TECHNISCHE UNIVERSITEIT DELFT
Faculteit Elektrotechniek, Wiskunde en Informatica

Midterm

TI1506 Web and Database Technology

Tuesday, 13 December 2016
13.30-15.30

INSTRUCTIONS:

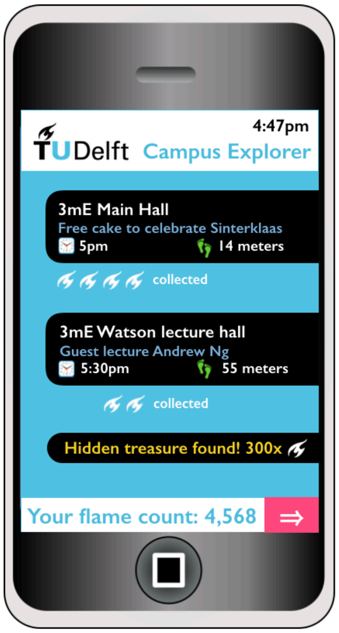
- This exam consists of 2 parts (DB and Web) and a total of 40 multiple-choice questions. All questions are worth an equal number of points.
- The usage of books, notes, old exams, and other written resources is explicitly **FORBIDDEN** during the exam. The use of electronic aids such as smart-phones, laptops, etcetera, is **ALSO NOT** allowed.
- There is only one right answer for each question. If you think there are more, pick the best one.
- You are not allowed to make corrections on the multiple-choice answer form (MAF). You are therefore advised to first mark the answers on this exam and later copy them to the MAF. If you need to make corrections anyway, ask for a new form and copy all your answers to it.
- You are not allowed to take the exam sheet with you after the exam. We will publish online the text of the exam together with its solutions.
- Note that the order of the answers on your MAF form is not always A-B-C-D.
- Be sure to fill in all header information on the MAF. Enter your *student number* on the form with digits as well as by filling the boxes.
- Sign the MAF. Without your signature, the form is not valid. Since you might forget this at the end, you are advised to do this at the start of the exam.

Part 1 – Web (25 questions)

The first eight questions relate to the **TU Delft Campus Explorer (TUDEX)** Web application, which consists of a server-side and a client-side component. TUDEX was designed to incentivise students to explore the TU Delft campus.

While students walk around campus, the app alerts them to events (guest lectures, free food, concerts, etc.) that are happening nearby in the near future - at most 100 meters from the current location and starting within the hour. These events are continuously updated by TU Delft staff members and stored in a database that the server-side component has access to. The app also contains a gamification mechanism: with each notification the user receives a number of points (or flames 🔥). Every day at several different locations on campus virtual "treasures" are hidden – if users pass in close vicinity to those locations they receive additional flames. Users can view the top-ranked campus explorer (i.e. those that have collected the most flames) across the entire university and for each faculty separately.

The client-side interface of TUDEX is shown below. Clients (i.e. Web browsers) poll the server every 90 seconds for new events and hidden treasures. Clients then filter the events to be shown to the user according to the user's current location and the current time.



4:47pm

TUDelft Campus Explorer

3mE Main Hall
Free cake to celebrate Sinterklaas
5pm 14 meters

🔥🔥🔥🔥 collected

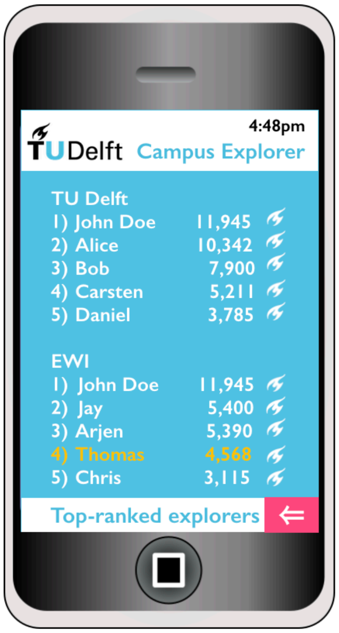
3mE Watson lecture hall
Guest lecture Andrew Ng
5:30pm 55 meters

🔥🔥 collected

Hidden treasure found! 300x 🔥

Your flame count: 4,568 ⇒

NOTIFICATIONS



4:48pm

TUDelft Campus Explorer

TU Delft	
1) John Doe	11,945 🔥
2) Alice	10,342 🔥
3) Bob	7,900 🔥
4) Carsten	5,211 🔥
5) Daniel	3,785 🔥
EWI	
1) John Doe	11,945 🔥
2) Jay	5,400 🔥
3) Arjen	5,390 🔥
4) Thomas	4,568 🔥
5) Chris	3,115 🔥

Top-ranked explorers ⇐

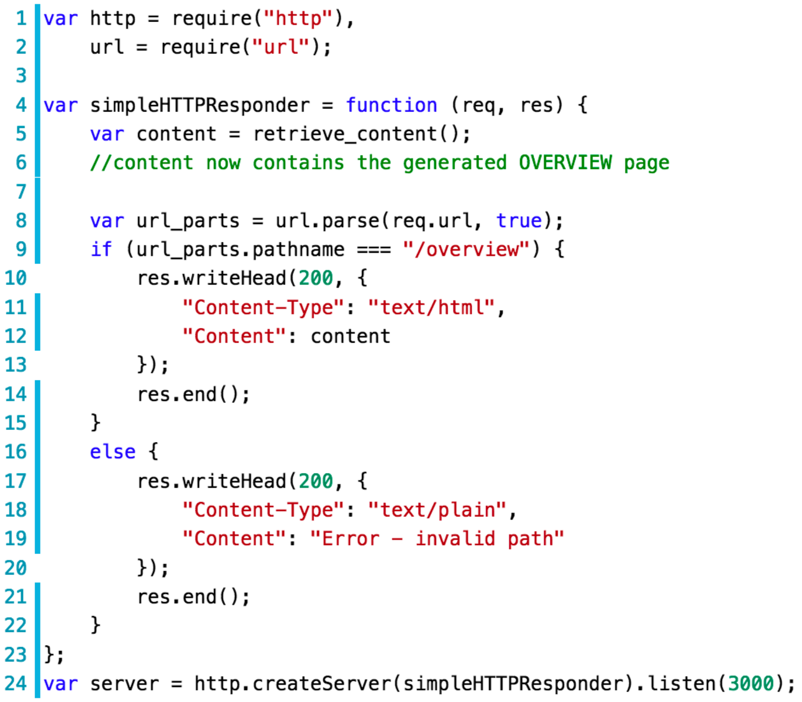
RANKINGS

The two Web pages of our Web application are:

- notifications: this page shows a list of all events in the user's vicinity that will start in the near future. The list is automatically updated after each server poll. On the client, filtering is performed to only list those events that are within a range of 100 meters from the user's current location and whose starting time is at most one hour into the future (and not in the past). A click on the pink arrow button at the bottom right brings the user to the rankings page.

- rankings: the rankings page contains the current ranking of top-performing users (i.e. those users that have collected the most flames) across the entire university and across the faculty the user is a student in. A click on the pink arrow button at the bottom right brinks the user back to the notifications page.

QUESTION 1. When inspecting the server-side source code of TUDEX, you discover the code snippet below. What is the **main issue** of this piece of code?



```
 1 var http = require("http"),
 2     url = require("url");
 3
 4 var simpleHTTPResponder = function (req, res) {
 5     var content = retrieve_content();
 6     //content now contains the generated OVERVIEW page
 7
 8     var url_parts = url.parse(req.url, true);
 9     if (url_parts.pathname === "/overview") {
10         res.writeHead(200, {
11             "Content-Type": "text/html",
12             "Content": content
13         });
14         res.end();
15     }
16     else {
17         res.writeHead(200, {
18             "Content-Type": "text/plain",
19             "Content": "Error - invalid path"
20         });
21         res.end();
22     }
23 };
24 var server = http.createServer(simpleHTTPResponder).listen(3000);
```

A) The way the `simpleHTTPResponder` function is passed as argument into `http.createServer` in line 24 is wrong and as a consequence, the server will not receive any HTTP requests on port 3000.

B) The HTTP response will be truncated to 200 bytes by the Web server, independent of the URL's path.

C) The Web browser will render a blank page, independent of the URL's path.

D) The HTTP response is never sent.

QUESTION 2. Having identified and fixed the issue in question 1, you continue to explore TUDEX which is accessible at http://tudex.nl:3000/. You access http://tudex.nl:3000/notifications with your browser and view the Web page's source code. You find the following lines:

```
<!doctype html>
<html manifest = "tudex.appcache">
```

You also take a look at `tudex.appcache` and find it to contain the following:

```
CACHE MANIFEST
FALLBACK:
/ /offline.html
```

What is the effect of this setup?

A) When the user first accesses http://tudex.nl:3000/notifications, the browser only stores http://tudex.nl:3000/offline.html in its appcache. In subsequent visits to http://tudex.nl:3000/notifications, the user is shown offline.html – independent of whether the user is offline or online at that moment.

B) When the user first accesses http://tudex.nl:3000/notifications, the browser only stores http://tudex.nl:3000/offline.html and http://tudex.nl:3000/notifications in its appcache. In subsequent visits to http://tudex.nl:3000/notifications, the user is only shown the cached version of notifications when the user is offline.

C) When the user first accesses http://tudex.nl:3000/notifications, the browser stores http://tudex.nl:3000/offline.html, http://tudex.nl:3000/notifications and http://tudex.nl:3000/rankings in its appcache. In subsequent visits to http://tudex.nl:3000/notifications, the user is only shown the cached version of notifications when the user is offline.

D) When the user first accesses http://tudex.nl:3000/notifications, the browser only stores http://tudex.nl:3000/offline.html and http://tudex.nl:3000/notifications in its appcache. In subsequent visits to http://tudex.nl:3000/notifications, the user is shown the cached version of notifications – independent of whether the user is offline or online at that moment.

QUESTION 3. The first version of TUDEX tracks users through fat URLs. The TUDEX developers wrote a whole blog post on their decision and listed the following four tracking abilities that fat URLs possess. Which of the listed abilities are **correct**?

1) Fat URLs allow TUDEX to track a user across different browsers and devices, the user can simply switch to a new browser/device and open http://tudex.nl:3000/ .
2) Fat URLs allow TUDEX to track users that access the Web through firewalls (thus obscuring the users' real IP address).
3) Fat URLs allow TUDEX to distinguish users on the same device that use different browsers.
4) Fat URLs reduce the load on the server compared to IP-address based user tracking.

A) Only 1, 3 and 4 are correct.

B) Only 2 and 3 are correct.

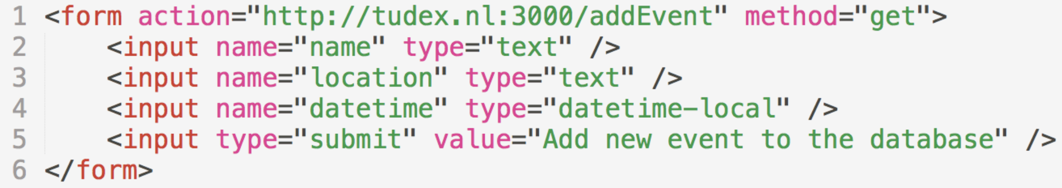
C) Only 1 and 2 are correct.

D) Fat URLs possess all of the listed tracking abilities.

