# BLOCKCHAINS

## ARCHITECTURE, DESIGN AND USE CASES


**SANDIP CHAKRABORTY**
COMPUTER SCIENCE AND ENGINEERING,
IIT KHARAGPUR

**PRAVEEN JAYACHANDRAN**
IBM RESEARCH,
INDIA

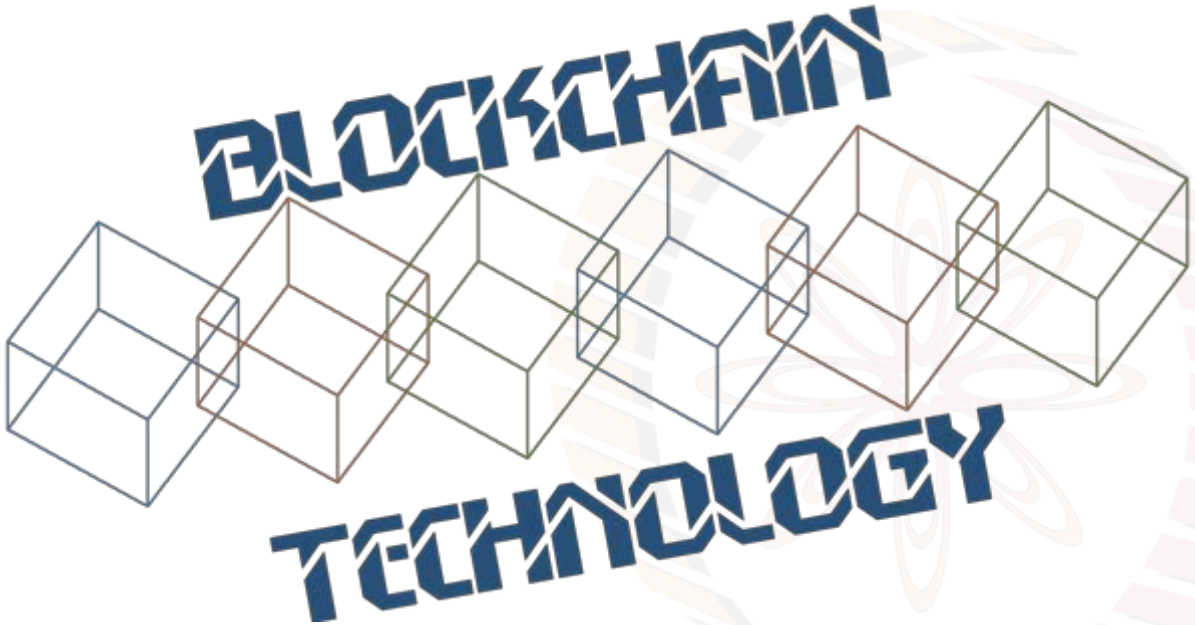



Image courtesy: http://beetfusion.com/

# BITCOIN BASICS I

# Bitcoin – The Beginning

- "A **decentralized** digital currency enables instant payments to anyone, anywhere in the world" – en.bitcoin.it

- No central authority, uses peer-to-peer technology

- Two broad operations
  - **Transaction Management** – transfer of bitcoins from one user to another
  - **Money Issuance** – regulate the monetary base

# Bitcoin Basics – Creation of Coins

- **Controlled Supply:** Must be limited for the currency to have value – any maliciously generated currency needs to be rejected by the network

- Bitcoins are generated **during the mining** – each time a user discovers a new block

- The rate of block creation is adjusted every 2016 blocks to aim for a **constant two week adjustment period**

**Information Source: https://en.bitcoin.it/wiki/**

# Bitcoin Basics – Creation of Coins

- The number of bitcoins generated per block is set to decrease **geometrically**, with a 50% reduction for every 210,000 blocks, or approximately 4 years

- This reduces, with time, the amount of bitcoins generated per block
  - Theoretical limit for total bitcoins: Slightly less than *21 million*
  - Miners will get less reward as time progresses
  - How to pay the mining fee – increase the transaction fee

