

Ethereum Developer Workshop and CBDE Preparation

Solidity, contracts, testing and Ethereum development foundations

FA-0715 · 5 days

Software Development · Cybersecurity

Introduction

Build Ethereum development foundations through small Solidity contracts, local-network testing and selected security reviews. The five-day workshop retains the source's CBDE preparation context while distinguishing current development practice from historical exam tooling.

This is a training course, not an award of certification or assurance of production-ready security. Exam registration, eligibility, fees and policies are managed separately by the provider and should be checked directly before booking.

Learning outcomes

- Explain blockchain accounts, transactions, Ethereum execution and proof-of-stake consensus.
- Write and compile small Solidity contracts using relevant data types, functions, events and inheritance.
- Interpret gas use, ABI-based interaction and selected EVM concepts.
- Test and deploy sample contracts on a local network or approved test network.
- Recognise ERC-20 and ERC-721 interfaces and use suitable maintained libraries.
- Review common contract risks, external interactions, randomness and upgrade trade-offs.
- Relate the practice topics to the provider's published CBDE scope without assuming current tools match its legacy references.

Prerequisites

- Basic programming, functions and object-oriented concepts; JavaScript familiarity is helpful for client interaction.
- A compatible Solidity and maintained development-tool setup with access to sample projects.
- Approved local or test-network accounts; no real assets, mainnet deployment or production private keys are required.

Training outline

1. Day 1 — Blockchain and Ethereum foundations

- Blockchain models, blocks, wallets, addresses, public/private keys and architectural trade-offs.
- Ethereum, Ether, smart contracts and decentralised applications; basic hashing and account concepts.
- Ethereum execution and proof-of-stake validators; mining is historical context, not current Ethereum operation.
- Local, test and public networks; node roles and a guided development-node demonstration.
- Short concept checks and an introduction to the published CBDE topic scope.

2. Day 2 — Solidity and the EVM

- Solidity file layout, types, arrays, structs, mappings and storage versus memory.
- Functions, state variables, getters, modifiers, receive/fallback behaviour, events and loops.
- Compile a small contract, inspect its interface and discuss transaction cost and gas limits.
- Introduce EVM opcodes and state structures, including the Merkle Patricia trie, at a conceptual level.

3. Day 3 — Tokens and contract interaction

- ERC conventions, ERC-20 and ERC-721 interfaces; review a small token example using compatible library components.
- Inheritance, ABI and metadata; invoke a contract using a maintained client such as ethers.js or viem.
- Distinguish historical Web3.js exam references from a maintained current lab stack.
- Handle reverts and external-call failures; introduce low-level calls and assembly with caution rather than making them the default.

4. Day 4 — Development workflow and testing

- Use a maintained framework such as Hardhat for compilation, a local network, tests and deployment.

- Read legacy Truffle examples as migration context; the archived suite is not the required current environment.
- Write tests for expected behaviour, boundaries and failure paths; inspect gas use under stated assumptions.
- Deploy only sample contracts to the local or approved test network and document configuration and results.

5. Day 5 — Security review and preparation consolidation

- Review reentrancy, access control, unexpected external behaviour and the limits of on-chain randomness.
- Use appropriate OpenZeppelin components without assuming that importing a library makes the entire application secure.
- Discuss ecosystem upgrades, compiler/tool versions, upgradeability and bug-fix planning.
- Review the small project against a threat-oriented checklist and identify issues requiring deeper independent review.
- Consolidate learning with practice questions and current provider guidance; passing an exam is neither promised nor included.

Certification context

The original source was organised around the Blockchain Training Alliance CBDE exam. Provider requirements and tooling references may differ from current development practice; consult the official CBDE page for current registration and assessment information. Training completion does not confer the credential.