QUESTION 4. The second version of TUDEX uses Secure HTTP. In Secure HTTP, the request and response data are encrypted before being sent across the network. Many different cryptographic protocols exist though. Who determines which protocol to use in a communication between client and server?

- **A)** Client and server negotiate the cryptographic protocol to use in their communication.
- **B)** The party initiating the communication determines the cryptographic protocol to use.
- **C)** The IETF regularly releases an overview of the latest cryptographic protocols. All clients and servers update their cryptographic implementations accordingly. This ensures that the most secure cryptographic protocols are always employed.
- **D)** The W3C regularly releases an overview of the latest cryptographic protocols. All clients and servers update their cryptographic implementations accordingly.

QUESTION 5. TUDEX also offers an administrative interface (accessible at http://tudex.nl:3000/addEvent) for TU Delft staff members in order to submit upcoming campus events to TUDEX's server-side component. Is this piece of code sufficient for this purpose?



```
1 <form action="http://tudex.nl:3000/addEvent" method="get">
2     <input name="name" type="text" />
3     <input name="location" type="text" />
4     <input name="datetime" type="datetime-local" />
5     <input type="submit" value="Add new event to the database" />
6 </form>
```

- **A)** No. The value of the `action` attribute has to be a JavaScript function that employs Ajax to establish a communication between the client and the TUDEX server.
- **B)** No. The value of the `method` attribute is incorrect; it has to be set to `put`, as data is sent from client to server.
- **C)** No. The `action` attribute has to be set in the `input` type `submit` (line 5) instead of the `form` element (line 1).
- **D)** Yes, the code fulfils its purpose.

QUESTION 6. After an initial trial period, the TUDEX developers decide to change the URL path of the notifications page from http://tudex.nl:3000/notifications to http://tudex.nl:3000/start. The developers want users that access TUDEX at http://tudex.nl:3000/notifications through their browsers to be automatically redirected to http://tudex.nl:3000/start. This can be achieved through an HTTP response with a particular status code. Which one?

- **A)** 201
- **B)** 301
- **C)** 401
- **D)** 501

QUESTION 7. In the third version of TUDEX, the developers have decided to track users by providing basic access authentication. What effects do Web caches have on the server hosting this version of TUDEX and on the clients using this version of the TUDEX application?

A) The server load will be reduced due to the number of Web caches maintaining copies of TUDEX. The client's demand on the origin server will decrease.
B) The demand on the server will not change significantly. The clients do not receive content any faster.
C) The server's processing power will increase as the number of Web caches maintaining copies of TUDEX increases. The clients will receive content faster.
D) The server's distance delay and redundant data transfer will be increased. The client's demand on the origin server will also be increased.

QUESTION 8. Which of the following HTTP request methods are minimally required to access the full functionality of TUDEX for both TU Delft staff members (that access the administrative interface from Question 5) and students?

A) GET
B) GET, HEAD
C) HEAD, POST
D) GET, HEAD, PUT

++ Note: The following questions do not refer to the TUDEX application ++

QUESTION 9. The `Content-MD5` header field in an HTTP response …

A) allows the Web cache to determine when the Web resource on the origin server was last modified.
B) provides a public key to decrypt the message body.
C) provides an integrity check for the message body.
D) allows the Web cache to employ MD5-based encoding on the origin server's HTTP response.

QUESTION 10. Which of the following statements best describes the HTML element `<input type="hidden">`?

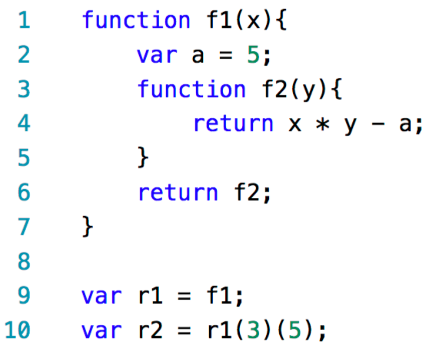
A) A form control that is displayed but whose value is not submitted to the server.
B) A form control that is displayed and whose value is submitted to the server.
C) A form control that is not displayed but whose value is submitted to the server.
D) A form control that is not displayed and whose value is not submitted to the server.

QUESTION 11. Which of the following statements about HTTP basic authentication (BA) are correct?

1. BA uses both HTTP header fields and the HTTP message body to transmit the credentials (username and password).
2. BA provides no confidentiality protection for the transmitted credentials.
3. The base-64 encoding ensures that only the correct username and password are transmitted.
4. The client's credentials need to be sent in every single HTTP request to a server requiring BA.
5. The client only sends the credentials in the first HTTP request to a server requiring BA (the server stores those credentials, thus removing the need for the client to resend the credentials).

A) Only 1, 2 and 4.
B) Only 2 and 4.
C) Only 1, 3 and 5.
D) Only 3 and 5.

QUESTION 12. After executing the JavaScript code snippet below in the browser, what is the type of `r2`?



```
function f1(x){
    var a = 5;
    function f2(y){
        return x * y - a;
    }
    return f2;
}

var r1 = f1;
var r2 = r1(3)(5);
```

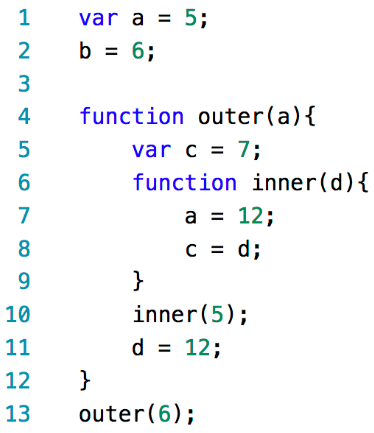
A) `Function`

B) `Number`

C) `null`

D) `undefined`

QUESTION 13. After executing the JavaScript code snippet below in the browser, how many variables have a global scope?



```
var a = 5;
b = 6;

function outer(a){
    var c = 7;
    function inner(d){
        a = 12;
        c = d;
    }
    inner(5);
    d = 12;
}
outer(6);
```

