AI CERTs™

Bitcoin+ Developer™ (5 Days)

Program Detailed Curriculum

Executive Summary

Bitcoin+ Developer™ certification program offers comprehensive training, covering fundamentals of Bitcoin and the underlying Blockchain technology, fundamentals of bitcoin scripting, building applications using bitcoin scripts, Layer 2 scaling solutions, real-world use cases, security best practices, integration with SDK/APIs, and insights into future trends. The program culminates with a hands-on project work for practical application and a capstone project.

Course Prerequisites

- Familiarity with general programming concepts like data structures, algorithms and networks
- Understanding of at least one legacy programming stack (e.g. Python, JavaScript, Java or similar)
- Fundamental knowledge to use command line consoles on any operating system
- Ability to understand developer concepts like SDKs, APIs, application development tools etc.
- Experience with building end to end tiered applications

Module 1

Introduction to Bitcoin and Blockchain

1.1 Overview of Bitcoin Technology

- **Introduction to Bitcoin:** Explore Bitcoin's fundamentals, including its key features and decentralized architecture to gain insights into the revolutionary technology.
- **Digital Currency:** Explore how digital currency reshapes money's concept, leveraging the internet for swift, borderless transactions, using cryptography.
- **Popular Cryptocurrency:** Explore the diverse landscape of popular cryptocurrencies, from Ethereum's smart contracts to Ripple's cross-border payments.
- **P2p Cash System:** Discover direct currency transactions without banks, leveraging blockchain for security, decentralization, and inclusivity.
- **Limited Supply:** Discover the scarcity-driven value of cryptocurrencies like Bitcoin, offering stability and long-term investment potential.
- **Bitcoin Ledger as a Store of Value:** Explore the role of Bitcoin Ledger in wealth preservation and its potential as a reliable store of value.
- **Bitcoin as a Platform for Permissionless Innovation:** Explore Bitcoin's capacity for unrestricted innovation, empowering individuals to drive change without centralized permission.
- **History of Bitcoin:** Explore the origins, evolution, and impact of Bitcoin, delving into its history, technology, and implications for the future.
- **Key Features and Their Utility:** Discover Bitcoin's pioneering features, including decentralization, security, transparency, and peer-to-peer transactions, revolutionizing finance.

- **Understanding the Decentralized Nature of the Bitcoin Network:** Learn about Bitcoin's decentralized structure, its implications, and functionalities.

1.2 Overview of Blockchain Technology

- **Fundamentals of Blockchain Technology:** Explore fundamentals of blockchain and dissect its structure to understand the world of distributed ledger systems.
- **How Blocks and Transactions are Structured in the Bitcoin Blockchain:** Learn the fundamental structure of Bitcoin blockchain, exploring blocks and transactions.

Module 2

Bitcoin Scripting Language

2.1 Fundamentals of Bitcoin Scripting Language

- **Stack-based Language:** Explore the role of Bitcoin script as a stack based programming environment. Evaluate its limited and and deterministic nature to prioritize security.
- **Limited and Deterministic:** Understand Bitcoin script's deterministic limits for security, avoiding loops and recursion to ensure predictable execution.
- **ScriptPubKey and ScriptSig:** Explore Bitcoin transactions with ScriptPubKey (output conditions) and ScriptSig (input data), defining spending conditions for secure fund transfers.

2.2 Components of Scripting Language Basics

- **Operators and Operations:** Learn Bitcoin simple operators manipulate stack data, including arithmetic, cryptographic, and conditional operations.
- **Standard Script Templates:** Master various standard script templates for efficient and secure cryptocurrency transactions, including P2PKH, P2SH, and multisig.
- **Public Key and Hash Operations:** Explore Bitcoin script's cryptographic functions like public keys, hashes, and their comparisons for secure transactions and validation.
- **Conditional Statements:** Delve into conditional statements, navigating program paths based on conditions, including essential if-else constructions.

2.3 Uses of the Bitcoin Scripts

- **Time-Locked Transactions:** Discover time-locked transactions to safeguard funds until set block height or time, essential for smart contract functionality.
- **OP_RETURN:** Understand how to empower transactions with non-spendable data, expanding blockchain utility beyond conventional spending capabilities.

2.4 Execution Condition of Scripts

- **Script Evaluation:** Learn how Bitcoin's script evaluation system validates transactions by comparing input and output scripts.
- **Segregated Witness (SegWit):** Explore revolutionizing cryptocurrency transactions by separating witness data and enabling intricate scripting possibilities.

2.5 Security and Vulnerabilities in the Scripts

- **Security Considerations:** Delve into script security for Bitcoin, addressing vulnerabilities and fortifying against potential threats through strategic considerations and restrictions.