**Information Source: https://en.bitcoin.it/wiki/**

# Projected Bitcoins

| Date reached | Block | Reward Era | BTC/block | Year (estimate) | Start BTC | BTC Added | End BTC | BTC Increase | End BTC % of Limit |
|---|---|---|---|---|---|---|---|---|---|
| 2009-01-03 | 0 | 1 | 50.00 | 2009 | 0 | 2625000 | 2625000 | infinite | 12.500% |
| 2010-04-22 | 52500 | 1 | 50.00 | 2010 | 2625000 | 2625000 | 5250000 | 100.00% | 25.000% |
| 2011-01-28 | 105000 | 1 | 50.00 | 2011* | 5250000 | 2625000 | 7875000 | 50.00% | 37.500% |
| 2011-12-14 | 157500 | 1 | 50.00 | 2012 | 7875000 | 2625000 | 10500000 | 33.33% | 50.000% |
| 2012-11-28 | 210000 | 2 | 25.00 | 2013 | 10500000 | 1312500 | 11812500 | 12.50% | 56.250% |
| 2013-10-09 | 262500 | 2 | 25.00 | 2014 | 11812500 | 1312500 | 13125000 | 11.11% | 62.500% |
| 2014-08-11 | 315000 | 2 | 25.00 | 2015 | 13125000 | 1312500 | 14437500 | 10.00% | 68.750% |
| 2015-07-29 | 367500 | 2 | 25.00 | 2016 | 14437500 | 1312500 | 15750000 | 9.09% | 75.000% |
| 2016-07-09 | 420000 | 3 | 12.50 | 2016 | 15750000 | 656250 | 16406250 | 4.17% | 78.125% |
| 2017-06-23 | 472500 | 3 | 12.50 | 2018 | 16406250 | 656250 | 17062500 | 4.00% | 81.250% |
| | 525000 | 3 | 12.50 | 2019 | 17062500 | 656250 | 17718750 | 3.85% | 84.375% |
| | 577500 | 3 | 12.50 | 2020 | 17718750 | 656250 | 18375000 | 3.70% | 87.500% |
| | 630000 | 4 | 6.25 | 2021 | 18375000 | 328125 | 18703125 | 1.79% | 89.063% |
| | 682500 | 4 | 6.25 | 2022 | 18703125 | 328125 | 19031250 | 1.75% | 90.625% |
| | 735000 | 4 | 6.25 | 2023 | 19031250 | 328125 | 19359375 | 1.72% | 92.188% |
| | 787500 | 4 | 6.25 | 2024 | 19359375 | 328125 | 19687500 | 1.69% | 93.750% |



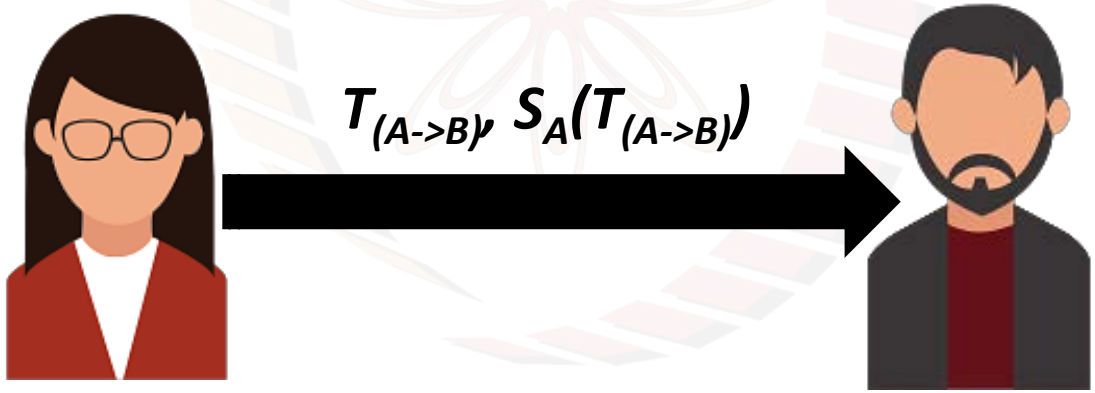
Information Source: https://en.bitcoin.it/wiki/

# Bitcoin Basics – Sending Payments

- Need to ensure that Eve cannot spend Alice's bitcoins by creating transactions in her name.
- Bitcoin uses **public key cryptography** to make and verify digital signatures.
- Each person has one or more addresses each with an associated pair of public and private keys (may hold in the bitcoin wallet)

