TECHNISCHE UNIVERSITEIT DELFT
Faculteit Elektrotechniek, Wiskunde en Informatica

Final Exam

TI1506 Web and Database Technology

Friday, 30 January 2015
09.00-11.00

INSTRUCTIONS:

- This exam consists of **2 parts** (DB and Web), and with a total of **36** 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 *studentnumber* 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.

Good Luck!

Part 1 – Databases

QUESTION 1. Which of the following modelling primitives is not part of the Entity Relationship model?

A) Primary Key Attribute
B) Complex (Composite) Attribute
C) Atomic Attribute
D) Derived Attribute

QUESTION 2. Which of the following definitions of the *degree of a relationship* in an ER diagram is correct?

A) The size of the set resulting from the Cartesian product of the participating entities.
B) The number of tuples of each participating entity.
C) The number of participating entities.
D) The number of attributes of the relationship type.

QUESTION 3. In an ER diagram, when is the specification of relationship *role names* not necessary?

A) When the relationship involves a single entity.
B) When the role of each entity participating in the relationship is not ambiguous.
C) When the relationship plays as an identifying relationship for a weak entity.
D) When the role of the relationship in the diagram can be inferred from the context.

QUESTION 4. Which of the following statements about SQL VIEWs are correct?
[1] In order to be used in other queries, the view must be always materialized in a dedicated SQL base table
[2] There are limitations to the type of SELECT operations that can be performed on a view.

A) None
B) Only [1]
C) Only [2]
D) Both [1] and [2]

QUESTION 5. How many of the following actions can be specified in the body of an SQL trigger defined over a MySQL database instance?
[1] Create new table tuple(s).
[2] Query a table of the same database as the one on which the trigger has been defined.
[3] Set temporary variables.
[4] Explicitly invoke another trigger.

A) 1
B) 2
C) 3
D) All

The next 2 questions are related to the SQL trigger depicted in Figure 1. The trigger refers to the IMDB database. You can assume that the trigger is executed in a database system with 100% SQL:2003 support.

```
1 : DELIMITER $$
2 : DROP TRIGGER IF EXISTS moviesgenres_test $$
3 : CREATE TRIGGER moviesgenres_test BEFORE INSERT ON movies_genres
4 : FOR EACH ROW
5 : BEGIN
6 :   DECLARE cur_genres FLOAT;
7 :
8 :   IF (CHAR_LENGTH(NEW.genre) = 0) THEN
9 :     SELECT 0 FROM `GENRE must not be empty` INTO @error;
10:   END IF;
11:
12:   SELECT COUNT(*) FROM movies_genres
13:     WHERE movie_id = NEW.movie_id INTO cur_genres;
14:
15:    IF (cur_genres > 2) THEN
16:        SELECT 0 FROM `Movies cannot have more than 2 genres!` INTO @error;
17:   END IF;
18: END $$
19: DELIMITER ;
```

Figure 1

QUESTION 6. Which of the following statements better describe the business logic implemented in the trigger of Figure 1?

- **A)** The trigger is activated when one or more new tuples of the `movies_genres` table are created. For each new tuple, the trigger checks if the table contains at least 2 other instances related to the movie identified by the `movie_id` attribute of the newly created tuple.
- **B)** The trigger is activated when one or more new tuples of the `movies_genres` table are created. The trigger checks if the table contains at least 2 other instances having genre name equal to the one specified for the newly created tuple;
- **C)** The trigger is activated when one or more new tuples of the `movies_genres` table are created. For each new tuple, the trigger checks if the new genre name is not empty or if the new instance relates to a movie (identified by the `movie_id` attribute of the newly created tuple) that already has more than 2 genres.
- **D)** The trigger is activated when one or more new tuples of the `movies_genres` table are created. Before a new tuple of the movies_genres table is stored, and for each new tuple, the trigger checks if the table contains at least 2 other instances related to the movie identified by the `movie_id` attribute of the newly created tuple.

C is wrong for 2 reasons: the two conditions are in AND (both must be verified), and the test is done before the tuple is inserted. D, although it does not mention the first condition, it is the most correct

QUESTION 7. Which of the following statements about the business constraints defined in Figure 1 are correct?

[1] The business constraints defined in the trigger could have also been expressed using a single `ASSERTION`.