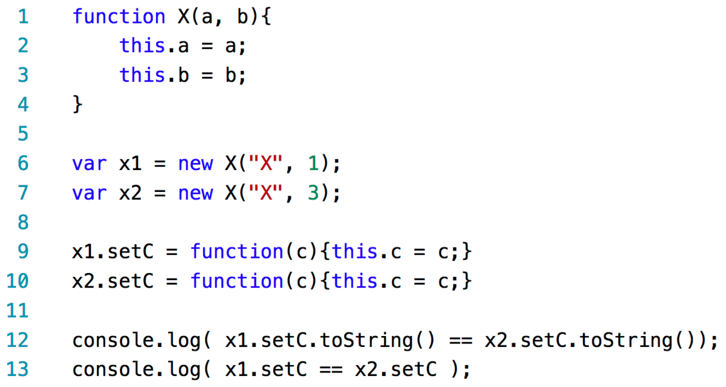
A) 1

B) 2

C) 3

D) 4

QUESTION 14. After executing the JavaScript code snippet below in the browser, what is the output on the console?



```
function X(a, b){
    this.a = a;
    this.b = b;
}

var x1 = new X("X", 1);
var x2 = new X("X", 3);

x1.setC = function(c){this.c = c;}
x2.setC = function(c){this.c = c;}

console.log( x1.setC.toString() == x2.setC.toString());
console.log( x1.setC == x2.setC );
```

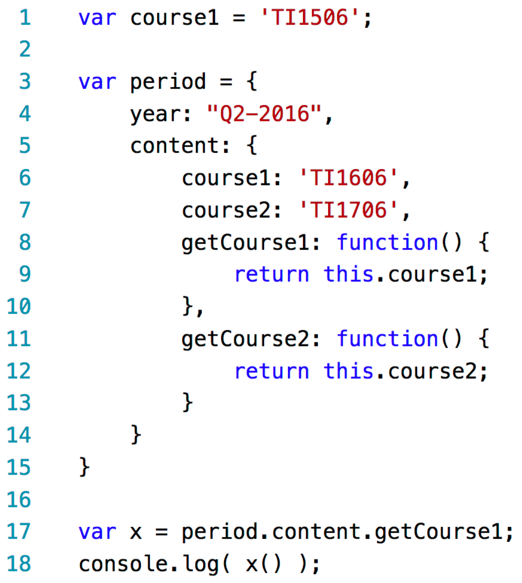
```
A) true
   false

B) false
   true

C) true
   true

D) false
   false
```

QUESTION 15. After executing the JavaScript code snippet below in the browser, what is the output on the console?



```
var course1 = 'TI1506';

var period = {
    year: "Q2-2016",
    content: {
        course1: 'TI1606',
        course2: 'TI1706',
        getCourse1: function() {
            return this.course1;
        },
        getCourse2: function() {
            return this.course2;
        }
    }
}

var x = period.content.getCourse1;
console.log( x() );
```

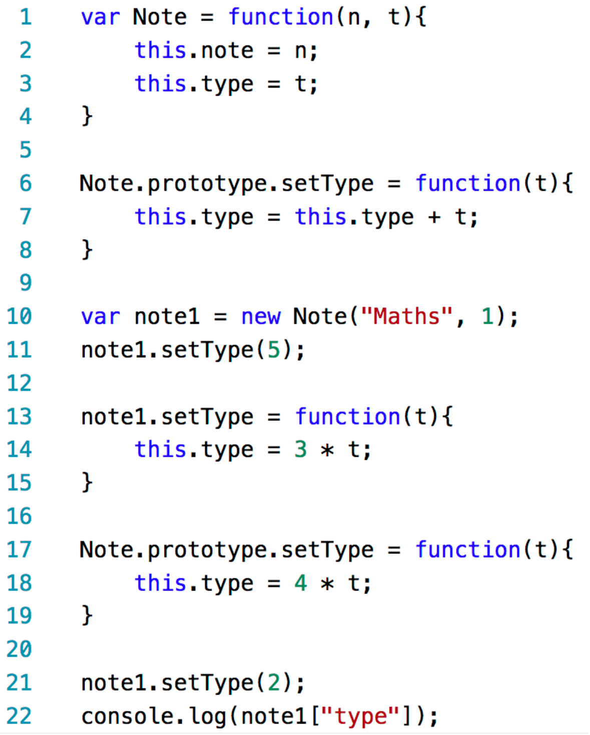
```
A) TI1506

B) TI1606

C) TypeError: x is
  not a function

D) undefined
```

<u>QUESTION 16.</u> After executing the JavaScript code snippet below in the browser, what is the output on the console?



```
var Note = function(n, t){
    this.note = n;
    this.type = t;
}

Note.prototype.setType = function(t){
    this.type = this.type + t;
}

var note1 = new Note("Maths", 1);
note1.setType(5);

note1.setType = function(t){
    this.type = 3 * t;
}

Note.prototype.setType = function(t){
    this.type = 4 * t;
}

note1.setType(2);
console.log(note1["type"]);
```

A) 6

B) 8

C) 18

D) 24

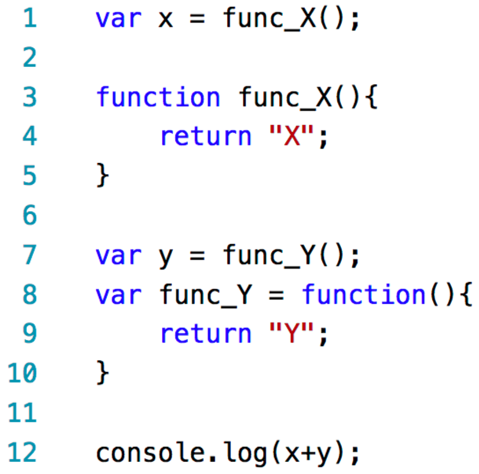
<u>QUESTION 17</u>. There are three events in the event queue of a node.js instance (in this order): (1) a request to update the in-memory array `myTodos`, (2) a request to send a file (size of 200 KB) from a client to the server, and, (3) a request to send the array `myTodos` in JSON format to a client. Other events do not enter the event queue. Which of the following statements is **<u>true</u>**?