# Bitcoin Basics – Sending Payments

- Alice wish to transfer some bitcoin to Bob.
  - Alice can sign a transaction with her private key
  - Anyone can validate the transaction with Alice's public key



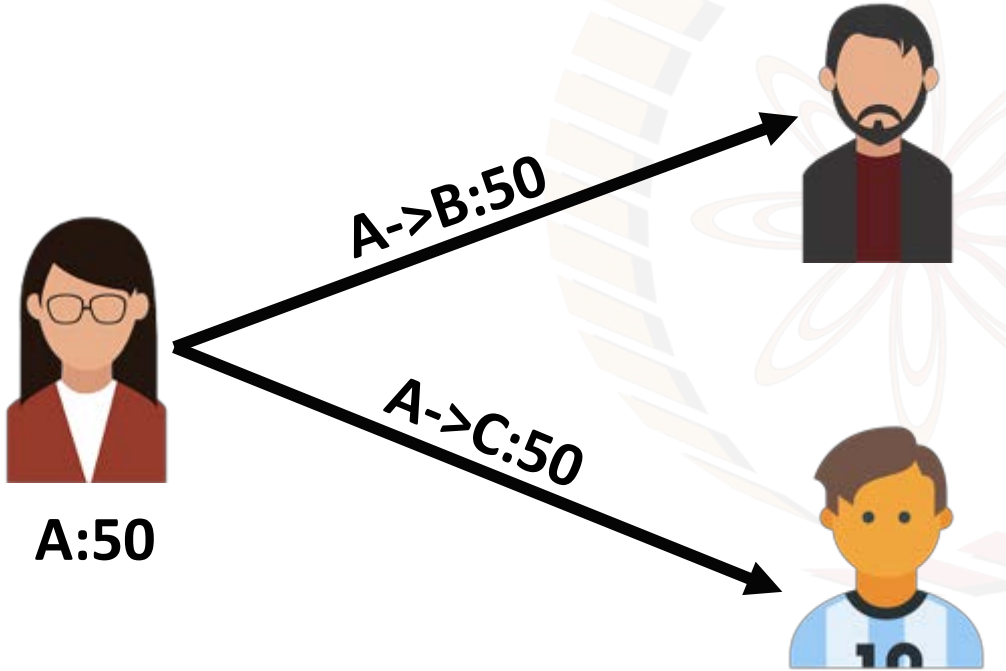
# Bitcoin Basics – Sending Payments

- Alice wants to send bitcoin to Bob
  - Bob sends his address to Alice
  - Alice adds Bob's address and the amount of bitcoins to transfer in a "transaction" message
  - Alice signs the transaction with her private key, and announces her public key for signature verification
  - Alice broadcasts the transaction on the Bitcoin network for all to see

**Information Source: https://en.bitcoin.it/wiki/**

# Double Spending

- Same bitcoin is used for more than one transactions




- In a centralized system, the bank prevents double spending
- How can we prevent double spending in a decentralized network?

# Handle Double Spending using Blockchain

- Details about the transaction are sent and forwarded to all or as many other computers as possible

- Use **Blockchain** – a constantly growing chain of blocks that contain a record of all transactions

- The blockchain is maintained by all peers in the Bitcoin network – everyone has a copy of the blockchain

**Information Source: https://en.bitcoin.it/wiki/**

# Handle Double Spending using Blockchain

- To be accepted in the chain, transaction blocks must be valid and must include **proof of work** – a computationally difficult hash generated by the mining procedure

- Blockchain ensures that, if any of the block is modified, all following blocks will have to be recomputed

**Information Source: https://en.bitcoin.it/wiki/**

# Handle Double Spending using Blockchain

- When multiple valid continuation to this chain appear, only the longest such branch is accepted and it is then extended further (**longest chain**)

- Once a transaction is committed in the blockchain, everyone in the network can validate all the transactions by using Alice's public address

- The validation prevents double spending in bitcoin

**Information Source: https://en.bitcoin.it/wiki/**

# Bitcoin Anonymity

- Bitcoin is permission-less, you do not need to setup any “account”, or required any e-mail address, user name or password to login to the wallet
- The public and the private keys do not need to be registered, the wallet can generate them for the users
- The **bitcoin address** is used for transaction, not the user name or identity

**Information Source: https://en.bitcoin.it/wiki/**

# Bitcoin Anonymity

- A **bitcoin address** mathematically corresponds to a public key based on ECDSA – the digital signature algorithm used in bitcoin

- A sample bitcoin address: 1PHYrmdJ22MKbJevpb3MBNpVckjZHt89hz

- Each person can have many such addresses, each with its own balance
  - Difficult to know which person owns what amount

**Information Source: https://en.bitcoin.it/wiki/**

# Bitcoin Script

- Alice makes a transaction of BTC 20 to Bob. How Bob will claim those transactions?