[2] The business constraints defined in the trigger could have been expressed with one or

more attribute-based constraints.

A) None
B) Only [1]
C) Only [2]
D) Both

[1] is TRUE: both conditions mustbe expressed on the SET of the tuples in the table (e.g. the count of tuples for which genre_name length is =0, AND the count of tuples returned from an aggregation sub-query on the movie id and having count > 2)

[2] is FALSE: the second constraint is not related to the value of an attribute.

The next 2 questions are related to the SQL VIEWs depicted in Figure 2. The trigger refers to the IMDB database. Assume the VIEW to be implemented in a MySQL DB system.

```
VIEW 1

1 : CREATE OR REPLACE VIEW kolossalmovies AS
2 :     SELECT name
3 :     FROM roles JOIN movies ON movies.id = roles.movie_id
4 :     GROUP BY movies.id
5 :     HAVING count(DISTINCT actor_id) > 200


VIEW 2

1 : CREATE OR REPLACE VIEW reevesActors AS
2 :     SELECT first_name, last_name FROM actors
3 :      WHERE last_name = "Reeves" WITH LOCAL CHECK OPTION;
```

Figure 2

QUESTION 8. Which of the following statements is correct?

A) Trying to perform an UPDATE operation on `reevesActors` may produce an error only if the update is performed on the `last_name` attribute
B) Trying to perform an UPDATE operation on `kolossalmovies` may produce an error only if the update is performed on the `movies.id` attribute
C) Trying to perform an UPDATE operation on `reevesActors` may produce an error only if the update is performed on the `first_name` attribute
D) Trying to perform an UPDATE operation on `kolossalmovies` will produce an error only if the update is performed on the `actor_id` attribute.

A is TRUE because of the CHECK OPTIONS clause.

QUESTION 9. Consider the following SQL query.

```
SELECT *
FROM FROM kolossalmovies JOIN movies ON kolossalmovies.name = movies.name
WHERE movies.id IN (SELECT movie_id FROM movies_genres WHERE genre = 'Drama')
```

Assume the IMDB database to be installed on a standard Macbook Air 13” from 2014. Which, among the following statements, is more likely to be correct?

A) Query execution will be fast because the `kolossalmovies` uses the MERGE execution strategy
B) Query execution will be fast because the `kolossalmovies` uses the TEMPTABLE execution strategy
C) Query execution will be slow because the `kolossalmovies` uses the MERGE execution strategy
D) Query execution will be slow because the `kolossalmovies` uses the TEMPTABLE execution strategy

A and C are FALSE: The view contains an aggregation. Therefore TEMPTABLE is the default algorithm
B is FALSE: TEMPTABLE is slower

The next 4 questions are related to the EER diagram depicted in Figure 3. The trigger is defined over the IMDB database. Remember: we use the words "Entity" and "Class" as synonyms.