A) It does not matter if any of these events are executed in a blocking or non-blocking fashion, the overall time it takes to process all three events is the same in both (the blocking and the non-blocking) settings.
B) It does matter if (3) is executed in a blocking or non-blocking fashion, the overall time required to process all three events is less when (3) is executed in a non-blocking fashion.
C) It does matter if (2) is executed in a blocking or a non-blocking fashion; the overall time to process all three events is less when (2) is executed in a non-blocking fashion.
D) None of the above statements are correct.

<u>QUESTION 18</u>. How many unique IP addresses exist in the IPv4 address space?

A) ~433.3 billion
B) ~4.3 billion
C) ~433 million
D) ~43 million

QUESTION 19. Executing the JavaScript code below in the browser leads to which error?



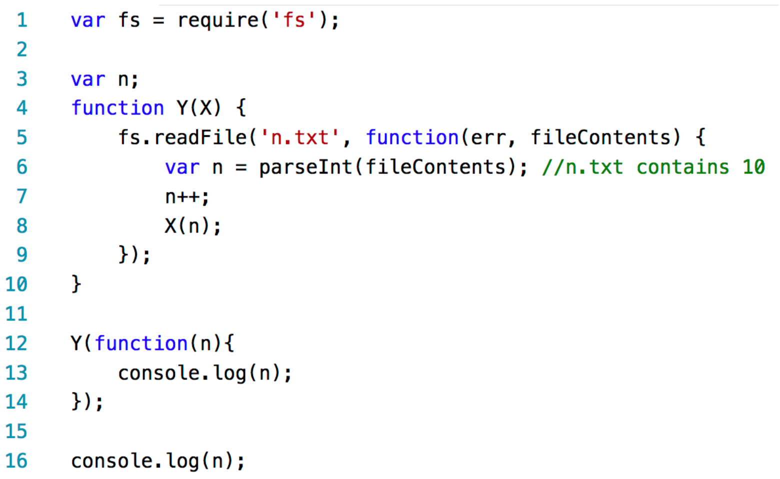
```
1     var x = func_X();
2
3     function func_X(){
4         return "X";
5     }
6
7     var y = func_Y();
8     var func_Y = function(){
9         return "Y";
10    }
11
12    console.log(x+y);
```

A) Line 1: `TypeError: func_X is not a function.`
B) Line 7: `TypeError: func_Y is not a function.`
C) Line 8: `FunctionError: function name not defined.`
D) Line 12: `TypeError: '+' only defined for type Number.`

QUESTION 20. What is the main purpose of a "CSS reset"? To reset …

A) all DOM elements to their default state as defined in the Web application's stylesheet, (i.e. to a state before user interactions may have changed their appearance).
B) all dynamic attribute values of the Web application stylesheet to their default value.
C) the browser-specific default stylesheet to a consistent baseline.
D) the user-specific stylesheet and replace it by the Web application's stylesheet.

QUESTION 21. Executing this node.js code yields which console output?



```
var fs = require('fs');

var n;
function Y(X) {
    fs.readFile('n.txt', function(err, fileContents) {
        var n = parseInt(fileContents); //n.txt contains 10
        n++;
        X(n);
    });
}

Y(function(n){
    console.log(n);
});

console.log(n);
```

