CERTLABZ CERTIFICATION BOARD

Certified Smart Contract Security Auditor (CSCSA)

Ten Sample Questions

These ten questions are written at the difficulty of the live exam. Six carry a figure, a data table, a configuration or a code listing; four are plain scenario questions, as on the live paper. Every one is a scenario a working smart contract security auditor would meet, and none of them appears in the live question bank.

The live exam is multistage adaptive: how you score on one stage selects the difficulty of the next, so no two candidates sit the same paper. It is a performance-based exam: each form can carry up to 5 to 12 performance-based questions alongside the multiple-choice questions, which means you build, sequence or complete something rather than only choosing from a list. Answers and full explanations follow each question here; the live exam does not disclose item-level answers.

Question 1

Domain 2: Common Vulnerability Classes

CONFIG

Emma Wilson finds this pattern during an audit.

```
sendReward(msg.sender);      // external call first
claimed[msg.sender] = true;  // state update after
```

Listing 1.1 The claim logic.

What finding should the auditor raise?

A. No finding is needed, because performing the external reward call before the state update is a safe and standard ordering.

B. Reentrancy risk; the external call precedes the state update, so recommend checks-effects-interactions or a reentrancy guard.

C. A gas finding only, because the sole issue is that the reward call consumes more gas than the state write that follows it.

D. A style finding only, because the two lines are simply in an unusual order and there is no security impact from that at all.

Correct answer: B
The external call runs before the claimed flag is set, so the recipient can re-enter and claim repeatedly. The auditor should raise a reentrancy finding and recommend checks-effects-interactions or a reentrancy guard. It is not safe, not merely a gas or style issue.

Question 2 — Domain 1: Audit Foundations — DATA

James Carter classifies findings by impact and likelihood.

Finding	Severity
Anyone can drain all funds easily	(to choose)
Funds at risk only in a rare edge case	(to choose)
Gas could be reduced	(to choose)
Owner can pause unexpectedly	(to choose)

Table 2.1 Findings to rate.

Which set of severities is the most defensible?

A. Low, critical, high and informational, so draining all funds easily is rated low and a gas tweak is rated critical instead.

B. Critical, medium, informational and medium, matching impact and likelihood to how much value is realistically at risk.

C. Informational, low, critical and high, so a gas suggestion is rated critical and easy fund drainage is only informational.

D. Medium, medium, medium and medium, rating every finding the same regardless of how much value each one puts at risk.

Correct answer: B
Easy drainage of all funds is critical, a rare edge case that risks funds is medium, a gas reduction is informational, and an unexpected owner pause is a medium centralization concern. Only option B matches severity to realistic impact and likelihood; the others invert or flatten the ratings.

Question 3 — Domain 3: EVM and Solidity Internals — CONFIG

Olivia Davis reviews a proxy whose storage layout differs from its implementation.

```
proxy slot 0     : address owner
impl  slot 0     : uint totalSupply
// same slot, different meaning
```

Listing 3.1 Storage layouts.

What is the risk the auditor should flag?

A. No risk, because a proxy and its implementation are free to use identical storage slots for completely different variables.

B A storage collision; the proxy and implementation share slot 0 for different data, which can corrupt state on a call.

C A gas risk only, because sharing a slot merely makes reads and writes to slot 0 slightly more expensive than usual.

D A naming risk only, because the sole problem is that owner and totalSupply should be given the same variable name.

Correct answer: B
When a proxy and its implementation map different variables to the same storage slot, a write through one meaning corrupts the other, a storage collision. The auditor should flag it and recommend aligned storage layouts. It is not harmless, not a gas issue, and not a naming issue.

Question 4 — Domain 4: DeFi and Protocol Risks — DATA

Liam Anderson matches each DeFi risk to its mitigation.

Risk	Mitigation
Spot price used as an oracle	(to choose)
Flash-loan price manipulation	(to choose)
Unbounded external dependency	(to choose)
Single admin key	(to choose)

Table 4.1 Risks and mitigations.

Which set of mitigations fits?

A A single admin key, a spot price feed, no checks and time-weighted prices, applied so a single key is used to fix oracle risk.

B Time-weighted price feed, resistance to atomic manipulation, validated interfaces, and multisig or timelock governance.

C Ignore the oracle, encourage flash loans, trust all interfaces, and keep one admin key, since none of these are real risks.

D A bigger contract, a faster block time, more storage and a shorter name, none of which address the risks in the table at all.

Correct answer: B
A manipulable spot-price oracle is mitigated with a time-weighted price feed, flash-loan manipulation is mitigated by designs resistant to atomic manipulation, an unbounded external dependency is mitigated with validated interfaces and limits, and a single admin key is mitigated with multisig or a timelock. Only option B fits.

Question 5 — Domain 5: Tools and Techniques — CONFIG

Ava Thompson lists the techniques used across an audit.

```
static analysis  : automated pattern scan
fuzzing          : random and edge inputs
symbolic exec    : explores many paths
manual review    : human reasoning
```

Listing 5.1 Techniques.

