UNIVERSITETET I BERGEN

KANDIDAT

120

PRØVE

INF226 0 Programvaresikkerhet

Emnekode	INF226
Vurderingsform	Skriftlig eksamen
Starttid	30.11.2023 14:00
Sluttid	30.11.2023 17:00
Sensurfrist	--
PDF opprettet	03.05.2024 11:03

[Practicalities]

Oppgave	Tittel	Oppgavetype
i	Practical information about the exam	Informasjon eller ressurser

[Exam Questions]

Oppgave	Tittel	Oppgavetype
1	1. Selected Questions [5%]	Nedtrekk
2	2. Cookies [5%]	Flervalg (flere svar)
3	3. Digital Exam [20%]	Langsvar
4	4. Escaping Consequences [15%]	Langsvar
5	5. Secure Gifting [10%]	Langsvar
6	6. HeadBook [5%]	Langsvar

[Not Exam Questions]

Oppgave	Tittel	Oppgavetype
7	7. Oblig [40%]	Muntlig
8	8. Multiple Choice Demo [0%]	Flervalg

1 1. Selected Questions [5%]

Pick the correct alternative for each statement. 1 point per correct answers, -0.5 points for incorrect answers and 0 points for no answer.

a)

For a URL to be considered same-origin *(strict, same-origin, same-site, secure)* , it must

have the same *protocol*, *port* and *host* as the current page, but a same-site (not encrypted, same-site, lax, same-origin) URL can have a different host name as long as it belongs to the same domain as the current page.

(For example, `https://uib.no/`, `https://git.app.uib.no/` *and* `https://mitt.uib.no/`
all have the same protocol, port and domain name, but the host names are different.)

b)
To make sure that a cookie is only sent when the host name matches exactly, you should

use the SameSite=strict attribute (use the SameSite=origin attribute, use the Secure attribute, leave the Domain attribute unset, use the SameSite=strict attribute) .

c)

According to best practice, authentication should be done using two-factor authentication (two-factor authentication, biometrics, OpenID Connect, shared secrets).

d)

The attack vector (threat model, risk profile, attack surface, attack vector) is all the possible ways a malicious user might use to break in to a system.

Maks poeng: 5

Knytte håndtegninger til denne oppgaven?
Bruk følgende kode:

7 6 4 0 9 0 5

2 2. Cookies [5%]

(For each question, up to 2 points / -1 points (respectively) for correct / incorrect answers; an extra point if everything is correct.)

a)

Assume that the web page at `uib.no` sets a cookie for the `uib.no` domain with `SameSite=Lax`. **In which of the following cases will the browser include the cookie with an HTTP request?** You can assume that the browser follows the newest standards, and that no CORS headers are in use.

Select one or more alternatives:

☑ A page at `mitt.uib.no` includes an image from `uib.no`

□ A page at `uib.no` includes an image from `mitt.uib.no`

□ A page at `samordnaopptak.no` includes an image from `uib.no`

☑ The user is visiting `samordnaopptak.no` and follows a link to `uib.no`

□ The user is visiting `samordnaopptak.no` and submits a form (with method=POST) to `uib.no`

b)

Assume that the web page at `uib.no` sets a cookie for the `uib.no` domain with `SameSite=Strict`. **In which of the following cases will the browser include the cookie with an HTTP request?** You can assume that the browser follows the newest standards, and that no CORS headers are in use.