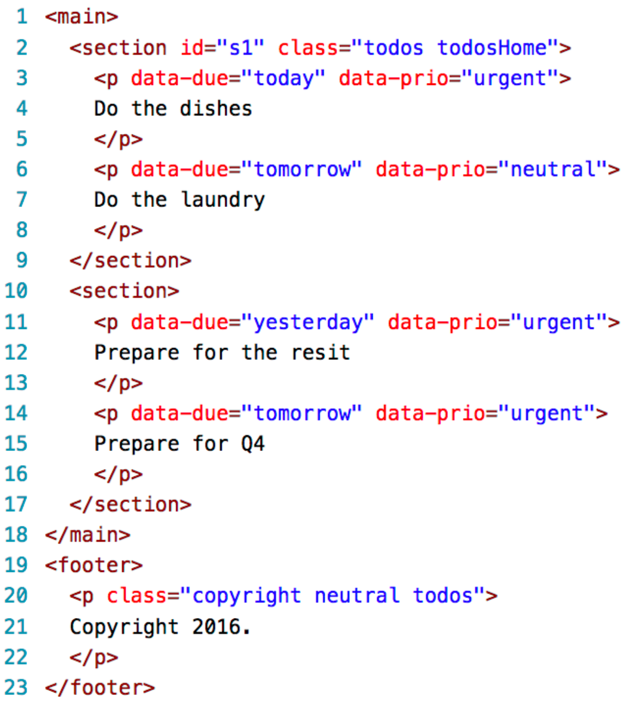
A) `undefined`
`undefined`

B) `11`
`undefined`

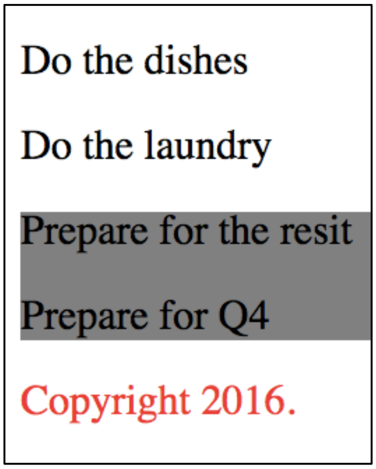
C) `undefined`
`11`

D) `11`
`11`

QUESTION 22. Consider the HTML page snippet below (left). For brevity, only the content inside `<body>` is shown; you can assume code in the header that loads the corresponding CSS. Which CSS snippet will generate the presentation of the content as shown below (right)?



```
<main>
  <section id="s1" class="todos todosHome">
    <p data-due="today" data-prio="urgent">
    Do the dishes
    </p>
    <p data-due="tomorrow" data-prio="neutral">
    Do the laundry
    </p>
  </section>
  <section>
    <p data-due="yesterday" data-prio="urgent">
    Prepare for the resit
    </p>
    <p data-due="tomorrow" data-prio="urgent">
    Prepare for Q4
    </p>
  </section>
</main>
<footer>
  <p class="copyright neutral todos">
  Copyright 2016.
  </p>
</footer>
```



Do the dishes

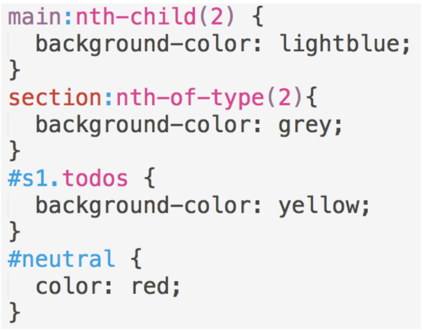
Do the laundry

Prepare for the resit

Prepare for Q4

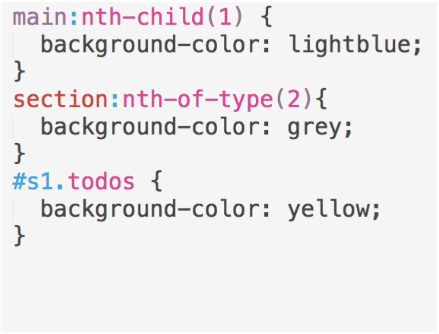
Copyright 2016.

A)



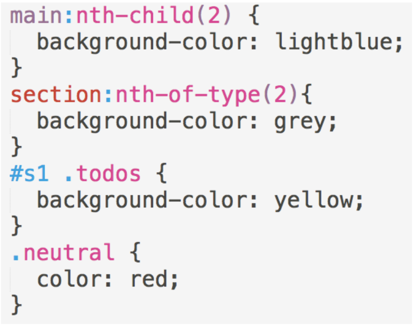
```
main:nth-child(2) {
  background-color: lightblue;
}
section:nth-of-type(2){
  background-color: grey;
}
#s1.todos {
  background-color: yellow;
}
#neutral {
  color: red;
}
```

B)



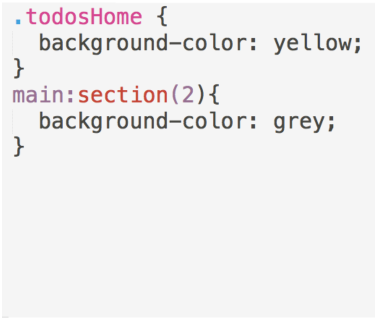
```
main:nth-child(1) {
  background-color: lightblue;
}
section:nth-of-type(2){
  background-color: grey;
}
#s1.todos {
  background-color: yellow;
}
```

C)



```
main:nth-child(2) {
  background-color: lightblue;
}
section:nth-of-type(2){
  background-color: grey;
}
#s1 .todos {
  background-color: yellow;
}
.neutral {
  color: red;
}
```

D)



```
.todosHome {
  background-color: yellow;
}
main:section(2){
  background-color: grey;
}
```

QUESTION 23. Consider the HTML page snippet from **Question 22** again. For all todo items, whose `data-prio` attribute is set to urgent, the string "Urgent!" should appear next to the item. How can this effect be achieved through CSS?

A)

```
p["urgent"]:after {
   content: "Urgent!";
}
```

B)

```
p[data-prio="urgent"]::after {
   content: "Urgent!";
}
```

C)

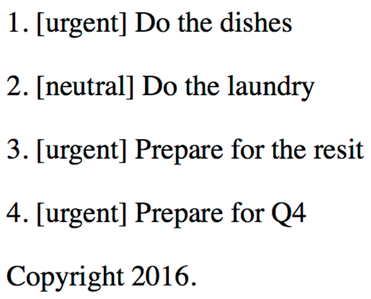
```
data-prio["urgent"]::after {
   attribute: "Urgent!";
}
```

D) Conditional value selection is not possible in CSS (i.e. none of the above CSS snippets are correct), additional JavaScript code is required to achieve this effect.

QUESTION 24. What is the relationship between JavaScript and ECMAScript?

A) ECMAScript is the subset of JavaScript that is specific to client-side JavaScript (i.e. all JavaScript components related to the Web browser).
B) ECMAScript is node.js' implementation of the JavaScript language.
C) ECMAScript is a scripting-language standard and JavaScript is one of its implementations.
D) JavaScript is an extension of ECMAScript, allowing not only client-side but also server-side applications to be implemented.

QUESTION 25. Consider the HTML page snippet from **Question 22** again. Which CSS snippet will generate the presentation of the content as shown below?



1. [urgent] Do the dishes

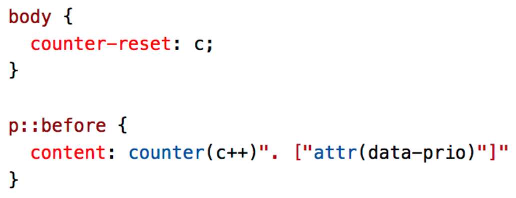
2. [neutral] Do the laundry

3. [urgent] Prepare for the resit

4. [urgent] Prepare for Q4

Copyright 2016.