- A transaction is characterized by two parameters
  - Alice sends some bitcoins: the output (*out*) of the transaction
  - Bob receives some bitcoins: the input (*in*) of the transaction

- We need to determine that a transaction input correctly claims a transaction output

# Bitcoin Script

- A programming language to validate bitcoin transactions
  - A list of instructions recorded with each transaction
  - Describes how the next person can gain access to the bitcoins, if that person wants to spend them

- FORTH-like language, stack based and processed left to write

scriptPubKey: OP_2DUP OP_EQUAL OP_NOT OP_VERIFY OP_SHA1 OP_SWAP OP_SHA1 OP_EQUAL

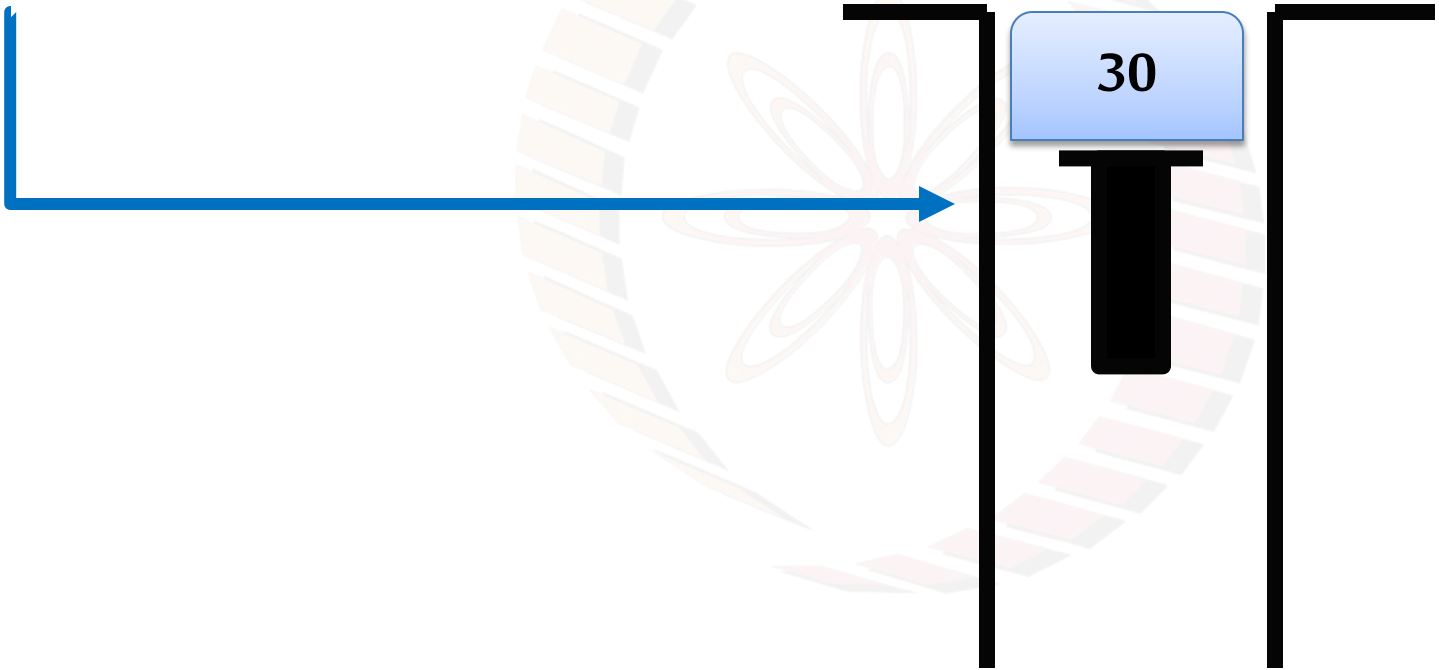
scriptSig: <preimage1> <preimage2>

# How FORTH Works

- A stacked based computer programming language originally designed by Charles Moore
  - A procedural programming language without type checking
  - Use a **stack** for recursive subroutine execution
  - Uses **reverse Polish notation** (RPN) or **postfix notation**