Module 3

Building on Bitcoin Script

3.1 Standard Transactions

- **Pay-to-Pubkey (P2PK) and Pay-to-Pubkey-Hash (P2PKH):** Learn about Pay-to-Pubkey (P2PK) and Pay-to-Pubkey-Hash (P2PKH) for Bitcoin transactions and everyday use.
- **Multisignature (Multisig):** Dive into enhanced security with Multisig, requiring multiple keys for collaborative finance management and corporate accounts.
- **Time-Locked Transactions:** Navigate through Bitcoin script's time-locked transactions, enabling delayed spending until a set date or block height. Various practical applications.
- **Hashed Time-Locked Contracts (HTLCs):** Explore secure off-chain transactions via Lightning Network using time-locked contracts with hash preimage conditions.

3.2 Trade and DeFi

- **Atomic Swaps:** Discover how atomic swaps revolutionize cross-chain trading, enabling trustless cryptocurrency exchange without centralized intermediaries.
- **Pay-to-Script-Hash (P2SH):** Understand how to simplify complex script creation, enhance blockchain efficiency by encoding spending conditions in hashed scripts.
- **Simple Token Systems:** Learn to create token systems atop Bitcoin blockchain, enabling representation and transfer for diverse applications.

3.3 Smart Contracts

- **Oracle Contracts:** Explore integrating Bitcoin script with external data sources for contracts reliant on real-world information in various applications.
- **Smart Contracts:** Master Bitcoin script for crafting smart contracts like escrow, multi-step transactions, and custom fund release conditions.

3.4 Asset Definition

- **Colored Coins:** Explore Bitcoin Scripting Language applications, focusing on asset definition through Colored Coins for representing and trading assets on the Bitcoin blockchain.

Module 4

Layer 2 Scaling Solutions

4.1 Basic of Layer 2

- **Bitcoin Layer 2:** Explore Layer 2 optimizations for Bitcoin, enhancing scalability, reducing latency, and improving efficiency beyond the base layer for transactions.

4.2 Different Layer 2 Projects

- **Lightning Network (LN):** Learn decentralized off-chain scaling solutions for fast, low-cost Bitcoin transactions via payment channels.
- **Liquid Network:** Unlock the potential of a Blockstream federated sidechain for faster, confidential Bitcoin transactions. Ideal for institutions.
- **Elements Sidechain:** Learn to utilize sidechains, enabling asset transfer between Bitcoin's blockchain and separate blockchains for interoperability.

- **Rootstock (RSK):** Explore the platform facilitating smart contract creation and DApp development on the Bitcoin blockchain.
- **Statechains:** Understand how to facilitate secure off-chain bitcoin transfers, reducing costs, and enhancing scalability efficiently.
- **Drivechain:** Uncover the potential of bidirectional transfer of bitcoins between main blockchain and sidechains.
- **Plasma:** Dive into a scalable blockchain framework using sidechains, adaptable to Bitcoin for enhanced scalability.
- **Counterparty:** Learn to create custom tokens on Bitcoin blockchain using Counterparty protocol for enhanced transaction functionality.

Module 5

Use Cases and Projects using Layer 2

5.1 Payments Use Cases

- **Microtransactions and Micropayments:** Discover methods to streamline microtransactions, benefiting content creators, pay-per-use services, and online tipping platforms.
- **Retail Payments:** Learn to expedite the payments and reduce costs of in-store/online purchases with Bitcoin via Layer 2 solutions.
- **Remittances:** Understand how to enhance cross-border transfers with Layer 2 solutions, minimizing costs and time for efficient remittance processes.

5.2 Assets and Defi

- **Gaming and Digital Assets:** Implement in-game transactions, digital asset ownership, and item trading with reduced fees and faster settlement using Layer 2 solutions.
- **Tokenized Assets and Securities:** Learn how to issue and trade tokenized assets like real estate or stocks on Bitcoin blockchain.
- **Decentralized Finance (DeFi):** Develop decentralized financial applications, including lending, borrowing, and decentralized exchanges, on Layer 2 solutions to enhance scalability and reduce transaction costs.
- **Streaming and Content Monetization:** Implement microtransactions for streaming content, enabling users to pay for content on a per-second or per-minute basis.
- **Non-Fungible Tokens (NFTs):** Create and trade NFTs representing digital art, collectibles, or virtual assets with faster confirmation times and reduced fees.