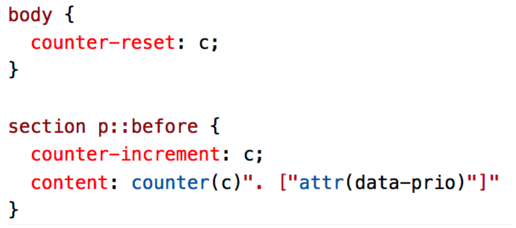
A)



```
body {
  counter-reset: c;
}

p::before {
  content: counter(c++)". ["attr(data-prio)"]"
}
```

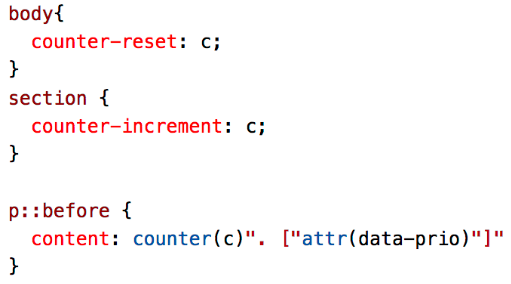
B)



```
body {
  counter-reset: c;
}

section p::before {
  counter-increment: c;
  content: counter(c)". ["attr(data-prio)"]"
}
```

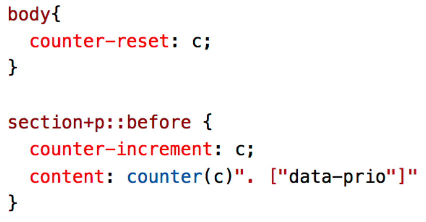
C)



```
body{
  counter-reset: c;
}
section {
  counter-increment: c;
}

p::before {
  content: counter(c)". ["attr(data-prio)"]"
}
```

D)



```
body{
  counter-reset: c;
}

section+p::before {
  counter-increment: c;
  content: counter(c)". ["data-prio"]"
}
```

Part 2 – Database

QUESTION 26. The separation of the data definition from the software application is known as:

A) Data dictionary
B) Data independence
C) Data integrity
D) Referential integrity

QUESTION 27. Which of the following items is not an advantage of a DBMS?

A) Improved ability to enforce standards
B) Improved data consistency
C) Local control over the data
D) Controlled data redundancy

QUESTION 28. The attribute (or set of attributes) that uniquely defines each row in a table is called the:

A) Identifier
B) Index
C) Primary key
D) Symmetric key

QUESTION 29. Which of the following statements about *relations* in the relational data model **are true**?

[1] NULL values represent the values of attributes that might not apply to a tuple.
[2] Tuples in a relation are, by default, ordered according to their insertion order.
[3] Attributes that are part of a superkey do not admit duplicate values.
[4] A relation might have more than one key.

A) Only [1] and [3]
B) Only [2] and [3]
C) Only [2] and [4]
D) Only [1] and [4]

QUESTION 30. Which of the following statements about the UPDATE SQL statement is correct?

A) The UPDATE statement allows updated values to be the columns' default values.
B) The UPDATE statement can be used to add a column to a table.
C) The UPDATE statement cannot be used to modify multiple tuples in a table.
D) The UPDATE statement does not allow column values to be left out from the statement.

QUESTION 31. Which of the following **are** elements of an SQL schema?

[1] Catalog
[2] Custom Constraint Types
[3] Views
[4] Indexes

A) Only [1] and [4]
B) Only [2] and [3]
C) Only [1] and [2]
D) Only [3] and [4]

QUESTION 32. Given the following relational schema,

```
Hotel(HotelName, Address, Stars, City)
```

and knowing that the attribute `Address` satisfies uniqueness constraints, how many superkeys does the `Hotel` relation have?

A) 1
B) 2
C) 4
D) 8

The following questions are related to the **Employees** database, reported in Appendix A.

SUGGESTION:detach the last page of this document, so that you can easily have it in front of you

The database can be used to manage the employment record of the employees of a big company. It includes tables for employees, departments, managers, salaries, and job titles.

Before answering the following questions: analyse the database schema, explore the properties of the tables, and understand how the tables relate to each other. Identify primary and foreign keys, and understand their constraints.

QUESTION 33. Which of the following statements best describes the result set produced by the following SQL query?

```
SELECT first_name, last_name
FROM salaries JOIN employees ON salaries.emp_no = employees.emp_no
GROUP BY salaries.emp_no
HAVING (count(*)-1) = 0
```

A) The result set contains the first name and last name of employees that never had a salary change while working at the company
B) The result set contains the first name and last name of employees that had only one salary change while working at the company
C) The result set contains the first name and last name of employees that had more than one salary change while working at the company
D) The query is syntactically incorrect: the expression in the HAVING clause refers to attributes not contained in the SELECT clause

QUESTION 34. Which of the following statements best describes the following SQL query? *Note: the DATEDIFF function returns the difference, in days, between two dates.*

```
SELECT *
FROM employees.employees
WHERE DATEDIFF('2016-12-13',hire_date) < 60
      AND DATEDIFF('2016-12-13',birth_date) < 40*365;
```

- **A)** Return the number of employees that are less than 40 years old, and that were hired in the last 60 days.
- **B)** Return the list of employees that are less than 40 years old, and that were hired in the last 60 days.
- **C)** Return the set of employees that are less than 40 years old, and that were hired for at least 60 days.
- **D)** Return the set of employees that in the last 40 years were hired for at least 60 days.