(Same as above, except with SameSite=Strict)

Select one or more alternatives:

☑ A page at `mitt.uib.no` includes an image from `uib.no`

□ A page at `uib.no` includes an image from `mitt.uib.no`

□ A page at `samordnaopptak.no` includes an image from `uib.no`

□ The user is visiting `samordnaopptak.no` and follows a link to `uib.no`

□ The user is visiting `samordnaopptak.no` and submits a form (with method=POST) to `uib.no`

Maks poeng: 5

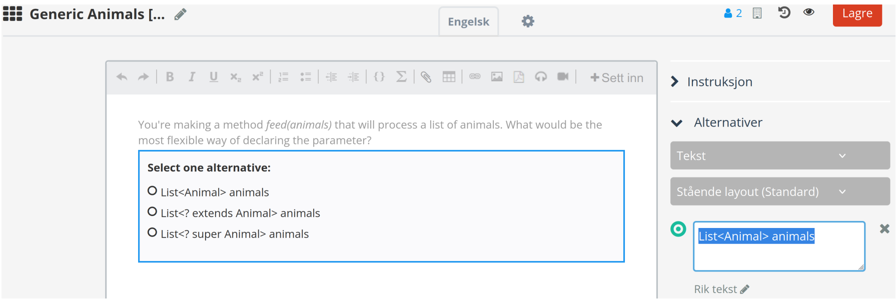
Knytte håndtegninger til denne oppgaven?
Bruk følgende kode:

0 8 3 5 7 0 6

3 3. Digital Exam [20%]

[This question has parts a)–e) + a bonus question.]

After graduation, the University hires you as a lecturer. Your first job is to make the exam for a Java programming course. You want to test the students' understanding of Java generics (e.g., `List<T>`*), so you make a multiple choice question like this:*



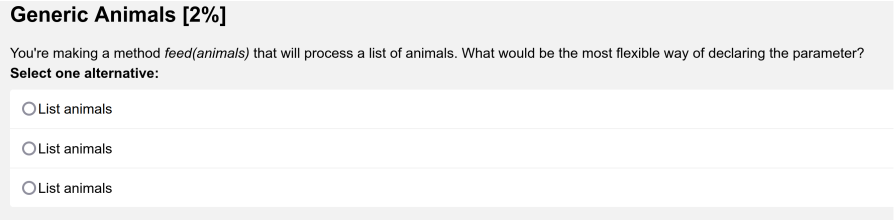
You're making a method *feed(animals)* that will process a list of animals. What would be the most flexible way of declaring the parameter?

Select one alternative:

○ List<Animal> animals
○ List<? extends Animal> animals
○ List<? super Animal> animals

List<Animal> animals

However, at the exam, confused students complain that all the answer alternatives look exactly alike:



Generic Animals [2%]

You're making a method *feed(animals)* that will process a list of animals. What would be the most flexible way of declaring the parameter?
Select one alternative:

○ List animals

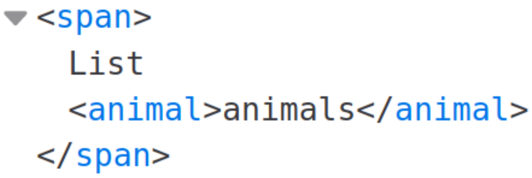
○ List animals

○ List animals

***a)* [3%]**

Even though everything looked OK in your editor, when the question is displayed to the candidates, any text between angle brackets ("<…>") disappears. You suspect the exam software[1] might be filtering anything that looks like an HTML tag. Why would it do this? **Explain.**

Upon closer inspection you notice that there isn't actually any filtering going on; the text disappears because the browser thinks it's an HTML tag – it doesn't know what to do with the `<animal>` *tag, so it just ignores it:*



```
▼ <span>
    List
    <animal>animals</animal>
  </span>
```

It turns out you can in fact insert arbitrary HTML into a multiple choice alternative and the browser will treat it as HTML – it will even let you execute JavaScript code!

***b)* [1%]**

Let's assume that an attacker is able to exploit this attack vector to run malicious JavaScript code on the candidates' computers. **What do we call this type of attack?** *(1 sentence)*