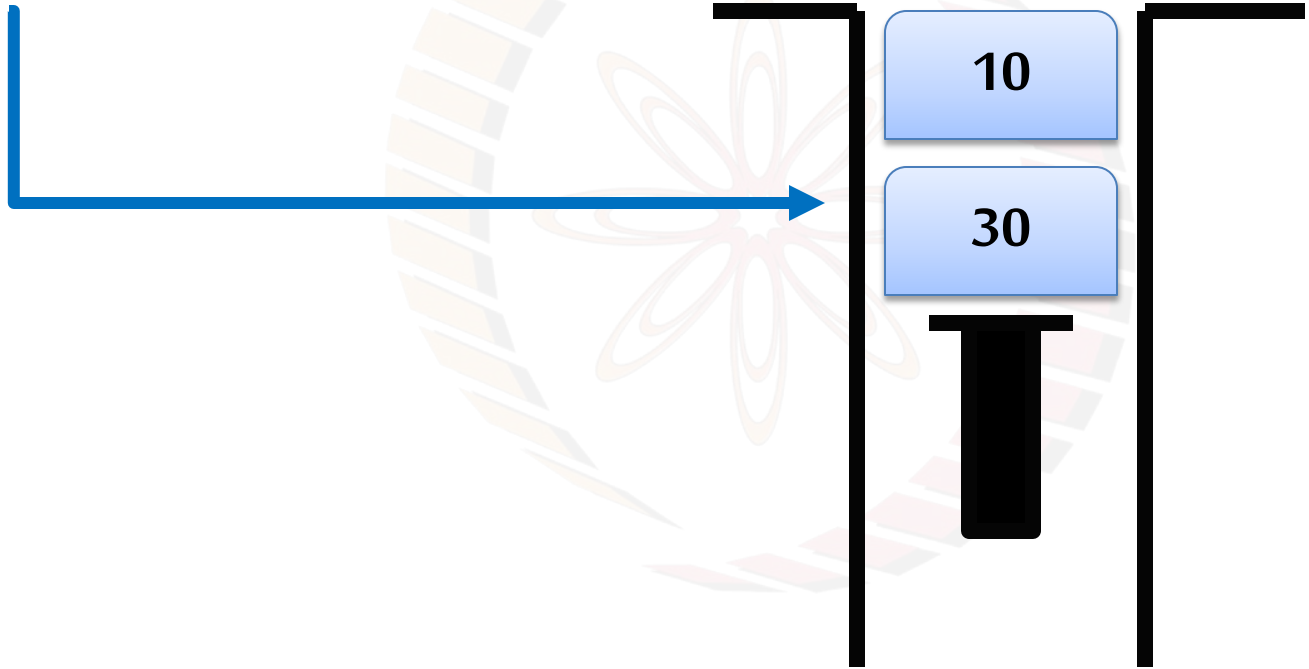
# FORTH – Sample Execution using RPN

**30 10 * 15 + CR .**



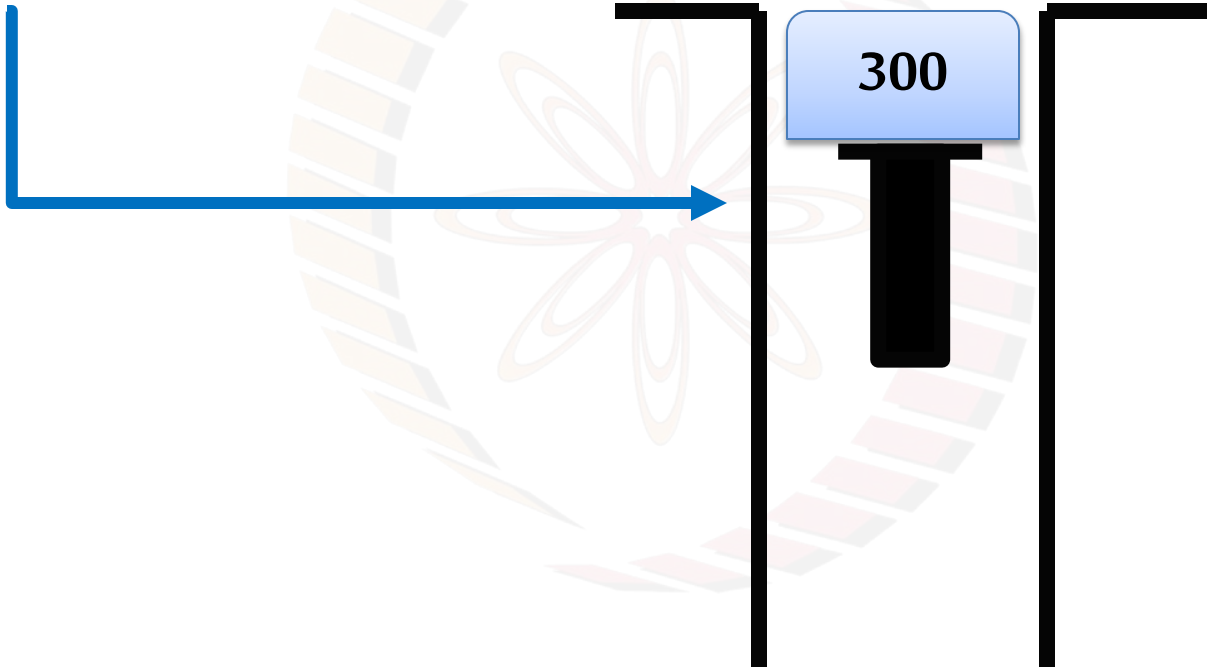
