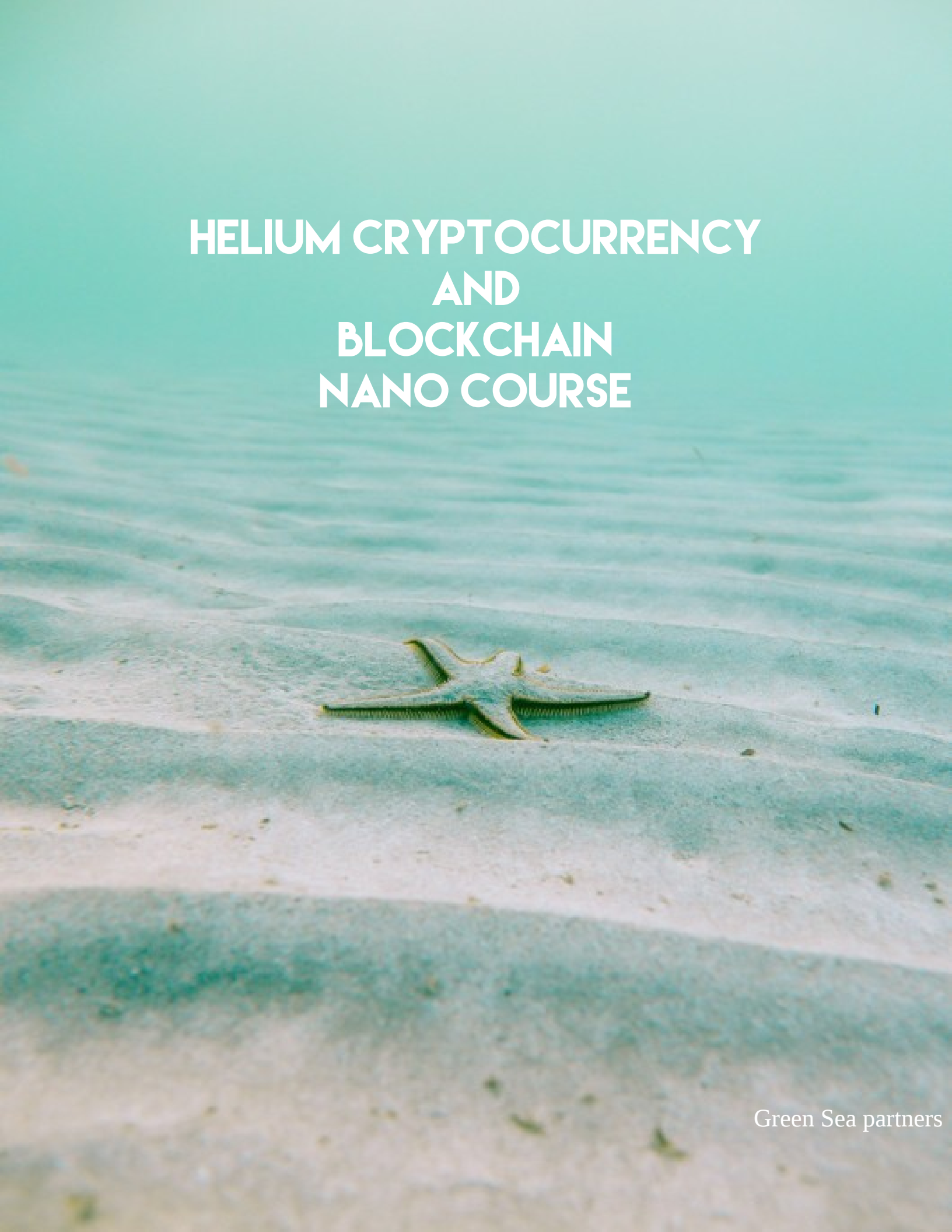
HELIUM CRYPTOCURRENCY AND BLOCKCHAIN NANO COURSE



Green Sea partners

The Helium Cryptocurrency Nano Course

Introduction

Blockchains and cryptocurrencies are fundamental paradigm shifts. Consider money or currency. The main characteristics of money is that it is a fungible medium of exchange and a store of value. Fungible means that one unit of a currency is identical to any other unit of that currency (*one dollar is the same as any other dollar)*. A currency is called a fiat currency if the issuing authority does not guarantee that it can be converted at a fixed rate into some other tangible asset such as gold or silver, for example. The US dollar is a prime example of a fiat currency. The value of a US dollar depends solely upon it's acceptance as a medium of exchange and a store of value by the public. It's value does not depend upon a guarantee that it can be converted at a specific fixed rate into a species such as gold.

