics

(1.5) 1 point - $\checkmark_1$

HTTP is

○ the tree structure of a webpage

○ a network protocol

○ a server configuration

○ a javascript method to fetch remote content

(1.6) 1 point - $\checkmark_1$

The default port for HTTP is

○ 22

○ 80

○ 3000

○ 8080

(1.7) 3 points - __my answer__

An HTTP request contains a _____, a query string, a set of _____ and an optional _____

(1.8) 3 points - __my answer__

An HTTP response contains a _____, a set of _____ and an optional _____

2. **Part 2**

(2.1) 2 points - ✓★

Which HTTP requests do not have a body:

       ☐ POST

       ☐ PUT

       ☐ GET

       ☐ PATCH

       ☐ DELETE

(2.2) 2 points - ✓★

Which HTTP methods are safe:

       ☐ POST

       ☐ PUT

       ☐ GET

       ☐ PATCH

       ☐ DELETE

(2.3) 2 points - ✓★

Which HTTP methods are idempotent:

       ☐ POST

       ☐ PUT

       ☐ GET

       ☐ PATCH

       ☐ DELETE

(2.4) 2 points - ✓★

Which methods are used when retrieving images, scripts and stylesheets:

☐ POST

☐ PUT

☐ GET

☐ PATCH

☐ DELETE

(2.5) 3 points - ●—●

Match each status code with its family:

1xx	●	●	Server error
2xx	●	●	Success
3xx	●	●	Client error
4xx	●	●	Information
5xx	●	●	Redirect

(2.6) 2 points - ✓★

Which methods can be used with Ajax:

☐ POST

☐ PUT

☐ GET

☐ PATCH

☐ DELETE

(2.7) 1 point - $\checkmark_1$

JSON is

     ⃝ a way to structure a NoSQL database

     ⃝ a way to represent structured data as strings

     ⃝ a special HTTP request to send structured data to the server

     ⃝ a specific javascript data structure

(2.8) 1 point - $\checkmark_1$

JSON data can be manipulated with Javascript only:

     ⃝ true

     ⃝ false

(2.9) 1 point - $\checkmark_1$

JSON is the only way to send structured data to the server:

     ⃝ true

     ⃝ false

(2.10) 1 point - $\checkmark_1$

The client notifies the server that the HTTP request contains JSON data:

     ⃝ by using a POST request

     ⃝ by setting a JSON parameter in the query string

     ⃝ by setting the `content-type` header

     ⃝ by setting the body mimetype

     ⃝ by setting a special cookie

(2.11) 1 point - $\checkmark_1$

The server notifies the client that the HTTP response contains JSON data:

     ⃝ by using a POST request

     ⃝ by setting a JSON parameter in the query string

     ⃝ by setting the `content-type` header

     ⃝ by setting the request mimetype

     ⃝ by setting a special cookie

(2.12) 1 point - $\checkmark_1$

When uploading a file, the content-type of the request should be set to:

      ○ application/json

      ○ application/x-www-form-urlencoded

      ○ multipart/form-data

      ○ the file mimetype

(2.13) 1 point - $\checkmark_1$

When downloading a file, the content-type of the response should be set to:

      ○ application/json

      ○ application/x-www-form-urlencoded

      ○ multipart/form-data

      ○ the file mimetype

3. **Part 3**

(3.1) 2 points - ✓★

In a typical application like the *Microblog* that manages messages and votes, what are the HTTP requests that follow the REST design principles (including the ones that we might not have implemented):

☐ GET /messages/

☐ GET /messages/89/

☐ POST /messages/

☐ POST /messages/89/

☐ PUT /messages/89/

☐ PATCH /messages/89/upvote/

(3.2) 1 point - ✓₁

What is pagination useful:

◯ to avoid retrieving entire collections

◯ to avoid overwriting existing data

◯ to split a big HTTP response into multiple ones

◯ to optimize database requests by caching data

(3.3) 2 points - ✓★

A piece of javascript code executed on the frontend (browser) can

☐ modify a local storage key/value pair

☐ modify an unprotected cookie key/value pair

☐ modify a session key/value pair

(3.4) 2 points - ✓★

A piece of javascript code executed on the backend (node.js) can

☐ modify a local storage key/value pair

☐ modify a cookie key/value pair

☐ modify a session key/value pair

(3.5) 3 points - ✓★

Cookies are exchanges between the client and the server:

☐ in the HTTP request query string

☐ in the HTTP request headers

☐ in the HTTP request body

☐ in the HTTP response headers

☐ in the HTTP response body

(3.6) 1 point - ✓₁

The session ID is

○ the username

○ the password salted hash

○ the primary key of the user's profile stored in the database

○ a random string

(3.7) 1 point - ✓₁

Stateful authentication is typically done

○ using the user's IP address

○ using the user's unique browser ID

○ using a database

○ using sessions

(3.8) 1 point - ✓₁

Assuming that the client has been authenticated (stateful authentication), what happens if the client tries to access to a protected resource without sending back the session ID cookie to the server:

○ the server recovers the session from the database, recreates the session ID cookie and grants access to the ressource

○ the server generates a new session, recreates the session ID cookie and grants access to the ressource

○ the server denies access to the ressource

(3.9) 2 points - ✓★

Using stateful authentication

☐ the same user can be authenticated on two different computers at the same time

☐ the same user can be authenticated on the same computer using two different browsers at the same time

☐ two users can be authenticated on the same browser at the same time

☐ two users can be authenticated on the same computer using two different browsers at the same time

(3.10) 4 points - ✓★ and **my answer**

A web application **A** uses a third-party authentication (OAuth) service from **B** to authenticate a client **C**. Considering the options below, check the ones that are part the OAuth scheme and leave the others unchecked. For all options that you checked (and only those ones), fill the blanks with either: **A**, **B** or **C**.

☐ _____ sends the login and password to _____

☐ the latter forwards the login/password to _____

☐ the latter verifies the login/password and returns a token to _____

☐ the latter forwards the token to _____

☐ the latter ask _____ to verifies the token and return the user's profile

(3.11) 1 point - ✓₁

HTTP/2 allows

○ the backend to bundle javascript and CSS files into one file sent back as an HTTP response

○ the frontend to request multiple resources with a single HTTP request

○ the frontend and the backend to compress HTTP requests and responses

○ the backend to reply to a single HTTP request with multiple HTTP responses

4. **Part 4**

(4.1) 1 point - ✓₁

When using basic authentication, login and passwords are sent:

○ in the query string

○ in the headers

○ in the body

○ in the cookie

(4.2) 1 point - ✓₁

When **not** using basic authentication, login and passwords are sent:

○ in the query string

○ in the headers

○ in the body

○ in the cookie

(4.3) 1 point - ✓₁

Over HTTP (without considering HTTPS), passwords are by default sent from the client to the server as:

○ clear text

○ encrypted

○ hashed

○ salted hashed

(4.4) 1 point - ✓₁

On the server, passwords are best stored as:

○ clear text

○ hashed

○ salted hashed

(4.5) 1 point - ✓₁

Insufficient Transport Layer Protection can be mitigated

  ◯ by using TLS (a.k.a SSL)

  ◯ by sanitizing user's inputs

  ◯ by protecting cookies

  ◯ by hashing passwords with a salt

(4.6) 2 points - ✓★

SQL injection attacks can be mitigated by:

  ☐ using TLS (a.k.a SSL)

  ☐ sanitizing user's inputs

  ☐ using NoSQL database

  ☐ using Memcached combined with an SQL database

(4.7) 1 point - ✓₁

When visiting a specific webpage, the browser says that "the website has a non trusted certificate", it means that

  ◯ the browser does not recognized the certificate because it is the first time I am visiting this website

  ◯ the browser does not know the private key associated with the certificate

  ◯ the certificate is not signed by a certificate authority known by my browser

  ◯ the certificate is not signed by the server hosting the website

(4.8) 1 point - ✓₁

Mixed-content occurs when

  ◯ the server sets the wrong content-type header

  ◯ the client retrieves HTTP and HTTPS content from the same domain

  ◯ the server does no correctly sanitize users inputs

  ◯ the client retrieves a malicious script

(4.9) 3 points - ●—●

Match each attack with the type of content it injects into the web application:

SQLi (SQL injection) ● ● HTML content

Content Spoofing ● ● URLs

XSS (Cross-Site Scripting) ● ● Javascript

CSRF (Cross-Site Request Forgery) ● ● Database queries

(4.10) 3 points - ●—●

Match each cookie flag with the type of attack it tries to mitigate (but not necessarily prevent)

HttpOnly ● ● Mixed-content

SecureFlag ● ● Content Spoofing

SameOrigin ● ● Cross-Site Scripting

● Cross-Site Request Forgery

● SQL Injection

(4.11) 2 points - ✓★

Assuming, browser loads a webpage from **A** that contains a piece of javascript code. Without considering any cross-sharing HTTP request, what resources from another domain **B** can be loaded by this piece of code without violating the same origin-policy:

☐ a script using the `SCRIPT tag`

☐ a stylesheet using the `LINK tag`

☐ an image using the `IMG tag`

☐ a piece of HTML using the `IFRAME tag`

☐ a script using an Ajax request

☐ some JSON data using an Ajax request

(4.12) 1 point - ✓₁

Considering the scenario given above, what happens when the same-origin policy is violated:

◯ the client's browser blocks the HTTP request going from the client to **B**

◯ the client's browser blocks the HTTP response coming from **B** back to the client

◯ **A** blocks the HTTP request going from the client to **B**

◯ **A** blocks the HTTP response coming from **B** back to the client

◯ **B** blocks the HTTP request coming from the client

(4.13) 2 points - ✓★

To enable the Cross-Origin Ressource Sharing (CORS)

☐ the client's browser sets the `Access-Control-Allow-Origin` header in the HTTP request going to **A**

☐ **A** sets the `Access-Control-Allow-Origin` header in the HTTP response going back to the client

☐ the client's browser sets the `Access-Control-Allow-Origin` header in the HTTP request going to **B**

☐ **B** sets the `Access-Control-Allow-Origin` header in the HTTP response going back to the client

5. **Part 5**

(5.1) 2 points - ✓★

Assuming that the function `foo` is an asynchronous callback-based function that reads the content of a file given its filename and a callback. What are the correct ways to display the file content:

☐
```
var content = foo('/path/to/file');
console.log(content);
```

☐
```
var content = foo('/path/to/file', function(err, result){
    return result;
});
console.log(content);
```

☐
```
foo('/path/to/file', function(err, result){
    content = result;
});
console.log(content);
```

☐
```
foo('/path/to/file', function(err, result){
    console.log(result);
});
```

(5.2) 2 points - ✓★

Assuming that the function `foo` is an asynchronous promise-based function that reads the content of a file given its filename. What are the correct ways to display the file content:

☐
```
var content = foo('/path/to/file');
console.log(content);
```

☐
```
var content = foo('/path/to/file').then(function(result){
                return result;
            });
console.log(content);
```

☐
```
foo('/path/to/file').then(function(result){
    var content = result;
});
console.log(content);
```

☐
```
foo('/path/to/file').then(function(result){
    console.log(result);
});
```

☐
```
var content = await foo('/path/to/file')
async console.log(content);
```

☐
```
(async function(){
    var content = await foo('/path/to/file')
    console.log(content);
}());
```

☐
```
(async function(){
    return await foo('/path/to/file')
}()).then(function(content){
    console.log(content);
});
```

(5.3) 2 points - ✓★

Assuming that the function `foo` is a callback-based function that reads the content of a file given its filename and a callback. How to transform this function into a promise-based function that reads the content of a file given its filename;

☐
```
function newFoo (filename){
    return new Promise(function(resolve, reject){
        foo(filename, function(err, result){
            resolve(result);
        });
    });
};
```

☐
```
function newFoo (filename){
    return foo(filename, function(err, result){
        new Promise(resolve, reject){
            resolve(result);
        };
    });
};
```

☐
```
function newFoo (filename){
    return new Promise(foo(filename));
}
```

(5.4) 2 points - ✓★

Assuming that the function `foo` is a promise-based function that reads the content of a file given its filename. How to transform this function into a callback-based function:

☐
```
var newFoo = function(filename, callback){
    return callback(foo(filename));
}
```

☐
```
var newFoo = function(filename, callback){
    return foo(filename).then(function(result){
        callback(null, result);
    });
};
```

☐
```
var newFoo = function(filename, callback){
    return foo(filename).then(callback);
};
```

☐
```
var newFoo = function(filename, callback){
    var content = await foo(filename)
    callback(null, content);
};
```

(5.5) 1 point - ✓₁

I18N (Internationalization) is the process of developing a software that

○ is language agnostic

○ is automatically adapted for a specific language using a translation API

○ is specifically adapted for a specific language using a locale

(5.6) 1 point - ✓₁

L10N (Localization) is the process of developing a software that

○ is language agnostic

○ is automatically adapted for a specific language using a translation API

○ is specifically adapted for a specific language using a locale

(5.7) 1 point - ✓₁

By default, a website can detect automatically my language locale based on

    ⃝ the browser that sets an HTTP header with my preferred language based on my geo-location

    ⃝ the browser that sets an HTTP header with my preferred language stored in the browser settings

    ⃝ the server that sets my preferred language based on my profile stored in the database

    ⃝ the server that sets my preferred language based on my IP address

(5.8) 1 point - ✓₁

To own the domain c09rocks.com, one should ask

    ⃝ a domain name registrar

    ⃝ an internet service provider

    ⃝ a certificate authority

    ⃝ the server that hosts the website

(5.9) 1 point - ✓₁

Assuming I owe the domain c09rocks.com, who should generate a certificate for that domain:

    ⃝ a domain name registrar

    ⃝ the internet service provider

    ⃝ a certificate authority

    ⃝ myself

(5.10) 1 point - ✓₁

An HTTP proxy cache is useful for

    ⃝ caching database requests

    ⃝ optimizing static content delivery

    ⃝ optimizing dynamic content delivery

    ⃝ distributing the load across multiple web servers

(5.11) 1 point - ✓₁

Memcached is

- ◯ a technique to shadow the web server's memory in case of a crash
- ◯ a library for optimizing raw database requests
- ◯ an operation that pushes memory overflow to the database
- ◯ a distributed shared cache library for dynamic content

(5.12) 1 point - ✓₁

A load balancer

- ◯ restructures a database dynamically and improve its efficiency
- ◯ takes incoming HTTP requests and spread them across multiple web servers internally
- ◯ synchronizes the databases between different web servers
- ◯ routes the traffic to different web servers by changing the DNS dynamically

(5.13) 1 point - ✓₁

A CDN is

- ◯ restructures a database dynamically and improve its efficiency
- ◯ takes incoming HTTP requests and spread them across multiple web servers internally
- ◯ synchronizes the databases between different web servers
- ◯ routes the traffic to different web servers by changing the DNS dynamically

(5.14) 2 points - ✓★

Web analytics can be done:

- ☐ on the browser side
- ☐ on the server side

(5.15) 2 points - ✓★

The Do Not Track browser option

- □ sends a special HTTP request header that requests the server not to track me

- □ does not forward third-party cookies

- □ does not cache the webpage

- □ does not record the page in the history

(5.16) 2 points - ✓★

In private/incognito mode, the browser:

- □ does not forward third-party cookies

- □ does not cache the webpage

- □ does not record the page in the history

(5.17) 2 points - **my answer**

In Javascript, multi-threading can be achieve by using

- _____ in the frontend (Javascript running in the browser)

- _____ in the backend (Javascript running with Node.js)

6. **Bonus**

(6.1) 1 point - $\checkmark_1$

Would you recommend this course to other students

    ○ no, better learn web development on their own

    ○ no, the course is too demanding

    ○ no, the course is too boring

    ○ meh

    ○ yes, it is interesting

    ○ for sure, easy A

(6.2) 1 point - $\checkmark_1$

Have you filled the course evaluation for this course?

    ○ of course!

    ○ yeah whatever

    ○ who cares?

    ○ I have no clue what you are talking about

    ○ 42 is the correct answer

(6.3) 3 points - **my answer**

Share something with the course staff

Thank you for this great semester and have a good summer!