***c)* [6%]**

Thinking in terms of security properties like CIA(-T), STRIDE and DREAD, what are (some of) the security implications/threats of this attack vector? I.e., how could this be exploited / what could go wrong? **Explain.** *(1–3 paragraphs)*

d) **[7%]**

How would you mitigate this threat / prevent this sort of attack from happening? **Explain, and mention multiple possibilities if you can.** *(1–2 paragraphs)*

e) **[3%]**

You contact the developer (through a secure channel, of course) to inform them of the problem. They claim[2] that the system is working as intended; it's useful to be able to insert arbitrary HTML code into the exam, and it's perfectly safe since only the lecturer and administrators are able to edit the questions. **Do you agree? Explain.** *(~1 paragraph)*

z) **[bonus+2%]**

While you're experimenting, you notice that any JavaScript code you insert using the `<script>` tag won't run, even though the script element gets properly inserted on the page (i.e., in the DOM). This is not surprising, since such scripts will only run while the page is loading, but the exam system loads the questions separately and inserts them later.

If you were the attacker, how would you get the browser to run your code without using the `<script>` tag? **Show an example and explain how it works.**

[1] Any similarity to real digital exam software that you may be familiar with is of course purely coincidental... For a demonstration of the problem, see "Multiple Choice Demo [0%]" at the end of this exam.

[2] The vendor's response is completely fictional, made to fit the exam question, and is not based on any real or imagined interaction with any actual software vendor.

Fill in your answer here

A: they might filter out text inside angle brackets as a poor way of trying to prevent Cross-site scripting. such as preventing any <script> tags from being run in the users browser. they would do this because they can not trust the user input from the person writing the exam.

B: this would be called a Cross-site scripting attack, in which user input is ran as javascript code. possibly even on other users browsers.

C:

The first security implication is Confidentiality. an attacker could run code that reads confidential information when the user inputs it, and sends it to the attackers server so they can read it.

They can also attack the integrity of the data, by tampering with it between when the user inputs it, and when it gets sent to the server. this would mean that we could no longer trust that the data is accurate.

This would also break repudiation, since a user can for example dispute that they input the wrong answer, instead insisting that the answer they input was tampered with after the fact.

D:

We can distrust all user input, and make sure to sanitize / escape any user input that we want to display in html. this will prevent the input from being interpreted as html as opposed to text. we can also use innerText, instead of InnerHtml when inserting via javascript. or we can insert user text into the html on the server instead of in the browser via javascript.

We can also set a Content security policy that prevents javascript from being ran.

E:

No we can never trust user input. There is no way of knowing that the lecturer has not been hacked, in which case a third party pretending to be the lecturer (spoofing) can be inputting malicious code. but the lecturer themselves might also have malicious intent.

z:

i would try to insert an img tag instead with a onerror attribute that contains my code, i would also make sure that the src points to a nonsense adress so that the img tag runs my onerror code.

```
<img src="qwerty" onError=(alert("remote code execution")) />
```

Ord: 355

Maks poeng: 22

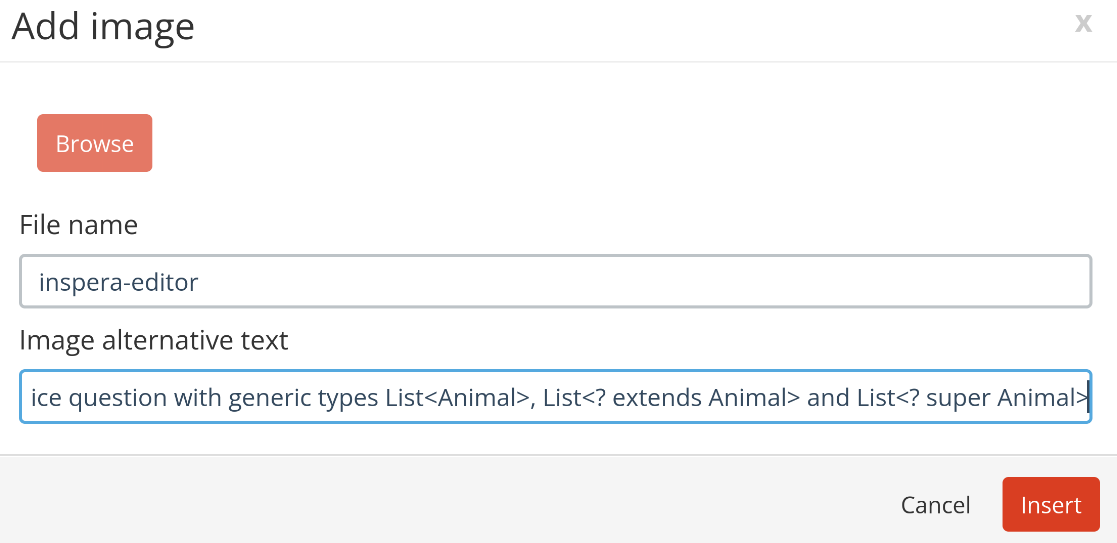
Knytte håndtegninger til denne oppgaven?
Bruk følgende kode:

9106315

4 4. Escaping Consequences [15%]

[This question has parts a)–e) + a bonus question.]

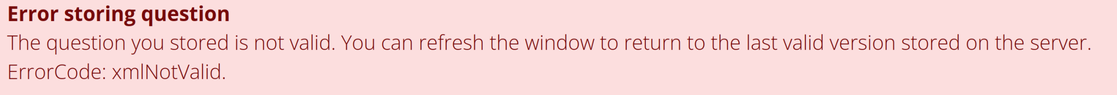
The Department is very happy with your work as a Java lecturer, so they hire you to teach software security as well. While making the exam, you think of the digital exam problem and realise it would make an interesting exam question. As part of the question, you include a screenshot of the exam system. Since you're mindful of accessibility concerns, you also include descriptive `alt` *text for the image, so it'll also make sense if the candidate is using a screen reader:*



File name: inspera-editor

Image alternative text: ice question with generic types List<Animal>, List<? extends Animal> and List<? super Animal>

Once you're done and hit "Save", you're confronted by this error message:



Error storing question
The question you stored is not valid. You can refresh the window to return to the last valid version stored on the server. ErrorCode: xmlNotValid.

The error disappears if you remove the image.

***a)* [2%]**
What could be the problem here? **Explain briefly.**
(1–3 sentences)

***b)* [3%]**
You add another image, with the alt text "`Multiple choice questions where all alternatives are just "List"`*". This works, with no "invalid XML" errors. However, the image doesn't actually appear in the exam, instead you get a truncated version of the alt text:*

> Multiple choice questions where all alternatives are just

Looking at the HTML code, you see that it's a bit garbled:

```
<img alt="Multiple choice questions where all alternatives are just "
     class="insertedImage" list="">
```

Why do you think this happens? **Explain briefly.**

(1–3 sentences)

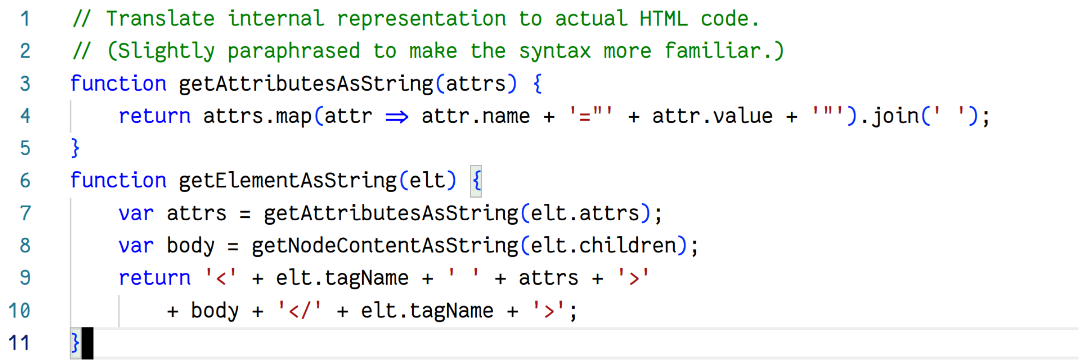
***c)* [4%]**

As seen in *a)*, the system prevents you from using < or > in the alt text, but the handling of double quotes (`"…"`) seems buggy. **Make an example of an alt text you could enter into the *Add Image* form to exploit this bug.**

(A line of code + 1–2 sentences explanation. It may be difficult to guess exactly what happens without trying it, so you can make your own assumptions about behaviour.