Cryptocurrencies upset the apple cart. Like a conventional currency, a cryptocurrency is a fungible medium of exchange and a store of value. But here is the crucial difference. A cryptocurrency is not issued by some central authority who can print the currency at will. The issuance of a cryptocurrency is distributed. In Bitcoin, it's issuance is governed by Mathematical algorithms. Typically issuance is correlated with the value of transactions transpiring on the cryptocurrency network. Furthermore, the amount of a new cryptocurrency that is generated decreases over time and then reaches a maximum. As an example, the bitcoin network will cease issuing new bitcoins some time in the year 2041 at which point in time there will be a maximum of twenty one million bitcoins in circulation. You can surmise from this that cryptocurrencies are inherently deflationary. A person holding a cryptocurrency

similar to Bitcoin which is accepted as a medium of exchange and a store of value will see the value of his cryptocurrency holdings appreciate over time. You may be inclined to ask whether a cryptocurrency can be shut down and the short answer is no since it is a completely decentralized currency with no central node or nodes of control. Interestingly, the inventor of cryptocurrencies, Satoshi Nakamoto, has disappeared from the face of the earth.

One of the basic features of conventional currency transactions is that these transactions are in principle untraceable as long as you do not use banking institutions to consummate transactions. Bitcoin does not guarantee complete anonymity. It is a pseudo-anonymous currency. Bitcoin transactions are traceable unless you take careful measures to ensure your anonymity. On the other hand, transactions with the cryptocurrency Monero are completely untraceable.

You can surmise from this exposition that cryptocurrencies have the potential of completely reorganizing societies into decentralized, distributed direct democracies. Simply because those who have controlled the money supply in history have controlled history.

A blockchain can be concisely described as a distributed, tamper-proof public ledger. Let's break this down with a concrete example. If you have an account with a bank, you rely upon the bank to maintain the integrity of your account. In other words, your relationship with your bank is based upon trust. You would not put your money in the bank if you did not trust it. The cost of maintaining the integrity of your bank account is reflected in the service charges that you pay every month. These charges are significant. Furthermore, banking transactions are terribly slow. A

typical transaction can take twenty four hours or more to clear as your bank and the government cooperate to vet your transaction (*the so-called AML/KYC rules*). But what if you did not need a bank to maintain your account and did not have to rely upon trusting anyone. Furthermore, what if your transaction fees were negligible if not zero and transactions were nearly instantaneous. All of this arcane magic is made possible with blockchains and distributed consensus algorithms. Blockchains are distributed ledgers that are guaranteed to be tamper proof. The distributed property means that no one entity controls the blockchain. Secondly, in order to make an entry to the blockchain you do not have to trust anyone and in particular you do not need to know the physical identity of the person with whom you are transacting. The tamper-proof property of blockchains also implies that transactions on the blockchain cannot be counterfeited. Blockchains and distributed consensus algorithms are a very significant advance in fundamental knowledge. And as you can surmise, blockchains strike at the very core of banking institutions. But beyond this, blockchains enable the censor-resistant distribution of information. The TRON project is an example of this (*https://tron.network*). The objective of TRON is to enable content creators to distribute their content without having to go through the gates of centralized entities such as Facebook, Google, Microsoft and so forth.

The Helium Cryptocurrency Course

The concepts of cryptocurrencies and blockchains depend on theorems in the mathematical field of Cryptology. No more than a dozen theorems underly the entire theoretical basis for cryptocurrencies and blockchains. And underlying these theorems are the properties of prime numbers. We are going to deep dive into cryptology, blockchains and cryptocurrencies. The objective of this course is to equip

you with all of the crypto and practical knowledge required to develop blockchain and cryptocurrency applications. If you know high school math you should be able to follow the course, providing that you are motivated.