5.3 Industry Use Cases

- **Supply Chain and Provenance:** Enhance transparency in supply chain management by recording product information and transactions on the Bitcoin blockchain through Layer 2 solutions.
- **Cross-Platform Token Swaps:** Facilitate trustless and decentralized token swaps between different blockchain platforms using Layer 2 solutions.
- **Decentralized Identity:** Develop decentralized identity solutions on the Bitcoin blockchain, allowing users to control and manage their identity without relying on centralized authorities.
- **Smart Contracts and Oracles:** Utilize Layer 2 solutions for executing more complex and scalable smart contracts, including those dependent on external data sources (oracles).
- **Off-Chain Voting Systems:** Implement secure and efficient off-chain voting systems for elections or governance processes on the Bitcoin blockchain.

5.4 Integration with Other Tech- IoT

- **Layer 2 Applications for IoT:** Explore IoT integration with Layer 2 applications, emphasizing secure micropayments and data transfer for cost-effective IoT device communication.

Module 6

Security and Best Practices

6.1 For Scripting Apps

- **Understand Bitcoin Script Fundamentals:** Master Bitcoin script fundamentals: operations, stack manipulation, transaction structure. Essential for secure application development.
- **Use Standard Script Templates:** Learn to implement widely supported and tested standard script templates like P2PKH, P2SH, and multisig.
- **Validate Input Data:** Explore how to validate input data, ensuring adherence to expected formats, including public keys and signatures.
- **Implement Multi-Signature for Enhanced Security:** Understand how to use multi-signature scripts to enhance security, especially for applications that involve multiple parties or require additional authorization for spending funds.

6.2 General Security Practices

- **Avoid Unnecessary Complexity:** Learn how to simplify scripts to meet requirements, avoiding complexity to mitigate potential security risks in applications.
- **Use Time-Lock Constraints Carefully:** Master time-lock constraints for transactions, ensuring proper implementation to prevent unexpected outcomes and pitfalls.
- **Follow Best Practices for Secure Coding:** Discover secure coding essentials: input validation, error handling, and adherence to coding standards to mitigate vulnerabilities.
- **Thoroughly Test Scripts:** Understand how to comprehensively test Bitcoin scripts in diverse scenarios, covering edge cases and security measures in testnet.
- **Regularly Update Dependencies:** Highlight the importance of regularly updating dependencies for enhanced security and performance in your software development.

6.3 Keys and Smart Contracts

- **Secure Key Management:** Implement secure key management practices, including the use of hardware wallets or secure key storage solutions. Protect private keys from unauthorized access and use.
- **Audit Code and Smart Contracts:** Learn to review Bitcoin script code, prioritize smart contract security, and engage third-party audits.
- **Monitor for Anomalies:** Develop expertise in detecting irregularities within Bitcoin script applications through robust monitoring strategies and tools.

6.4 Off the Chain

- **Educate Users on Security Best Practices:** Delve into essential security protocols, educating users on private key management, secure wallets, and threat avoidance.
- **Secure Communication:** Learn to safeguard communication channels with encryption and secure protocols for protecting data during transit.

6.5 Layer 2

- **Secure Funding Transactions:** Master securing funding transactions for Layer 2 interactions, ensuring validation of inputs, outputs, and amounts.
- **Implement Secure Multi-Signature Wallets:** Master the implementation of secure multi-signature wallets for Layer 2 applications using hardware wallets and key management.
- **Carefully Design Smart Contracts:** Refine your skills in crafting secure smart contracts for resilient Layer 2 application implementations.

- **Secure Channel Updates:** Familiarize yourself with safeguard payment channels through secure mechanisms and signature validation.
- **Monitor Channel States:** Learn to monitor and respond to channel state changes swiftly to mitigate disputes and premature closures.
- **Use Secure Communication Channels:** Understand how to secure Layer 2 communication with encryption and authentication for safeguarding sensitive data transmission.
- **Channel Management Security:** Implement secure channel management practices, including proper handling of channel closures, force closures, and penalty transactions.
- **Watchtower Services:** Explore watchover services for Layer 2 app security, preventing fraud and ensuring reliability against cheats.
- **Conduct Rigorous Testing:** Thoroughly test Layer 2 applications in various scenarios including normal operation, edge cases, and potential attacks.
- **Security Audits:** Learn to utilize third-party experts for independent audits, promptly addressing identified vulnerabilities in your application.
- **Regularly Update Dependencies:** Outline the importance of maximizing security and performance by keeping dependencies current; update libraries and SDKs promptly for enhancements.
- **Privacy Considerations:** Master privacy in Layer 2 apps, safeguard sensitive data, and implement effective protection measures for users.
- **Stay Informed About Layer 2 Ecosystem:** Acquire knowledge about latest developments, security considerations, protocol updates, vulnerabilities, and best practices.