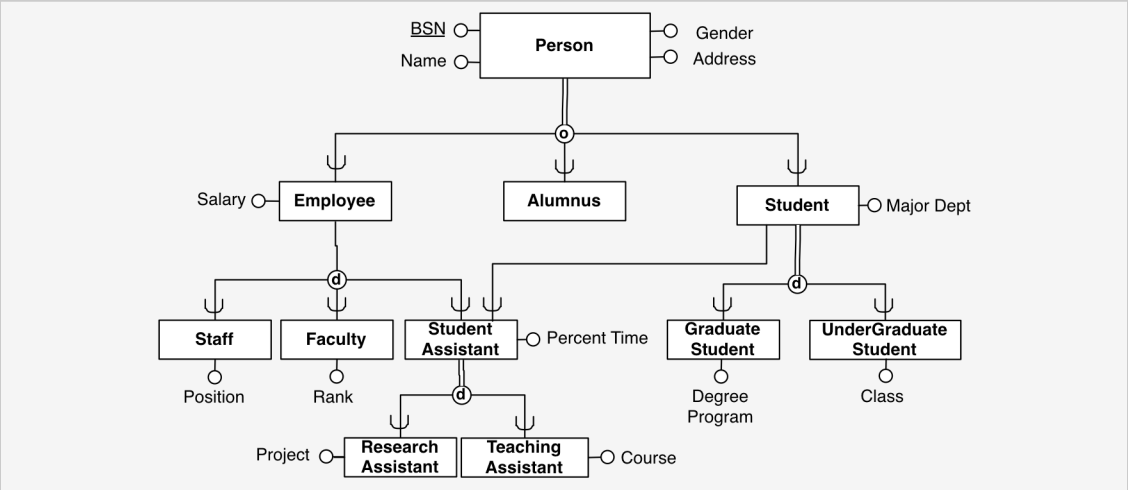


Figure 3

QUESTION 10. Consider the diagram above. Which of the following statements are correct?
[1] The diagram in Figure 3 is a specialization lattice because some entities are shared subclasses
[2] The diagram in Figure 3 is a specialization lattice because every subclass participates as a subclass in only one class/subclass relationship

A) None
B) Only [1]
C) Only [2]
D) Both

QUESTION 11. Consider the diagram above. Which of the following statements are correct?
[1] A member of the `Research Assistant` entity can also be a member of the `Alumnus` entity
[2] A member of the `Teaching Assistant` entity cannot also be a member of the `Graduate Student` entity

A) None
B) Only [1]
C) Only [2]
D) Both

QUESTION 12. Consider the diagram above. Which of the following statements are correct?

[1] Due to multiple inheritance, a member of the `Teaching Assistant` entity includes the `Salary` attribute or the `Major Dept` attribute
[2] Due to multiple inheritance, a member of the `Teaching Assistant` entity includes two `Name` attributes.

A) None
B) Only [1]
C) Only [2]
D) Both

QUESTION 13. Which of the following statements does not correctly describe the mini-world modelled by the EER diagram in Figure 3?

A) All person entities represented in the database are members of the `Person` entity type.
B) All members of the `Employee` entity must be members of the `Staff`, `Faculty`, or `Student Assistant` entities
C) An `Employee` cannot be a `Staff` member and a `Faculty` member at the same time.
D) All members of the `Student Assistant` entity are both members of the `Employee` and `Student` entities.

The next 5 questions are related to the EER diagram in Figure 4, drawn according to the notation used in the course's slides. The diagram represents a database about programming languages.

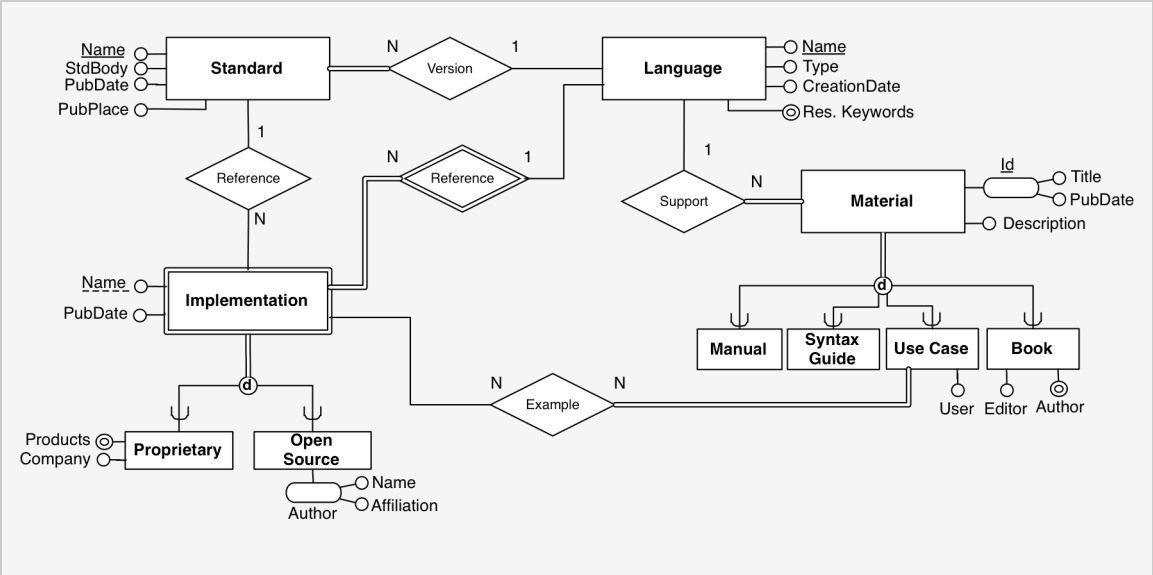


Figure 4

QUESTION 14. Which of the following statements about the diagram in Figure 4 are correct?