This course will walk you through very carefully step by step in the construction of a cryptocurrency and blockchain application. By the end of this course you will have a working cryptocurrency. The ambitious target of this course is to bootstrap the developer with zero knowledge to the level of an advanced blockchain developer. In order to accomplish this objective, two books are included in the course. *Python Rocket Science* teaches the Python language from scratch. *The Developers Guide To Cryptocurrencies And Blockchains* provides you with all of the theory that you need in Cryptology, blockchains and cryptocurrencies.

Resources for this course include:

- Complete source code with exhaustive inline comments
- Presentation slides which guide you through the course
- *The Developers Guide To Cryptocurrency And Blockchains* book
- *Python Rocket Science* book

Software developers know that time is a scarce and valuable commodity. Everyday we are faced everyday with a deluge of new information in Computer Science. Bearing this in mind, we have designed this course to be economical, lucid and to the point. After you finish this course, you should have a clear understanding of cryptocurrencies and should be able to initiate your blockchain startup in C, C++, Java, Python, Erlang or any other suitable language of your choice.

The blockchain revolution is underway, so lets get on with it.

green sea partnership

Syllabus

Preface

The presentation slides are your mentor and guide you through the course.

0. **Python Language Review**

   You will need to be familiar with the Python language constructs in pages 9-37, 41-86, 102-105 and 110-117.

1. **Introduction**

   - What are cryptocurrencies
   - The Helium Cryptocurrency Project
   - Resources for this course
   - Read the Preface in the book: *Developers Guide To Cryptocurrencies And Blockchains*

2. **Environment Setup**

   - Installing Python
   - Installing the virtual environment
   - Installing modules
   - Copying the source code
   - Testing the setup
   - Follow the instructions in README.txt

3. **Cryptology Module**

   - motivation
   - Read chapters on Cryptographic Hash Functions, Public Key Cryptography and Digital Signatures in *Developers Guide To Cryptocurrencies And Blockchains*
   - *rcrypt.py* code walkthrough

4. **Cryptocurrency Configuration**
    - Common cryptocurrency global parameters and what they do
    - *config.py* code walkthrough

5. **Blockchain Design**
    - block structure
    - linking blocks in a blockchain
    - block validation and other blockchain functions
    - previous block hashes
    - Read chapter on Blockchains in *Developers Guide To Cryptocurrencies And Blockchains*
    - Read the chapter on Cryptocurrency Addresses And Anonymity in *Developers Guide To Cryptocurrencies And Blockchains*
    - blockchain.py code walkthrough

6. **Blockchain Networks**
    - Network nodes
    - bootstrapping nodes
    - block and transaction propagation on the network
    - Read the chapter on Blockchain Networks in *Developers Guide to Cryptocurrencies and Blockchains*

7. **Transactions**
    - transaction structure
    - transaction validation
    - transaction functions
    - merkle tree construction
    - Stack machine construction and operation
    - Unlocking transaction values
    - Read Chapter on Cryptocurrency Transactions in *Developers Guide to Cryptocurrencies and Blockchains*
    - *tx.py* code walkthrough
    - *stackmachine.py* walkthrough

8. **Mining**
    - The purpose of mining
    - Proof of work algorithms
    - How blocks are mined
    - Adding mined blocks to the blockchain
    - Read Chapter on Mining in *Developers Guide to Cryptocurrencies and Blockchains*
    - *mining.py* code walkthrough

9. **Distributed Consensus Algorithm**
    - Blockchain forks
    - Distributed blockchain convergence
    - Trust
    - How the blockchain convergence algorithm works
    - Read Chapter on The Distributed Consensus Algorithm in *Developers Guide to Cryptocurrencies and Blockchains*
    - *mining.py* code walkthrough

10. **UTXO**
    - What is the purpose of the chainstate database
    - Structure of the chainstate database
    - Chainstate database functions
    - *utxo.py* code walkthrough

11. **Unit Tests**
    - Read the chapter on Debugging Python in *Python Rocket Science*
    - *unit_tests* folder code walkthrough

12 **Testnet**
    - Designing a blockchain simulator for testing
    - *testnet* folder code walkthrough