Course Title: Introduction to Blockchain with Celo and Solidity Development

---

Course Overview:

This intensive one-month course introduces blockchain technology with a focus on the Celo platform and Solidity development. The course covers blockchain fundamentals, smart contract development, and building decentralized applications (dApps) on Celo. It is designed for developers and tech enthusiasts with basic programming knowledge who want to quickly dive into blockchain and dApp development.

Course Objectives:

By the end of the course, participants will:

- Understand blockchain technology and the Celo platform.
- Write and deploy smart contracts using Solidity.
- Build decentralized applications on the Celo blockchain.
- Utilize Celo development tools like Remix, Truffle, Hardhat, and integrate smart contracts with Web3.js or Ethers.js.

Prerequisites:

- Basic programming experience (JavaScript or Python preferred).
- Familiarity with web development (optional).

Course Duration:

4 weeks
10-12 hours per week (combination of lectures, hands-on labs, and project work).

Course Outline:

Week 1: Blockchain Fundamentals and Celo Introduction

Day 1-2: Introduction to Blockchain

- **Topics:**
    - Overview of blockchain technology: decentralized networks, consensus mechanisms, immutability.
    - Types of blockchains (public, private, permissioned, permissionless) and their use cases.
- **Hands-on:**
    - Introduction to blockchain tools (Remix IDE, Celo CLI).
    - Basic wallet setup and token transfers on Celo Testnet (Alfajores).

Day 3-4: Celo Blockchain Overview

- **Topics:**
    - Overview of the Celo platform: mobile-first design, stablecoins (cUSD, cEUR), and CELO.
    - Celo's Proof-of-Stake consensus mechanism and its architecture.
- **Hands-on:**
    - Create a Celo wallet and interact with the Alfajores Testnet using the Celo Faucet.

Day 5-6: Introduction to Smart Contracts

- **Topics:**
    - Introduction to smart contracts and Solidity programming language.
    - Basic Solidity syntax: data types, functions, conditionals, loops.
- **Hands-on:**
    - Write and deploy a simple "Hello World" smart contract using Remix IDE on Celo.

Assignment:

- Deploy a basic smart contract on Celo's Testnet and share the contract address.

Week 2: Solidity Development and Celo Tools

Day 1-2: Advanced Solidity Concepts

- **Topics:**
    - Solidity data structures: arrays, mappings, and structs.
    - Smart contract events, modifiers, and inheritance.
- **Hands-on:**

    - Develop a token contract (ERC-20-like) on Celo using Solidity.

Day 3-4: Celo Development Tools

- **Topics:**
    - Introduction to Truffle, Hardhat, and Celo SDK.
    - Deploying contracts on Celo using Truffle/Hardhat.
- **Hands-on:**
    - Set up Truffle or Hardhat for Celo development.
    - Deploy a smart contract on the Celo Alfajores Testnet using Hardhat or Truffle.

Day 5-6: Smart Contract Security

- **Topics:**
    - Common vulnerabilities in smart contracts: reentrancy, overflow/underflow.
    - Best practices for securing smart contracts.
- **Hands-on:**
    - Implement security measures in an existing Solidity contract.

Assignment:

- Refactor and secure an ERC-20 token contract using security best practices.

Week 3: Building Full-stack Decentralized Applications (dApps)

Day 1-2: Frontend Integration with Web3.js and Ethers.js

- **Topics:**
    - Introduction to Web3.js and Ethers.js libraries for interacting with smart contracts.
    - Connecting smart contracts to frontend applications.
- **Hands-on:**
    - Build a simple web interface to interact with your deployed smart contract.

Day 3-4: Building dApps on Celo

- **Topics:**
    - Architecture of dApps: smart contracts, frontend, backend.
    - Introduction to Celo APIs and mobile-first dApp development.
- **Hands-on:**
    - Develop a basic dApp (e.g., a voting system or crowdfunding platform) using Celo's smart contract infrastructure and Web3.js.

Day 5-6: Testing and Debugging dApps

- **Topics:**
  - Strategies for testing and debugging smart contracts and dApps.
  - Use of testing frameworks like Mocha and Chai for smart contract testing.
- **Hands-on:**
  - Write test cases for your smart contract using Truffle or Hardhat.

Assignment:

- Develop and test a simple full-stack dApp, ensuring it can interact with your deployed contract.

Week 4: Project Development and Deployment

Day 1-2: Scaling and Deploying dApps

- **Topics:**
  - Deploying dApps to the Celo Mainnet.
  - Strategies for scaling dApps: optimizing gas fees, contract upgrades, performance considerations.
- **Hands-on:**
  - Finalize your dApp and prepare for Mainnet deployment.

Day 3-4: Final Project Development

- **Topics:**
  - Integrating Celo wallets (Valora) into your dApp.
  - Ensuring compatibility and user experience on mobile devices.
- **Hands-on:**
  - Continue working on the final dApp project with mentoring support.

Day 5-6: Project Presentation and Mainnet Deployment

- **Topics:**
  - Review deployment strategies and best practices for going live on the Mainnet.
  - Demonstrating and presenting your final project.
- **Hands-on:**
  - Deploy your final project on the Celo Mainnet and demonstrate its functionality.

Final Project Submission:

- Complete and deploy a full-stack decentralized application on Celo. Submit the Mainnet contract address, project files, and a project report.

Tools and Resources:

- **Celo Documentation**: docs.celo.org
- **Solidity Documentation**: soliditylang.org
- **Remix IDE**: remix.ethereum.org
- **Truffle Suite**: trufflesuite.com
- **Hardhat**: hardhat.org
- **Web3.js Documentation**: web3js.readthedocs.io

Assessment:

- **Assignments** (40%): Weekly coding exercises and smart contract deployments.
- **Final Project** (60%): A complete decentralized application built on Celo and deployed on the Mainnet.