[1] For every instance I_n of the `Implementation` entity that is related to a `Use Case` entity instance U_m, there exists an instance L_o in the `Language` entity that is related to both I_n and U_m.

[2] An instance S_n of the `Standard` can be related to an instance L_o in the `Language` entity, but must be related to at least one instance I_n of the `Implementation` entity

A) None
B) Only [1]
C) Only [2]
D) Both

QUESTION 15. Consider the EER diagram in Figure 4. Which of the following constraints is not expressed?

A) An implementation of a given programming language must be either open source or proprietary.
B) A standard must be described in at least one book.
C) The syntax of a programing language might be described in a syntax guide.
D) A support material must be related to at most one programming language.

QUESTION 16. Consider the EER diagram in Figure 4. How many internal identifiers are defined?

A) 2
B) 3
C) 4
D) 5

IMPLEMENTATION name is not, alone, an internal identifier, as it requires an attribute. Also, title and isbn in MATERIAL work as a single identifier

QUESTION 17. Consider the restructuring of the EER diagram in Figure 4, performed as a preliminary operation functional to the translation into a relational model. The restructuring is performed according to the standard method presented during lectures. However, instead of the standard specialization removal method, you are asked to use the *child-collapsing* method. Which is the minimum number of tables resulting from the transformation?

A) 8
B) 13
C) 14
D) 16

QUESTION 18. Consider the same EER restructuring scenario of Question 17. This time, you are asked to use a *parent-collapsing* specialization removal method. Which of the following statements correctly describe the resulting EER diagram?

[1] The `Material` entity participates in the `Example` relationship with a total participation constraint.
[2] The `Implementation` entity includes 5 attributes, but only the `Author` attribute becomes optional because it is multi-valued
[3] The `Implementation` entity features 6 attributes, but only 3 become optional
[4] The `Example` relationship changes its cardinality from N:N to 1:N because not all the instances from the Material entities will be related to an instance of the `Implementation` entity

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

Part 2 – Web

QUESTION 19. Consider the HTML snippet shown below:

```
<main>
        <h2 id="todoHeader">Todos</h2>
        <p class="urgent" data-numTodos="5 items">Today's todos</p>
                <ul>
                        <li>Grocery shopping</li>
                        <li>Web assignment</li>
                        <li>Logic assignment</li>
                        <li>OOP assignment</li>
                        <li>Meetup Amsterdam</li>
                </ul>
        <p class="urgent" data-numTodos="2 items">Tomorrow's todos</p>
                <ul>
                        <li>Meetup Delft</li>
                        <li>OOP assignment</li>
                </ul>
        <p data-numTodos="No items" >Saturday's todos</p>
        <p data-numTodos="No items" >Sunday's todos</p>
</main>
```

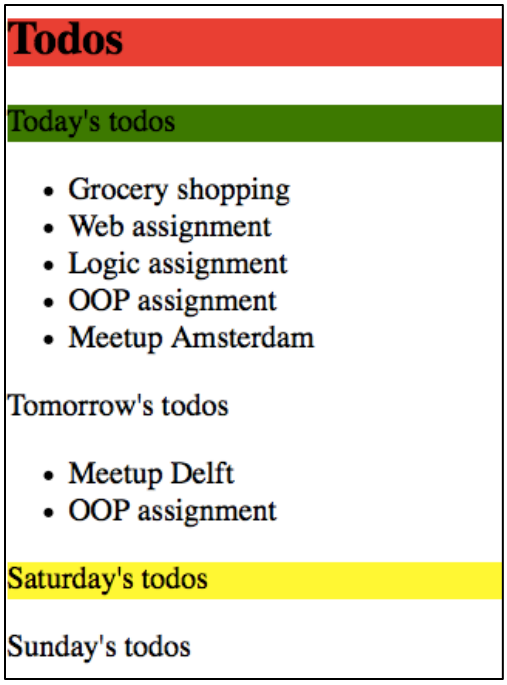
Select the CSS code that will result in the following styled page:

A)
```
p:nth-child(2) {
    background: green;
}
p:nth-of-type(3) {
    background: yellow;
}
#todoHeader {
    background: red;
}
```

