CERTLABZ CERTIFICATION BOARD

Certified Polygon Developer (CPGD)

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 Polygon developer 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 1: Polygon and Ethereum Scaling Foundations

DATA

Emma Wilson compares scaling approaches by where each keeps its data, with one row missing.

Approach	Where data lives
Rollup	On chain, on L1
Sidechain	Own consensus
Validium	(to choose)
State channel	Between two parties

Table 1.1 Scaling approaches and data.

Where does a validium keep its data?

A. On chain as calldata
B. Off chain by a DAC
C. Inside the mempool
D. On each user wallet

Correct answer: B

A validium keeps transaction data off chain, held by a data availability committee, while proofs post to Ethereum. This lowers cost but adds a trust assumption on that committee. Rollups instead publish data on chain, and neither the mempool nor wallets store validium state.

Question 2 — Domain 2: Polygon PoS Architecture — DATA

James Carter maps each Polygon PoS component to its role, with one role missing.

Component	Role
Heimdall	(to choose)
Bor	Block production
Validators	Stake and secure
Checkpoints	Commit to Ethereum

Table 2.1 PoS components and roles.

What is the role of Heimdall?

A. Executes the EVM
B. Holds user keys
C. Consensus layer
D. Sets gas prices

Correct answer: C
On Polygon PoS the Heimdall layer runs validator consensus and submits checkpoints to Ethereum, while Bor produces blocks and executes transactions. Validators stake to secure the chain, but they do not hold user keys or set gas prices directly.

Question 3 — Domain 3: Smart Contract Development — CONFIG

Olivia Davis reviews a withdraw function before deployment.

```
function withdraw() external {
  uint256 bal = balances[msg.sender];
  (bool ok, ) = msg.sender.call{value: bal}("");
  require(ok);
  balances[msg.sender] = 0;
}
```

Listing 3.1 A withdraw function.

What flaw does this function show?

A. Reentrancy risk
B. Integer overflow
C. Missing import
D. Wrong pragma line

Correct answer: A
The function sends value with a low level call before it zeroes the balance, so a malicious receiver can call back in and drain funds. Updating state before the external call, or using a reentrancy guard, prevents this. The snippet shows no overflow, import, or pragma fault.

Question 4 — Domain 4: zkEVM and Rollups — DATA

Liam Anderson compares two rollup styles, with the zk finality wait left blank.

Property	Optimistic	zkEVM
Proof type	Fraud proof	Validity proof
Finality wait	About 7 days	(to choose)

Table 4.1 Rollup finality by style.

How long is the zkEVM finality wait?

A. About seven days
B. Several weeks long
C. Never finalizes
D. Minutes not days

Correct answer: D
A zkEVM posts validity proofs, so once a proof is verified the state is final and withdrawals settle in minutes to hours. Optimistic rollups wait out a challenge window of about seven days. The other choices misstate how validity proofs shorten finality.

Question 5 — Domain 5: Bridges, Tokens and Interoperability — DATA

Ava Thompson maps each token standard to its use, with the ERC-721 use missing.

Standard	Use
ERC-20	Fungible tokens
ERC-721	(to choose)
ERC-1155	Multi token batch

Table 5.1 Token standards and use.

What is the ERC-721 standard for?

A. Unique NFTs
B. Fungible cash tokens
C. Gas fee refunds
D. Staking rewards

Correct answer: A
ERC-721 defines non fungible tokens where each token is unique and non interchangeable. ERC-20 covers fungible tokens and ERC-1155 handles batched multi token transfers. Gas refunds and staking rewards are not what the ERC-721 standard describes.

Question 6

Domain 6: Security, Testing and Deployment

CONFIG

David Brown reviews an owner setter and asks what safeguard is missing.

```
function setOwner(address n) external {
  owner = n;
}
```

Listing 6.1 An owner setter.

What is missing from this setter?

A. A reentrancy guard
B. A payable modifier
C. A gas refund line
D. An access check

Correct answer: D
The setter lets any caller change the owner because it has no access control, so an attacker can seize the contract. Adding a check such as onlyOwner restricts the call to the current owner. A payable tag, reentrancy guard, or refund line would not close this gap.

Question 7

Domain 3: Smart Contract Development

A team must fix a bug in a deployed contract without changing the address that users and other contracts already trust. Which pattern lets them do this?

A. Redeploy fresh
B. Edit the bytecode
C. Proxy pattern
D. Fork the chain

Correct answer: C
A proxy pattern keeps a stable address that delegates to a swappable logic contract, so teams can ship a fix without moving user funds or changing the address. Redeploying fresh changes the address, editing bytecode is not possible, and forking the chain is unrelated.

Question 8

Domain 4: zkEVM and Rollups

A rollup posts compressed transaction data to layer one every batch, which raises cost. An engineer asks why this posting is worth keeping. What does it provide?

A. Removes all gas cost
B. Lets anyone verify
C. Hides the payload
D. Speeds up mining

Correct answer: B
Posting compressed transaction data to layer one gives data availability, so anyone can rebuild the rollup state and verify it independently. That guarantee is what keeps the rollup trust minimized. It does not remove gas cost, hide the payload, or speed up mining.

Question 9

Domain 5: Bridges, Tokens and Interoperability

A cross-chain bridge locks large balances and releases them on a small multisig of three signers. A reviewer asks what the largest risk is. Which is it?

A. Key compromise
B. Slightly high fees
C. Slow block times
D. Small supply cap

Correct answer: A
A bridge that locks large balances behind a small multisig fails badly if those signing keys are compromised, letting an attacker unlock the pooled funds. Hardening key custody and widening the signer set reduce this risk. Fees, block times, and supply caps are minor by comparison.

Question 10

Domain 6: Security, Testing and Deployment

A team ships a new contract straight to mainnet without running it on a testnet or a forked environment first. Which practice did they skip?

A. Skip all reviews
B. More marketing
C. Testnet trials
D. Bigger gas cap

Correct answer: C
Deploying straight to mainnet skips testnet and forked environment trials, where bugs surface cheaply before real value is at stake. Running tests on a testnet first is the missing practice. Marketing, review skipping, and larger gas caps do not address untested code.

Published by the CertLabz Certification Board, a division of CertLabz LLC. Companion to objectives document CPGD-001. https://certlabz.com/certifications/certified-polygon-developer.html