Module 7

Integration (SDK/APIs) and Deployment

7.1 Basic Overview of SDK & APIs

- **Integration SDKs (Software Development Kits) and APIs:** Explore Bitcoin application development using Integration SDKs and APIs, gaining essential tools for seamless interaction with the blockchain.

7.2 BitcoinJS

- **Bitcoin Applications with JavaScript:** Understand the role of BitcoinJS as a JavaScript library for Bitcoin, enabling address, transaction, and cryptographic operations in web applications.
- **Scripting Support:** Explore Bitcoin scripting using BitcoinJS, enabling custom transaction logic and detailed code explanations.

7.3 BitcoinJ

- **BitcoinJ:** Discern the role of Java library for Bitcoin in facilitating wallet creation, network interaction, and Java application development, empowering developers in Bitcoin ecosystem.

7.4 Bitcoinlib

- **Bitcoinlib (python):** Explore the role of Bitcoinlib (Python) in Python library facilitating Bitcoin address, transaction, and cryptographic operations, ensuring simplicity and ease of use.

7.5 Bitcoin RPC Client (Python)

- **Bitcoin RPC Client (Python):** Explore the role of Python library in facilitating Bitcoin Core RPC interaction, empowering developers to script seamless communication with Bitcoin Core nodes.

7.6 Deployment Strategies

- **Testnet Environment:** Learn how to utilize Bitcoin testnet for script deployment and testing without real bitcoin usage. Risk-free experimentation.
- **Regtest Mode:** Explore Bitcoin Core's "regtest" mode: run a private, locally-mined blockchain for sandbox testing control.
- **Unit Testing:** Learn to test Bitcoin script components individually using relevant testing frameworks for error-free functionality.
- **Integration Testing:** Verify application component interactions, including with Bitcoin blockchain, through integration testing for issue detection.
- **Functional Testing:** Delve into functional testing to ensure Bitcoin script application meets requirements and functions correctly.
- **Scenario Testing:** Learn to craft and test scenarios replicating real-world conditions for robust Bitcoin script functionality.
- **Script Debugger:** Explore Bitcoin script debuggers for script execution visualization, aiding in understanding and troubleshooting script behavior.

Module 8

Future Trends and Innovations

8.1 Innovation in Bitcoin Applications

- **DeFi on Bitcoin:** Explore integrating decentralized finance (DeFi) features into Bitcoin ecosystem through layer 2 solutions and sidechains.
- **Privacy Improvements:** Discover advancements like CoinJoin, Confidential Transactions, and Schnorr Signatures enhance transaction privacy effectively.
- **Smart Contracts on Bitcoin:** Unlock advanced smart contract capabilities on Bitcoin through initiatives like Taproot and Schnorr Signatures for efficiency.
- **Decentralized Autonomous Organizations (DAOs) on Bitcoin:** Explore decentralized governance structures and decision-making on the Bitcoin blockchain through concepts like Decentralized Autonomous Organizations.

8.2 Innovation in Bitcoin Layer 2

- **Layer 2 Scaling Solutions::** Delve into Bitcoin Layer 2 advancements, focusing on Lightning Network and sidechains. Tackle scalability issues with off-chain solutions for improved efficiency.

8.3 Innovation in Asset Definition

- **Tokenization and Colored Coins:** Learn about tokenization and colored coins, enabling representation and trade of real-world assets on Bitcoin.
- **NFTs on Bitcoin:** Explore adding NFT functionality to Bitcoin via layer 2 solutions, tapping into the expanding NFT market.

8.4 Innovation in Bitcoin Interoperability

- **Cross-Chain Interoperability:** Discover how to enable seamless interactions and decentralized exchanges among diverse blockchain networks for trustless asset transfers.
- **Integration with Web3 and Web3.0 Standards:** Explore Bitcoin applications leverage standards, enhancing interoperability and collaboration within blockchain networks.

8.5 Innovation in Bitcoin Identity and Users

- **Decentralized Identity and Notary Services:** Learn leveraging Bitcoin for decentralized identity and notary services, ensuring secure and tamper-proof record-keeping.
- **Improved User Experience:** Delve into enhancing Bitcoin app UX with user-friendly wallets, intuitive interfaces, and educational resources.

8.6 Innovation in Bitcoin Dev Tools

- **Bitcoin Development Frameworks:** Explore advanced frameworks, empowering developers in Bitcoin application creation with streamlined processes and expanded resources for innovation and efficiency.

Module 9

Capstone Project

- **Capstone Project:** Apply course concepts in a hands-on Bitcoin application development, addressing real-world scenarios for comprehensive learning and skill integration.