BcU Certified Blockchain Ethereum Professional (CBEP) Syllabus

Released
Version 1.0 (in 2019)

Blockchain United

Copyright Notice

This document may be copied in its entirety, or extracts made, if the source is acknowledged. All Blockchain United resources including this document are © of Blockchain United (hereafter referred to as BcU).

Revision History

Version	Date	Remarks
1.0	January 2019	First official release

Table of Contents

Purpose of this document

This syllabus forms the basis of Blockchain Untied in regards to its Certified Blockchain Ethereum Professional (CBEP) certification. This document defines what you need to know in order to pass the certification test for Certified Blockchain Ethereum Professional (CBEP), and is copyright of Blockchain United. The certification exam will only cover concepts and knowledge that are described in this document.

Resources of the BcU

An overview of BcU resources as well as all relevant information about the BcU certification and other types of BcU certifications is available on www.blockchain-united.org – the official website of Blockchain United. The information on www.blockchain-united.org includes:

- A complete list of recognized BcU training providers and available courses. Note that training is recommended but not required in order to take the BcU CBEP certification exam.
- The BcU Syllabus (this document) for download.
- A sample exam set of 10 BcU CBEP questions with answers for training purposes – all recognized training providers also have a full 40-question mock exam for training purposes.
- We aim to have the documents available in further languages as soon as possible. For currently available language versions, please check www.blockchain-united.org.

What is Ethereum?

Ethereum is the world's largest Blockchain platform for building distributed, smart contract driven Blockchain applications. At the same time, Ethereum is undergoing rapid changes to its API, consensus mechanisms and the solidity programming language used for writing smart contracts. Many online learning resources tend to become outdated very quickly. For someone wishing to learn Ethereum, it becomes quite challenging to find up to date courses and learning content. To solve this problem, we have designed this course to enable participants to learn from the latest version of the platform. We keep the course up to date by continuously updating the content whenever changes occur.

The key objective of the course is to help you go from zero knowledge of Blockchain or Ethereum, to learn all the key concepts and get started with writing Ethereum smart contracts and DApps (distributed applications) within 3 days. After taking this course, you should be able to build DApps and smart contracts for different use cases.

About BcU Certified Blockchain Ethereum Professional (CBEP)

BcU Certified Blockchain Ethereum Professional (CBEP) is a practitioner level course for anyone interested in learning the fundamentals of Ethereum Blockchain and start building decentralized distributed smart contract driven applications. This course will teach the participants to write Ethereum smart contracts using the Solidity programming language, and to create the frontend UI using JavaScript and the Web3.js library.

Business Outcomes

Business Outcomes (BOs)

BO 1	Identify use cases where a Blockchain can be used.
BO 2	Select the right consensus mechanism for different types of Blockchain application.
BO 3	Decide whether Ethereum fits a specific business use case.
BO 4	Work with product managers to plan and build a proof of concept Blockchain application.
BO 5	Setup the development environment and workflow for other developers in a team.
BO 6	Deploy DApps to AWS cloud
BO 7	Design and write smart contracts for different use cases.
BO 8	Design and write tests for testing smart contracts.
BO 9	Design and build complete DApp (distributed/decentralized application) with smart contracts and Frontend UI using ReactJS (Next.js).

Learning Objectives/Cognitive Levels of Knowledge

Learning objectives (LOs) are brief statements that describe what you are expected to know after studying each chapter. The LOs are defined as follows:

- K1: Remember
- K2: Understand
- K3: Apply

The main list of LOs for the BcU CBEP certification:

LO1	Explain what a Blockchain is and how it works under the hoods. (K1)
LO2	Understand the different types of Blockchain platforms and which ones to use for specific use cases. (K2)

LO3	Understand how smart contracts work and what problems they solve. (K2)
LO4	Understand what kinds of Blockchain projects are a good fit to build on Ethereum. (K2)
LO5	Explain how Ethereum works. (K2)
LO6	Explain how Ethereum cryptography works at a high level (K2)
LO7	Explain the consensus algorithms in Ethereum (K2)
LO8	Understand the working of different Ethereum networks (K2)
LO9	Install and administer Geth. (K3)
LO10	Install and use Ethereum wallets for development and transactional use cases. (K3)
LO11	Setup the application development environment. (K3)
LO12	Understand and use the Solidity programming language for writing smart contracts (K3)
LO13	Write smart contracts without using any framework. (K3)
LO14	Write smart contracts using the truffle framework. (K3)
LO15	Write unit tests to test and ensure the correctness of smart contract functionality. (K3)
LO16	Write complete smart contract driven applications that end-users can interact with through beautiful front-end UIs built using ReactJS (Next.js). (K3)
LO17	Deploy DApps on AWS cloud. (K3)
LO18	Understand some of the best practices when building DApps on Ethereum. (K2)
LO19	Understand Ethereum Request for Comments and Ethereum Improvement Proposals (K2)
LO20	Understand the structure and working of common token contracts (K2)

JavaScript Knowledge Prerequisites

Ideal: At least 2 years' experience with any object-oriented programming language and basic knowledge of JavaScript and web development.