B)
```
p:nth-child(2) {
    background: green;
}
p:nth-of-type(4) {
    background: yellow;
}
.todoHeader {
    background: red;
}
```

C)
```
p:nth-child(1) {
    background: green;
}
p:nth-of-type(3) {
    background: yellow;
}
#todoHeader {
    background: red;
}
```

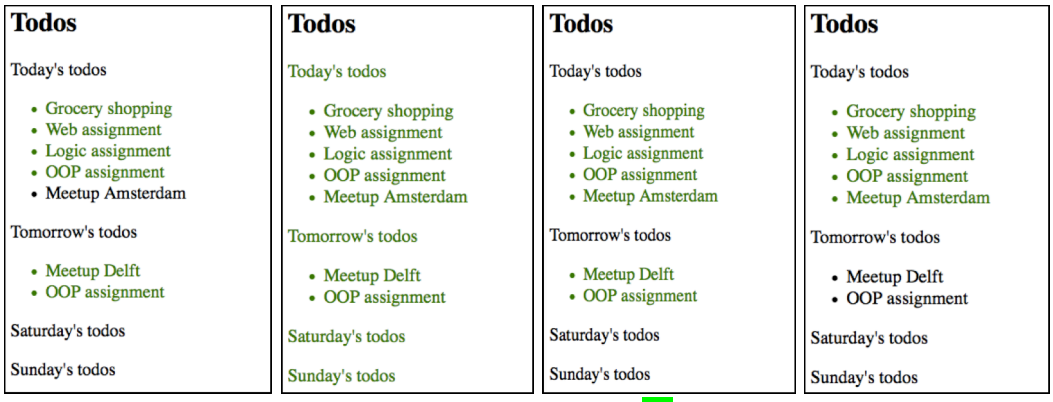
D)
```
p:nth-child(3) {
    background: green;
}
p:nth-of-type(3) {
    background: yellow;
}
#todoHeader {
    background: red;
}
```



QUESTION 20. Consider the HTML snippet of question 19 again. The following CSS is applied to it:

```
main ul :not(.urgent) {
    color: green;
}
```

What will be the result of this style?



A) **B)** **C)** **D)**

QUESTION 21. Consider the HTML snippet shown below:

```
<main>
        <form action="" method="post">
                <input id="todoItem" type="text" placeholder="Add your todo" />
                <label for="todoItem"> </label>

                <input id="urgency" name="urgency" type="number" min="1" max="5"
                                        placeholder="Level of urgency" required />
                <label for="urgency"> </label>

                <input id="todoSubmit" type="submit" value = "Add Todo"/>
                <label for="todoSubmit"> </label>
        </form>
</main>
```

If the user provides an input for the level of urgency that is neither of {1,2,3,4,5}, you want to provide the user with a warning message, which should appear immediately to the right of this input field. Which CSS selector achieves this behaviour?

A)
```
#urgency:invalid label::after {
        /* style added here */
}
```

B)
```
input[type=number]:invalid + label::after {
        /* style added here */
}
```

C)
```
#urgency:invalid:error + label {
        /* style added here */
}
```

D)
```
input[type=number]:error label::after {
        /* style added here */
}
```

QUESTION 22. Consider the HTML of **question 19**, in particular the `data-numTodos` attribute. Rendering the HTML without any CSS will not show the content of these attributes. Which CSS snippet will achieve the following rendering:

Todos

Today's todos (5 items)

- Grocery shopping
- Web assignment
- Logic assignment
- OOP assignment
- Meetup Amsterdam

Tomorrow's todos (2 items)

- Meetup Delft
- OOP assignment

Saturday's todos (No items)

Sunday's todos (No items)

A)
```
p::after {
    content: " (" attr(data-numTodos) ")";
}
```

B)
```
p ::after {
    content: " (" attribute(data-numTodos) ")";
}
```

C)
```
p::after {
    config: " (" attr(data-numTodos) ")";
}
```

D)
```
p ::after {
    config: " (" attribute(data-numTodos) ")";
}
```

QUESTION 23. Lets consider the HTML example of **question 19** and the following CSS snippet:

```
ul {
    counter-reset: countTodo;
    list-style-type: none;
}
li::before {
    counter-increment: countTodo;
    content: counters(countTodo,".") ": ";!
}
```

How will the HTML + CSS be rendered?