QUESTION 35. Which of the following statements best describes the following SQL query?

```
SELECT emp_no
FROM titles
GROUP BY emp_no,title
HAVING count(*) > 1 AND title = 'Technique Leader'
```

- **A)** Return the ids of employees that have the 'Technique Leader' title in the company
- **B)** Return the ids of employees that changed their title more than once and currently hold a 'Technique Leader' title
- **C)** Return the id of employees that obtained a 'Technique Leader' title more than once
- **D)** The query is syntactically wrong; a condition on attribute values expressed in the HAVING clause causes a query parsing error.

QUESTION 36. A developer is asked to count the total number of employees receiving more than 120,000 euros of salary, but only for employees that have more than 6 salary records in the database. The developer writes the following query:

```
SELECT salary, COUNT(*)as Total
FROM salaries
WHERE salary > 120000
GROUP BY emp_no
HAVING COUNT(*) > 6
```

Why is the query not correct?

- **A)** The query is not correct because the HAVING clause is not defined over the `Total` attribute.
- **B)** The query is not correct because the WHERE clause is executed before the HAVING clause, thus altering the total count of salary records of a given employee.
- **C)** The query is not correct because the HAVING clause should be written before the WHERE clause, to make sure that the count of salary records of a given employee is correct.
- **D)** The query is not correct because the `GROUP BY emp_no` instruction is executed only after counting salaries greater than 6.

QUESTION 37. Which of the following statements about the miniworld modeled by the **Employees** database is **correct**?

A) A *Department* cannot have more than one manager.
B) An *Employee* cannot hold the same Title multiple times.
C) The hire date of an *Employee* cannot be different from the first *Salary* date.
D) An *Employee* can work for two *Departments* at the same time.

QUESTION 38. Which of the following constraints **can** be enforced by the **Employees** database?

A) The Manager of a department must be employed in the same department.
B) An employee cannot work two times for the same department.
C) All *Employees* in the database must be employed in a department.
D) No two *Employees* in the database can have the same hire_date.

QUESTION 39. The MySQL DBMS hosting the **Employees** database receives the following SQL statement:

```
ALTER TABLE ‘titles’ REMOVE CONSTRAINT `titles_ibfk_1`
```

to remove the pre-existing foreign key constraint (named ‘`titles_ibfk_1`’) defined on the `emp_no` attribute. What effect does the execution of such a statement have on the database state?

A) No effect. The statement is syntactically correct.
B) All the tuples in the *employees* table related to the *titles* table are removed.
C) All the tuples in the *titles* table are removed.
D) The DBMS returns the following error: *ALTER TABLE failed because one or more objects access this column*

QUESTION 40. The following SQL statement is submitted for execution on the MySQL DBMS hosting the **Employees** database:

```
ALTER TABLE `titles` ADD CONSTRAINT `titles_ibfk_2`
    FOREIGN KEY (`from_date`) REFERENCES `employees` (`hire_date`)
```

Which of the following statements **best describe** how the execution of the statement will influence the data integrity functionality of the DBMS:

A) Deleting a tuple from the *employees* table will cause the tuples in the *titles* table having *hire_date* equal to *from_date* to be deleted.
B) All new tuples in the *titles* table must have a *from_date* value that is equal to the *hire_date* attribute value of at least one *employees* tuple.
C) Modifying a tuple from the *employees* table will cause the database to reject the operation if the modified attribute is *hire_date* .
D) The DBMS returns the following error: *Cannot add foreign key constraint* .

Appendix A – Employees Database Schema

```
CREATE TABLE employees (
    emp_no      INT             NOT NULL,
    birth_date  DATE            NOT NULL,
    first_name  VARCHAR(14)     NOT NULL,
    last_name   VARCHAR(16)     NOT NULL,
    gender      ENUM ('M','F')  NOT NULL,
    hire_date   DATE            NOT NULL,
    PRIMARY KEY (emp_no)
);

CREATE TABLE departments (
    dept_no     CHAR(4)         NOT NULL,
    dept_name   VARCHAR(40)     NOT NULL,
    PRIMARY KEY (dept_no),
    UNIQUE  KEY (dept_name)
);

CREATE TABLE dept_manager (
   emp_no       INT             NOT NULL,
   dept_no      CHAR(4)         NOT NULL,
   from_date    DATE            NOT NULL,
   to_date      DATE            NOT NULL,
   FOREIGN KEY (emp_no) REFERENCES employees (emp_no)ON DELETE CASCADE,
   FOREIGN KEY (dept_no) REFERENCES departments (dept_no)ON DELETE CASCADE,
   PRIMARY KEY (emp_no,dept_no)
);

CREATE TABLE dept_emp (
    emp_no      INT             NOT NULL,
    dept_no     CHAR(4)         NOT NULL,
    from_date   DATE            NOT NULL,
    to_date     DATE            NOT NULL,
    FOREIGN KEY (emp_no)  REFERENCES employees   (emp_no)  ON DELETE CASCADE,
    FOREIGN KEY (dept_no) REFERENCES departments (dept_no) ON DELETE CASCADE,
    PRIMARY KEY (emp_no,dept_no)
);

CREATE TABLE titles (
    emp_no      INT             NOT NULL,
    title       VARCHAR(50)     NOT NULL,
    from_date   DATE            NOT NULL,
    to_date     DATE,
    FOREIGN KEY (emp_no) REFERENCES employees (emp_no) ON DELETE CASCADE,
    PRIMARY KEY (emp_no,title, from_date)
);

CREATE TABLE salaries (
    emp_no      INT             NOT NULL,
    salary      INT             NOT NULL,
    from_date   DATE            NOT NULL,
    to_date     DATE            NOT NULL,
    FOREIGN KEY (emp_no) REFERENCES employees (emp_no) ON DELETE CASCADE,
    PRIMARY KEY (emp_no, from_date)
);
```