Participants must have knowledge of the below concepts:

- Writing JavaScript functions with callbacks
- Writing modern JavaScript with promises and async/await
- Basic HTML and CSS
- Basic understanding of how CommonJS modules are written in NodeJS applications
- Ability to use npm to install and use JavaScript libraries
- Basic understanding of using JSON files for configuration
- Conceptual understanding of JavaScript tools like Webpack, Gulp, or Babel

Chapter 1 – Introduction to Blockchain

LO1	Explain what a Blockchain is and how it works under the hoods. (K1)
LO2	Understand the different types of Blockchain platforms and which ones to use for specific use cases. (K2)

Summary:

- Why do we need Blockchain?
- How does Blockchain work?
- Blockchain vs Traditional Storage
- Types of Blockchain Platforms
- Consensus Models

Background:

Case study: Why do we need a Blockchain?

In this section, you will understand the scenario in which using a Blockchain makes sense for a specific business use case. You will work through a real-life case study and understand the problems that can occur when you do not use a Blockchain. You will then work through the problem to design the ideal architecture that uses Blockchain for the business use case.

How does a Blockchain work, exactly?

Learning about Blockchain by reading and working through diagrams and presentations is fine. But the real fun is in learning with practical examples. In this section, we will use a demo to explore the various components and working of a Blockchain system.

Blockchain vs. Traditional storage

It is important to understand when to use a Blockchain, as well as how it compares to using a standard database like RDBMS or NoSql. In this section, you will get answers to three important questions:

- How does a Blockchain compare with traditional storage mediums such as RDBMS or NoSql?
- What are the pros and cons?
- When do we use a Blockchain vs. a database?

Types of Blockchain Platforms

There are hundreds of Blockchain platforms available for building applications. Some are designed for a general purpose while others for specific use cases. It is extremely important to choose the right platform for a given use case. Choosing wrong will lead to fighting the platform and to compromises in the application design and architecture. In this section, you will understand the different types of Blockchain platforms and how to select the right one for various use cases.

Consensus Models

The consensus mechanism in a Blockchain is very important. In this section, you will learn some of the popular consensus models and how they can impact DApp design and scalability. You will also learn why we need consensus in a Blockchain, as well as how various consensus models fit different use cases.

Chapter 2 – Smart Contracts

LO3	Understand how smart contracts work and what problems they solve. (K2)

Summary:

- What are smart contracts?
- Why and when should smart contracts be used?

Background:
What are Smart Contracts?
Smart contracts are an exciting feature of Blockchain, and most real-world applications use smart contracts in some form. In this section, you will learn the basics of smart contracts, and how they help in automation and elimination of third parties in business processes.

Why and when do we need Smart Contracts?
It is easy to get carried away and try to build everything in an application as a smart contract. However, not everything is meant to be built as a smart contract. You will learn the key scenarios where a smart contract must be used and when they must be avoided.

Chapter 3 – Working with the Ethereum Platform, Tools and Development Environment

LO4	Understand what kinds of Blockchain projects are a good fit to build on Ethereum. (K2)
LO5	Explain how Ethereum works. (K2)
LO6	Explain how Ethereum cryptography works at a high level (K2)
LO7	Explain the consensus algorithms in Ethereum (K2)
LO8	Understand the working of different Ethereum networks (K2)
LO9	Install and administer Geth. (K3)
LO10	Install and use Ethereum wallets for development and transactional use cases. (K3)
LO11	Setup the application development environment. (K3)

Summary:

- The Business case for Ethereum
- The Ethereum Ecosystem
- Ethereum concepts, virtual machines and architecture
- Ethereum cryptography
- Consensus algorithms in Ethereum
- Ethereum networks
- Ethereum wallets and transactions
- Installing Geth node and node Administration
- Ethereum DApp Developer: Tools for Your Trade
- Setting up your Ethereum development environment

Background:

The Business Case for Ethereum

In this section, you will learn the business scenarios and use cases where Ethereum is an ideal fit. You will also understand the pros and cons of Ethereum so that you can make the right choice when asked to develop a Blockchain application.