A)

Todos

Today's todos

1: Grocery shopping
2: Web assignment
3: Logic assignment
4: OOP assignment
5: Meetup Amsterdam

Tomorrow's todos

6: Meetup Delft
7: OOP assignment

Saturday's todos

Sunday's todos

B)

Todos

Today's todos

1.1: Grocery shopping
1.2: Web assignment
1.3: Logic assignment
1.4: OOP assignment
1.5: Meetup Amsterdam

Tomorrow's todos

2.1: Meetup Delft
2.2: OOP assignment

Saturday's todos

Sunday's todos

C)

Todos

Today's todos

1.1: Grocery shopping
1.2: Web assignment
1.3: Logic assignment
1.4: OOP assignment
1.5: Meetup Amsterdam

Tomorrow's todos

1.6: Meetup Delft
1.7: OOP assignment

Saturday's todos

Sunday's todos

D)

Todos

Today's todos

1: Grocery shopping
2: Web assignment
3: Logic assignment
4: OOP assignment
5: Meetup Amsterdam

Tomorrow's todos

1: Meetup Delft
2: OOP assignment

Saturday's todos

Sunday's todos

QUESTION 24. Which of the following statements is true about CSS animations and transitions?

A) A CSS animation is a CSS transition with 3 states: start, middle and end state.
B) A CSS transition is a CSS animation with 2 states: start and end state.
C) CSS animations and CSS transitions are synonyms, and cover exactly the same concepts.
D) CSS animations are used to animate pseudo-classes and pseudo-elements only, CSS transitions can animate additional elements.

QUESTION 25. Consider the Figure 5 below. It contains a schematic overview of an Ajax request with some important information missing. What concepts do X and Y refer to?

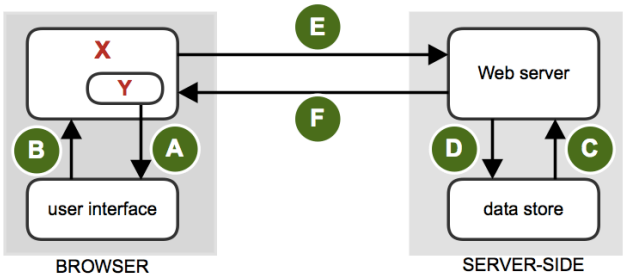


Figure 5

A) X: HttpRequest, Y: HTMLRequest
B) X: XMLHttpRequest, Y: HttpRequest
C) X: HtmlRequest, Y: callback
D) X: XMLHttpRequest, Y: callback

QUESTION 26. Consider Figure 5 above. Which of the following lists the correct sequence of actions taken by the client and the server when an Ajax request is issued?

A) A, B, E, D, C, F
B) F, A, B, E, D, C
C) D, C, F, A, B, E
D) B, E, D, C, F, A

QUESTION 27. Consider the node.js script shown below:

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

function restrict(req, res, next) {
    var authorization = req.headers.authorization;
    if (!authorization)
        return next();

    var parts = authorization.split(' ')
        var scheme = parts[0]
        var auth = new Buffer(parts[1], 'base64').toString().split(':')
        var user = auth[0]
        var pass = auth[1];

        //normally: authenticate via user information stored in DB
        //here: match for user/password
        if(user == "user" && pass == "password") {
            next();
        }
}

function logger(request, response, next) {
    console.log('%s\t%s\t%s', new Date(), request.method, request.url);
    next();
}

function helloWorld(request, response, next) {
    response.setHeader('Content-Type', 'text/plain');
    response.end('Hello World!');
}

var app = connect();
app.use(logger);
app.use('/admin',restrict);
app.use(helloWorld);
app.listen(4403);
```

Assume you run this code on your local machine (localhost) and open your Web browser with the following URL: `http://localhost:4403/admin`
What will happen?

A) The server returns a response containing "Hello World!"
B) The server returns an error message: unauthorized access
C) The server returns an empty page
D) The server will not return anything

QUESTION 28. Consider the `node.js` script and the `ejs` file shown below:

```
var express = require("express");
var url = require("url");
var http = require("http");
var app;

app = express();
http.createServer(app).listen(3005);

var ts = [];
ts.push({ message: 'Final exam', dueDate: '30/01/2015, 2pm', difficulty: '5'  });
ts.push({ message: 'Resit', dueDate: '05/03/2015, 5pm', difficulty: '3' });
ts.push({ message: 'Sign up for classes', dueDate: '06/02/2015, 3pm', difficulty: '1'
});

app.set('views', __dirname + '/views');
app.set('view engine', 'ejs');

app.get("/showlist", function (req, res) {
    res.render('showlist', { title: 'My todo list', data_array: ts });
});
```

```
<!DOCTYPE html>
<html>
        <head><title><%= title %></title></head>
        <body>
                <h1>A list of todos</h1>
                <div id="todos">
                        <% for(var i=0; i<data_array.length; i++) {
                                var todo = data_array[i];
                        %>
                        <div class="todo">
                                        ?????
                                        ?????
                                        ?????
                        </div>
                <% } %>
        </div>
        </body>
</html>
```

The node.js script is started on localhost and you can assume that the `ejs` file is available in the folder views. You open your Web browser at `http://localhost:3005/showlist` and see the following rendering:

Which lines of `ejs` code (missing in the EJS file above) will make this a reality?

A list of todos

Due at: 30/01/2015, 2pmCET

Item: Final exam...

Level: 5

Due at: 05/03/2015, 5pmCET

Item: Resit...

Level: 3

Due at: 06/02/2015, 3pmCET

Item: Sign up fo...

Level: 1

A)
```
<p>Due at: <%=: todo.dueDate | append:' CET' %></p>
<p>Item: <%=: todo.message|truncate:10 %>...</p>
<p>Level: <%=: todo.difficulty| last %></p>
```

B)
```
<p>Due at: <%= todo.dueDate | append: CET %></p>
<p>Item: <%= todo.message|truncate:15 %>...</p>
<p>Level: <%= todo.difficulty| last %></p>
```

C)
```
<p>Due at: <%: todo.dueDate | append: CET %></p>
<p>Item: <%: todo.message  %>...</p>
<p>Level: <% todo.difficulty %></p>
```

D)
```
<p>Due at: < todo.dueDate | append:' CET' ></p>
<p>Item: < todo.message|truncate:10 >...</p>
<p>Level: < todo.difficulty| last ></p>
```

QUESTION 29. The browser B currently has not stored any cookies. The server sends the following cookies to B:

```
Set-Cookie: bg=blue; Expires=Tue, 20-Jan-2016 21:47:38 GMT; Path=/; Domain=.tudelft.nl; HttpOnly
Set-Cookie: fs=233; Path=/; Domain=.tudelft.nl;
Set-Cookie: br=cr; Expires=Thu, 01-Jan-1970 00:00:01 GMT; Path=/; Domain=.tudelft.nl;
Set-Cookie: hq=342322; Path=/; Domain=.tudelft.nl
```

Browser B crashes 5 minutes later and the user restarts B. How many of these cookies can the user not access after the restart with client-side JavaScript?

A) All 4 cookies can be accessed
B) 1
C) 2
D) 3

QUESTION 30. After accessing the following URL http://login.meebo.com for the first time, the server sent the following cookies to browser B:

```
Set-Cookie: ID1=32sfs32; Path=/todos; Expires=Fri, 30 Jan 2015 01:01:01 GMT; Secure; HttpOnly
Set-Cookie: ID2=532aaaa; Domain=meebo.com; Path=/; Expires=Fri, 30 Jan 2015 01:01:01 GMT; HttpOnly
Set-Cookie: ID3=ssd33dd; Domain=login.meebo.com; Path=/; Expires=Fri, 30 Jan 2015 01:01:01 GMT; HttpOnly
```

Next, browser B tries to access the following URL: `http://www.meebo.com/todos`. How many of the three cookies shown above are sent back to the server?

A) 0
B) 1
C) 2
D) 3

QUESTION 31. Consider the following node.js script running at `todos.meebo.com:3007`