What is the soundest way to use these?

A. Rely on static analysis alone, because an automated scan finds every class of vulnerability without any human review at all.

B. Combine them, using automated tools to surface candidates and manual review to reason about logic and confirm real issues.

C. Use fuzzing only, because random inputs will always reach every reachable state and make the other techniques unnecessary.

D. Skip manual review, since human reasoning adds nothing that the automated tools have not already found on their own.

Correct answer: B
Automated tools such as static analysis, fuzzing and symbolic execution surface candidate issues, but manual review is needed to reason about business logic and confirm real vulnerabilities, so a sound audit combines them. No single automated technique finds everything, and manual review is essential, not redundant.

Question 6

Domain 6: Reporting and Remediation

DATA

A finding is written up with the fields below.

Field	Included
Clear description of the issue	Yes
Severity and rationale	Yes
Concrete remediation	Yes
Verification after the fix	(to decide)

Table 6.1 The finding.

What completes a professional finding and its lifecycle?

A. Leave verification out, because once a fix is suggested the audit is complete and no re-check of the change is ever needed.

B. Verify the fix after remediation, confirming the change resolves the issue without introducing a new one, then close it.

C. Replace the remediation with a link to a forum, since pointing the team elsewhere is enough to close out any finding.

D. Raise the severity to critical for every finding at closing, so the report looks thorough regardless of the real impact.

Correct answer: B
A finding is not closed until the fix is verified: the auditor re-checks that the remediation resolves the issue without introducing a new problem, then closes it. Skipping verification leaves the fix unconfirmed, outsourcing remediation is not guidance, and inflating severity misrepresents risk.

Question 7 Domain 1: Audit Foundations

A client says a clean audit report guarantees their contract is completely free of bugs. How should the auditor respond?

A. Agree fully, because a clean audit is an absolute guarantee that no vulnerability can ever exist in the contract afterward.

B. Explain that an audit reduces risk and reflects the reviewed scope and time, but cannot guarantee the absence of all bugs.

C. Say the report only covers the contract name, so the client should treat the rest of the code as entirely unreviewed anyway.

D. Say audits guarantee correctness only if the client also raises the gas limit on the deployed contract after the review.

Correct answer: B
An audit reduces risk within the reviewed scope and time budget but cannot prove the absence of all bugs, so it is not a guarantee. The client should understand that limitation. The report covers more than a name, and gas limits have nothing to do with audit assurance.

Question 8 Domain 4: DeFi and Protocol Risks

A lending protocol prices collateral using the instantaneous spot price from a single on-chain pool. Why is this dangerous?

A. It is not dangerous, because an instantaneous spot price from one pool is the most accurate and manipulation-proof source.

B. An attacker can move that pool price within a transaction, for example with a flash loan, to borrow against inflated collateral.

C. It is dangerous only because reading a pool price costs too much gas, which is the sole real problem with this design choice.

D. It is dangerous only because the pool address might change, which is unrelated to how the collateral value is computed here.

Correct answer: B
A single-pool spot price can be moved within one transaction, including via a flash loan, letting an attacker borrow against artificially inflated collateral and drain the protocol. The mitigation is a manipulation-resistant oracle such as a time-weighted feed. Gas cost and pool address changes are not the core danger.

Question 9 Domain 3: EVM and Solidity Internals

During review you see a contract use delegatecall to run code from an address supplied by the caller. Why is this a serious concern?

A. It is not a concern, because delegatecall to caller-supplied code is a safe and recommended way to add flexibility.

B. delegatecall runs external code in this contract storage and context, so caller-controlled code can hijack state and funds.

C. The only concern is readability, because delegatecall makes the code slightly harder for other developers to follow.

D. The only concern is gas, because delegatecall to another address always costs far more than a normal internal call.

Correct answer: B
delegatecall executes the target code in the calling contract storage and context, so allowing a caller to supply that target lets attacker-controlled code manipulate this contract state and funds. That is a critical finding. It is not safe, and the issue is far more than readability or gas.

Question 10 Domain 6: Reporting and Remediation

An auditor discovers a critical, actively exploitable bug in a live protocol during an engagement. What is the responsible course?

A. Publish full exploit details publicly at once, so that everyone can see the bug before the team has any chance to respond.

B. Follow responsible disclosure: privately report it to the team with enough detail to fix it, and coordinate timing of any release.

C. Exploit it quietly to prove the point, because demonstrating the drain on the live protocol is the clearest way to report it.

D. Ignore it until the engagement ends, since a live bug is out of scope once the formal review window has already started.

Correct answer: B
Responsible disclosure means privately reporting a critical live bug to the team with enough detail to remediate, and coordinating the timing of any public release. Publishing exploit details immediately aids attackers, exploiting it is unethical and likely illegal, and ignoring a live critical bug is negligent.

Published by the CertLabz Certification Board, a division of CertLabz LLC. Companion to objectives document CSCSA-001. https://certlabz.com/certifications/certified-smart-contract-security-auditor.html