***d)* [3%]**

You try to debug the issue, and come across the code that renders exam questions on screen. Apparently, the exam questions are loaded dynamically from the server, stored in some internal representation, and then serialised to HTML text and loaded into the document via `innerHTML` *(or similar):*



```
// Translate internal representation to actual HTML code.
// (Slightly paraphrased to make the syntax more familiar.)
function getAttributesAsString(attrs) {
    return attrs.map(attr => attr.name + '="' + attr.value + '"').join(' ');
}
function getElementAsString(elt) {
    var attrs = getAttributesAsString(elt.attrs);
    var body = getNodeContentAsString(elt.children);
    return '<' + elt.tagName + ' ' + attrs + '>'
        + body + '</' + elt.tagName + '>';
}
```

You decide to contact the vendor and offer some hints on how to improve the system.

What would be a safe way of setting attribute values on elements, given that the values come from an untrusted source?

(1–2 paragraphs)

***e)* [3%]**

After digging around a bit more, you find:

- Alt text when inserting an image seems unsafe (as seen above)
- If you right-click an image in the editor and select *Image Properties*, you can set many of the attributes for the image tag, including `src`, `alt`, `width`/`height`, `class` and `style`.
- In this case, quotes in the alt text seem to be handled safely.
- For the style attribute, the user interface requires you to enter semicolon-separated key-value pairs (`key1:value1; key2:value2`). Your exploit attempts result in either a warning, and “invalid XML” error, or some parts being dropped.
- There doesn’t seem to be an obvious way to exploit any of the other settable attributes.

The vendor offers[1] to fix the quotes-in-alt-text-while-inserting-image bug by using a well-reputed library to escape the quote characters before saving to the server. They don’t believe the *Image Properties* can be exploited, so it will remain unchanged.

Do you believe this is a satisfactory solution? Explain.

(~1 paragraph)

[1] *The vendor's response is completely fictional, made to fit the exam question, and is not based on any real or imagined interaction with any actual software vendor.*

***z)* [bonus+2%]**

Are there any HTML element attributes that are considered generally unsafe for user input?

(1–3 sentences)

Fill in your answer here

a: the angle brackets <> are interpreted as an xml tag, and this breaks the xml format since there are no closing tags for the ones in the input.

B:

the user input is not escaped / sanitized properly, so the quotes in the input is injected directly into the code, thereby closing the alt text early. the qoute that was supposed to close the alt text will instead be used to close the list attribute on the img tag. the list attribute might stop the image from being shown.

C:

user input = some alt text" onError=(alert('code injection')) src='kløjsdf'

this will set the alt text to "some alt text", and register an alert in the case there is an error during the loading of the image. i also set the src for the image to some random text, to stop the image from loading properly.

D:

the first thing to do is to sanitize the attributes, so that for example the quotes gets treated as part of a string instead of as code.

the second thing to do would be to use javascripts inbuilt methods for inserting attributes, instead of writing code to dynamically construct an html tag.

It would also maybe be better to construct the html on the server via templates, instead of doing it in the browser.

E:

No, i would not be satisfied with only handling the alt text. they should use a well know library to handle all attributes. the fact that they are not able to think of an attack, does not mean that no attack can be carried out. it is also more likely that a well used library will be faster to learn about and mitigate future attacks that are not currently know.