```
var express = require("express");
var http = require("http");
var credentials = require('./credentials.js');
var cookies = require("cookie-parser");
var sessions = require('express-session');
var app = express();
app.use(cookies(credentials.cookieSecret));
app.use(sessions(credentials.cookieSecret));
http.createServer(app).listen(3007);

app.get("/visiting", function (req, res) {
        var session = req.session;
        if(session.views) {
                session.views++;
                res.send("You again!");
        }
        else {
                session.views = 1;
                res.send("A newcomer!");
        }
});
```

A user starts up his browser (which contains no cookies so far) and accesses:

- `http://todos.meebo.com:3007/login`

He closes the browser, starts it up again and accesses

- `http://todos.meebo.com:3007/visiting`

He then tries to access

- `http://todos.meebo.com:3007/mytodos`

How many times in the process does the browser send cookies to the server?

A) 0
B) 1
C) 2
D) 3

QUESTION 32. Consider the following list of abilities a malicious user (the attacker) may have who managed to intercept all of your network traffic

[1] The attacker can eavesdrop (read all your HTTP requests)
[2] The attacker can inject additional HTTP requests with your source address
[3] The attacker can modify HTTP requests
[4] The attacker can drop HTTP requests

Which of the listed abilities is needed to steal one of your session cookies?

A) Only [1]
B) [1] and [3]
C) Only [2]
D) None of these abilities is required

QUESTION 33. Consider the setup and list of attacker abilities given in question 32 again. Which of the listed abilities is needed to perform a reflected XSS attack on you?

A) Only [1]
B) [1] and [3]
C) Only [2]
D) None of these abilities is required

QUESTION 34. You built a Web application that contains a 'user information page' (accessible at `/retrieveUserData`) where a user is provided with two form fields and asked for username and password. With the right username/password combination the user can retrieve all the information the application has stored about him.
You implemented the following node.js code.

```
app.get("/retrieveUserData", function (req, res) {
        var query = url.parse(req.url, true).query;
        var uname = ( query["username"]!=undefined) ? query["username"] : "";
        var passw = ( query["password"]!=undefined) ? query["password"] : "";
        var sqlQuery = "SELECT * FROM users WHERE name='"+uname+"' AND
password='"+passw+"'";
        /*
         * query the SQL backend with the created query and return the result to the
client
         */
});
```

The table users was created as follows:

```
CREATE TABLE `users` (
  `name` varchar(255) DEFAULT NULL,
  `password` varchar(255) DEFAULT NULL,
  `extra_info` varchar(3000) DEFAULT NULL
);
```

A malicious user wants to retrieve not just the information stored about himself, but about all other users as well and tries a number of inputs:

[1] username: john
password: 1111' or 'a'='a

[2] username: john' or 'a'='a'
password:

[3] username:
password: 1111' or 'a'=='a

[4] username: john
password: 1111' or TRUE

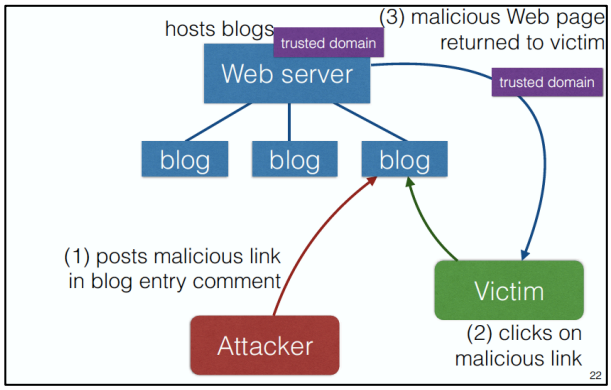
Which of them will return the desired information?

A) 1)
B) 3)
C) 1) and 2)
D) 1), 2) and 4)

QUESTION 35. Assume you have an account at `bol.com`. When you log in with your account name and password, you receive a session ID (transmitted through a cookie), which may look as follows: `D5D96C4F4F6B75542D0368BB5550FC2D`. What is the purpose of creating such long and complex session IDs? Why are we not simply using as sessionID the combination of username and timestamp of the login?

A) Session IDs that contain the username and other predictable information can quickly be found by an attacker through a brute-force approach.
B) Session IDs that contain the username are vulnerable to CRFS attacks.
C) Session IDs that are complex make it more difficult for an attacker to extract them from cookies.
D) Session IDs that are complex are less vulnerable to unvalidated redirects than predictable session IDs.

QUESTION 36. Consider the scenario sketched in the image below. Which attack type is shown here?



A) XSS
B) Path traversal
C) Session hijacking
D) CSRF