Blockchain Innovation Program Tutorial Framework

BIP BLOCKCHAIN INNOVATION PROGRAM

POWERED BY

BSV ACADEMY

Blockchain Innovation Program
Tutorial Framework

The Blockchain Innovation Program is a designed as an intensive 10-week programme which will provide students with educational and entrepreneurial support for them to develop a comprehensive understanding of what's involved in Bitcoin Application Development.
The programme will see the students complete the three bitcoin primitive courses Hash Functions, Merkle Trees, and Digital Signatures before they complete the newly refactored Introduction to Bitcoin Development.

Educators from the Bitcoin SV Academy team will prescribe weekly resource and question packs to stimulate the students to develop a deeper consideration for what is involved in creating a scalable Bitcoin application. The students will attend fortnightly tutorials where their answers to the question pack will be evaluated and discussed in greater detail.

Live session #1

Hash Functions	• Hash Functions and Hash Tables • Content Addressed Distributed Data Structures • Efficient Provable Data Possession for Hybrid Clouds

Course pre-requisite: Hash Functions (primitives)
Worksheet to complete prior to the live session: Week 1 – Hash Functions

Live session #2

Merkle Trees	• GitHub MerkleDAG • ForkBase: Immutable, Tamper-evident Storage • Substrate for Branchable Applications • Merkle-CRDTs - MerkleDAGs meet CRDTs • Merkle^{2}: A Low Latency Transparency Log System
Digital Signatures	• Digital Signatures • Legitimating Technologies: Digital Signatures Case Study. • Segwit, Mixing and Law • SigHash Flags

Course pre-requisite: Merkle Trees and Digital Signatures (Primitives)
Worksheet to complete prior to the live session: Week 2 Merkle Trees and Week 3 Digital Signatures.

Live session #3

Data and Databases	• What is DBaaS? • SQL vs NoSQL • What is Cloud Storage? • What is Object Storage? • Block vs File Storage • What is a Load Balancer? • Kubernetes vs Docker
API led Event-Driven & Microservices Architectures	• API vs SDK • What is API Management? • What is a REST API? • What is an API Gateway? • What is Event Driven Architecture? • What are Microservices? • Architecting a Cloud Native API Solution. • Blockchain Enabled Trustless API Marketplace • Unofficial API and Browser Extension Development for Augmenting Student Resources

Course pre-requisite: Bitcoin Development Chapter 1
Worksheet to complete prior to the live session: Week 4 Data and Databases and API led Event-Driven & Week 5 Microservices Architectures

Live session #4

Debunking the Blockchain Trilemma, CAP Theorem & Application Scalability	• Myths of Decentralisation • On Decentralisation • The Wizard of Blockchain • Cost Performance Trade-Off Evaluation in Microservices impacted by the CAP Theorem Limitations
Working Blockchain & Overlay Networks	• A Survey and Comparison of P2P Overlay Network Schemes. • Virtual Networking Explained • What is a Content Delivery Network • Mandala Network • SPV • Working Blockchain

Course pre-requisite: Bitcoin Development Chapter 2&3
Worksheet to complete prior to the live session: Week 6 Debunking the Blockchain Trilemma, CAP Theorem & Week 7 Application Scalability and Working Blockchain & Overlay Networks.

Live session #5

Intro to Git and Github	• Git and GitHub for Beginners • Getting Started With OpenSSH Key Management. • Setting up an Nx monorepo with Angular • Setting up CI/CD with Github Actions and Vercel
Constructing Transactions & Script	• Introduction to Bitcoin Transactions • MintBlue API, SDK and Integrations

Course pre-requisite: Bitcoin Development Chapter 4-5
Worksheet to complete prior to the live session: Week 8 Intro to Git and Github and Week 9 Constructing Transactions and Bitcoin Script.

Live session #6

Metanet	• Metanet Overlay • Dagda • The Birth of Ontology & the DAG • Tutorial on directed Acyclic Graphs • A.N.N.E preview.
End of programme	• Feedback on project completed by students • Wrap up of the programme

Course pre-requisite: Bitcoin Development Chapter 6
Worksheet to complete prior to the live session: Week 10 Metanet

Blockchain Innovation Program

Worksheets

Week 8: Bitcoin Script and Transactions

1. **Introduction to Bitcoin Transactions**

INTRODUCTION TO
BITCOIN
TRANSACTIONS
WEDNESDAY 13th JULY
3PM EST

a. What is a predicate?
b. Does the predicate have to be based upon public and private ECDSA key?
c. Why does bitcoin script not have declarative loop statements?
d.

nLockTime	
Version	
Input Count	
Output	
Input	
Output Count	

Match the strings with their data items.

i. 67e7105b52e8534596af29dba949921cffe3dbaa555b8ed96121346c6755adae000000006a47304402206e4db9dee8449b861e5fdc00ba3bdb80fba8cd52c75489376c54bd65d26262650220453569438e6bc6f957b1f7ff6fff4af2e42edaae1ac885382373d42fa569b17c41210267d2d1f8b3affffa10b68b2756ba7f6f4efafcadbecd145181016178d00b379bffffffff

ii. 01

iii. 01

iv. 00000000

v. 01000000

vi. 9c276bee0000000976a914accd105073775756cc04962bc1e4893694f50c5588ac

e. What distinguished the input from the output?

2. More proof that UTXO is superior to account-based systems

https://zemgao.com/more-proof-that-utxo-is-superior-to-account-based-systems/

a. What is the difference between UTXO and account-based systems?

b. What is the back to genesis problem?

3. Learn Forth

https://skilldrick.github.io/easyforth/#introduction

https://wiki.bitcoinsv.io/index.php/Opcodes_used_in_Bitcoin_Script

a. Which opcodes would perform the same functionality as the pre-defined words dup, drop, swap and rot in Forth?

4. Smart Contracts for Bitcoin SV

https://by-example.scrypt.io/

a. How does a high level language like sCrypt work in relation to bitcoin script?

b. Which of the following above examples demonstrate that the bitcoin system can act as a Turing Machine?