Z:

yes the most well know are script, style, alt, onerror, but if attributes are not sanitized properly, any user input may be dangerous

Ord: 316

The code from *d)* as text:

```
// Translate internal representation to actual HTML code.
// (Slightly paraphrased to make the syntax more familiar.)
```

```
function getAttributesAsString(attrs) {
    return attrs.map(attr => attr.name + '="' + attr.value + '"').join(' ');
}
function getElementAsString(elt) {
    var attrs = getAttributesAsString(elt.attrs);
    var body = getNodeContentAsString(elt.children);
    return '<' + elt.tagName + ' ' + attrs + '>'
        + body + '</' + elt.tagName + '>';
}
// …
mainContent.innerHTML = getElementAsString(questionElt);
```

Maks poeng: 17

Knytte håndtegninger til denne oppgaven?
Bruk følgende kode:

5 7 3 1 5 4 8

5 5. Secure Gifting [10%]

You have received an email from Ms. Rudolph, co-founder and main developer at a new Internet startup that aims to "disrupt the holiday gift-giving market through the use of innovative Presents-as-a-Service (PaaS) technology". She wants your help to improve the security of their software.

Their software consists of a simple web app, where users can self-register any good or bad deeds (or "incidents", as Ms. Rudolph calls them) they do throughout the year. At the end the year, their behaviour will be analysed, and they'll then receive a special gift, selected based on how "good" they've been. For example, a "good" user might get a nice toy, a good book, or a fun game, while a "bad" user might see their behaviour rewarded with a lump of 100% carbon-neutral bio-coal.

***a)* [5%]**

In the current system, each user has to monitor their own behaviour and register their own (mis)deeds on the web site. Ms. Rudolph would like to implement a new feature where a user can authorize a computer system to register incidents on their behalf. For example, you might use a "smart home" assistant to monitor you at home, or let a well-respected, trusted app like HeadBook report on your social media behaviour automatically. *Concretely:*

- The user should be able to grant a third-party application permission to interact with the system on their behalf
- The user interface for granting such authorization should be clear and obvious
- The user should never have to share their password
- The user should be able to revoke the authorization at any time
- Access should be limited to whatever is needed – for example, a behaviour monitoring app might get access to the incident-reporting API, while a "high score" display app might get access only to summary data

How would you implement something like this? Explain. (Apart from the above criteria, you can make whatever assumptions you like.)

(1–3 paragraphs)

***b)* [3%]**

Ms. Rudolph is worried about hackers getting access to and leaking the "good/bad" list. Apart from the technical issues, what would be the likely consequences of such a data breach? Are there any particular steps that should be taken in preparation for or in response to such a leak? **Explain.** (You can assume that any laws that apply are similar to what you find in Norway or the EU.)

(~1 paragraph)

***c)* [2%]**

Ms. Rudolph says that another popular holiday-present service is offering a lot of money for access to their database (i.e., users' behavioural data). Such a deal could really help the start-up's financial situation.

Do you think they should accept the deal? What would be the privacy implications of selling access in this way? Would *you* help implement this kind of database access? **Explain your thoughts/opinions.**

(~1 paragraph)

Fill in your answer here

A:
the service should create api keys for each third party service. the api key should give the third party access on behalf of the user. the user should be able to select what the api key gives third parties access to change. and the key can be invalidated at any time.

this can be acchievd by using oauth where the user is redirected from the third party to our service, at which point they authorize themselves at our service. they are then presented with a checklist of permissions they want to grant. at the end they are redirected to the third party via a callback url. at which time the newly created api key is sent along as a cookie to the third party.

B:
the likely consequences of private sensitive information being leaked, is that we may face fines from Norwegian data protection laws, as well as European laws such as GDPR. hackers may also encrypt our data in a ransomware attack. ways of mitigating this is by encrypting all data we store in our database. this will hopefully ensure that the data will not be leaked. we should also keep backups of our data to prevent a ransomware attack.

C:
This depends on what permissions we have gotten from the user. if we ask the user in a clear way, such that we have informed consent, it might be ethical to share the users data with third parties. there may be Norwegian or European laws that prevent sharing data even in this case, so we would have to consult a lawyer. but given we have informed consent from the user, and the user can withdraw that consent in the future, i would not have any issue implementing such a feature.

the privacy implication of selling this data is that we might sell a lot of sensitive user data, that the user might not know we collected. such as religious belief, sexual orientation etc.. there are strict rules about collecting and storing sensitive data. and it is hard to verify the security of the third parties that we share our data with. they might get hacked, and this might lead to information disclosure.