The Ethereum Ecosystem

The Ethereum world is huge! Let us explore this universe and identify some of the tools and projects that are relevant for building Blockchain applications.

Ethereum Concepts, Virtual Machine and Architecture

In this section, we will go under the hoods and explore the key concepts and components that make up Ethereum. We will explore the architecture of the Ethereum Virtual Machine so that you have a good understanding about how everything functions internally.

Ethereum Cryptography

Cryptography is the foundation on which Blockchain applications are built. In this section, you will learn the important cryptography concepts and algorithms used in Ethereum.

Consensus Algorithms in Ethereum

Ethereum supports multiple consensus models. In this section, you will learn the pros and cons of each one, and the criteria to choose the right one for a use case.

Ethereum Networks

In this section, you will explore the Ethereum Main Network and the different types of test networks available. You will also learn which networks to use when building and testing DApps.

Ethereum Wallets and Transactions

Ethereum has multiple wallets. Learn how to use each one of them and when to use the right wallet for specific use cases.

Installing a Node and interacting with it

You will install the Geth client and configure it on your local machine. You will also learn how to interact with it and configure it for development. Finally, you will learn about other clients that are available and how they compare with Geth.

Tools of the Trade

There are a number of tools that you can use to build smart contract driven applications. However, the variety of choices can be quite confusing. In this section, you will learn and use tools such as Ganache, Remix, Truffle, Webpack, etc. that will help you quickly build, compile, test and deploy Blockchain applications. We will show how to do rapid development with the various tools, and also the pros and cons of each one.

Setting up the Ethereum Development Environment

Having built multiple products on Ethereum, we have identified the best development setup that will help you start working quickly without installing too many tools and software. In this section, you will use this setup for building Ethereum smart contracts and DApps in the rest of this course.

Chapter 4 - Writing DApps (Distributed / Decentralized Applications)

LO12	Understand and use the Solidity programming language for writing smart contracts (K3)
LO13	Write smart contracts without using any framework. (K3)
LO14	Write smart contracts using the truffle framework. (K3)
LO15	Write unit tests to test and ensure the correctness of smart contract functionality. (K3)
LO16	Write complete smart contract driven applications that end-users can interact with through beautiful front-end UIs built using ReactJS (Next.js). (K3)
LO17	Deploy DApps on AWS cloud. (K3)
LO18	Understand some of the best practices when building DApps on Ethereum. (K2)

Summary:

- Solidity
- Writing smart contracts without frameworks
- Test driven smart contract development
- Writing smart contracts with Truffle framework
- Build a complete Blockchain application
- Best Practices

Background:
Writing Smart Contracts using Solidity
Solidity is a programming language that is designed specifically to write smart contracts for Ethereum. In this section, you will also learn why solidity was created to build smart contracts instead of using an existing programming language. You will also learn the key solidity language types and constructs used in writing smart contracts. This is an extremely important section, in which you will learn how to write different types of smart contracts and best practices.

Writing Smart Contracts without Frameworks
Frameworks are designed to make us productive and get results quickly. At the same time, frameworks such as truffle undergo fast developments and changes, and sometimes this leads to bugs or issues that can break your existing workflow. In this section, you will build a mini framework from scratch to help with your DApp development.

Testing Driven Smart Contract Development
Writing tests is crucial in order to ensure that your smart contracts function as intended. You will learn the different ways to write tests and run them before deploying your application to a network.

Writing Smart Contracts with Truffle Framework
Now that you have a very strong foundation of working without a framework, you will learn the truffle framework and how it helps speed up your development.

Build a complete Blockchain application
This is the most exciting part! You will build a complete smart contract driven application from start to finish. Once you are done, you will deploy the completed application to AWS cloud.

Best Practices
It is easy to make mistakes when building DApps by thinking of them as regular web applications. In this section, let us explore some of the best practices used in the Blockchain domain so that they are secure and provide a good user experience for end users.

Chapter 5 – Tokenomics

LO19	Understand Ethereum Request for Comments and Ethereum Improvement Proposals (K2)
LO20	Understand the structure and working of common token contracts (K2)

Summary:

- Ethereum Request for Comments (ERC)
- Ethereum Improvement Proposals (EIP)

Background:

In this section, you will learn what are ERCs and EIPs. We will explore ERC specifications like ERC-20, ERC-223, ERC-721 and ERC-777 and their respective use cases.

What next?

Now that you are capable of writing smart contract driven applications using Ethereum, where do you go from here? We will share links to additional learning resources that will help you move forward in your Blockchain journey.

References:

- Ethereum Documentation: http://www.ethdocs.org/en/latest/
- Solidity Documentation: https://solidity.readthedocs.io/en/latest/index.html