Ord: 365

Maks poeng: 10

Knytte håndtegninger til denne oppgaven?
Bruk følgende kode:

1812857

6 6. HeadBook [5%]

Consider the *HeadBook* project from Assignment 2 and 3.

***a)* [3%]**

What are the **3 most important things you learned** while working on the project, or during the review process?
(1–2 paragraphs)

***b)* [2%]**

What do you feel would be the **2 most important things** to learn more about to become an effective developer of secure web applications?
(1–2 paragraphs)

Fill in your answer here

A:
That we should distrust all user input, we need to sanitize any user input to prevent for example sql injections.
we have no control over anything happening in the users browser. as such any security checks we do with JavaScript on the frontend is not enough, and will have to be rechecked on the backend.

I also learned that we need to salt and hash passwords, instead of saving them in clear text. but that we perferably should us Oauth to handle user authentication, instead of rolling our own.

lastly i learned about both static analysis in which you read the code to look for security bugs. as well as dynamic analysis where you use tools to automate the checking of possible security bugs.

B:
I think the most important lesson for secure web applications is to not roll your own security, and instead use libraries and tools well know to the community. we should for example use Oauth instead of storing passwords ourselves. but if we are storing passwords we should use established encryption libraries, instead of trying to write our own encryption methods.

secondly i learned that we should use tools such as OWASP zap to do static analysis, as well as code review by our peers to do static analysis.

Ord: 214

Maks poeng: 5

Knytte håndtegninger til denne oppgaven?
Bruk følgende kode:

2391199

7 7. Oblig [40%]

The results from your compulsory exercises will be inserted here.

Maks poeng: 40

Knytte håndtegninger til denne oppgaven?
Bruk følgende kode:

6247164

8 8. Multiple Choice Demo [0%]

*This is **not a question**, it's just a demonstration of one of the problems in the “Digital Exam” question. You'll get **no points** for answering it!*

One of the alternatives below contains JavaScript code which will insert the text *“Exam started N minutes ago!”*. (Unless, of course, the actual exam looks different from the preview, which would be another issue.)

Select one alternative:

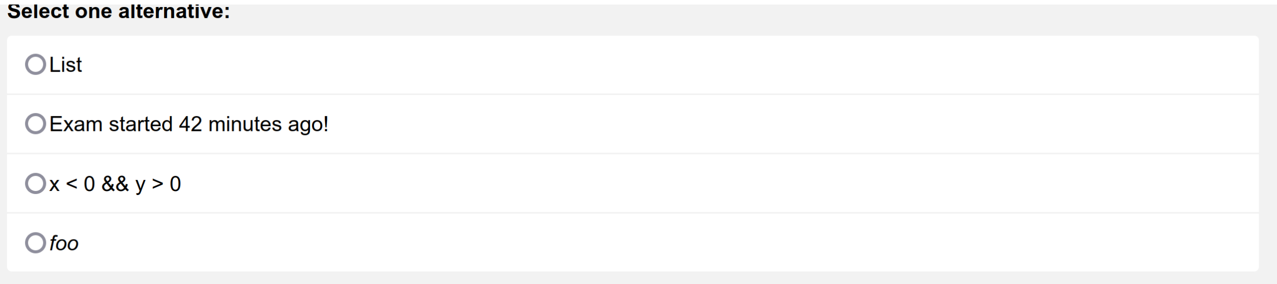
○ List

○ Exam started 222964 minutes ago!

○ x < 0 && y > 0

○ *foo*

Here's what it *should* look like (at 2023-11-30T15:42CET):



Select one alternative:

○ List

○ Exam started 42 minutes ago!

○ x < 0 && y > 0

○ *foo*

Maks poeng: 0

Knytte håndtegninger til denne oppgaven?
Bruk følgende kode:

9106152