# FORTH – Sample Execution using RPN

30 10 * 15 + CR .



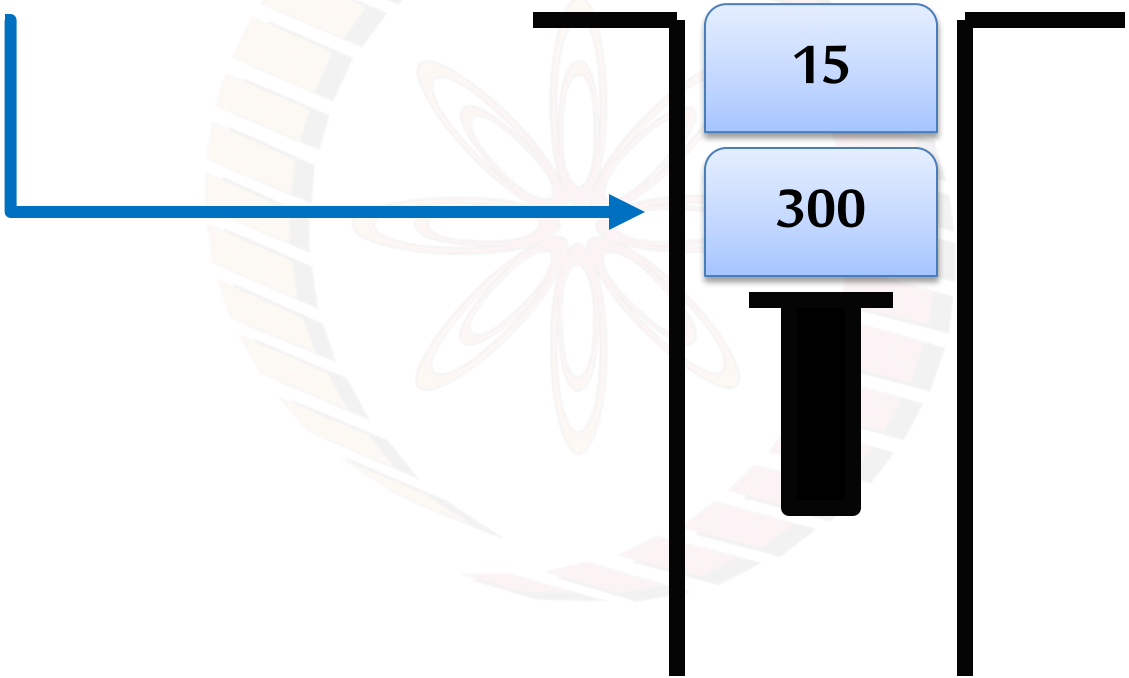
# FORTH – Sample Execution using RPN

**30 10 * 15 + CR .**



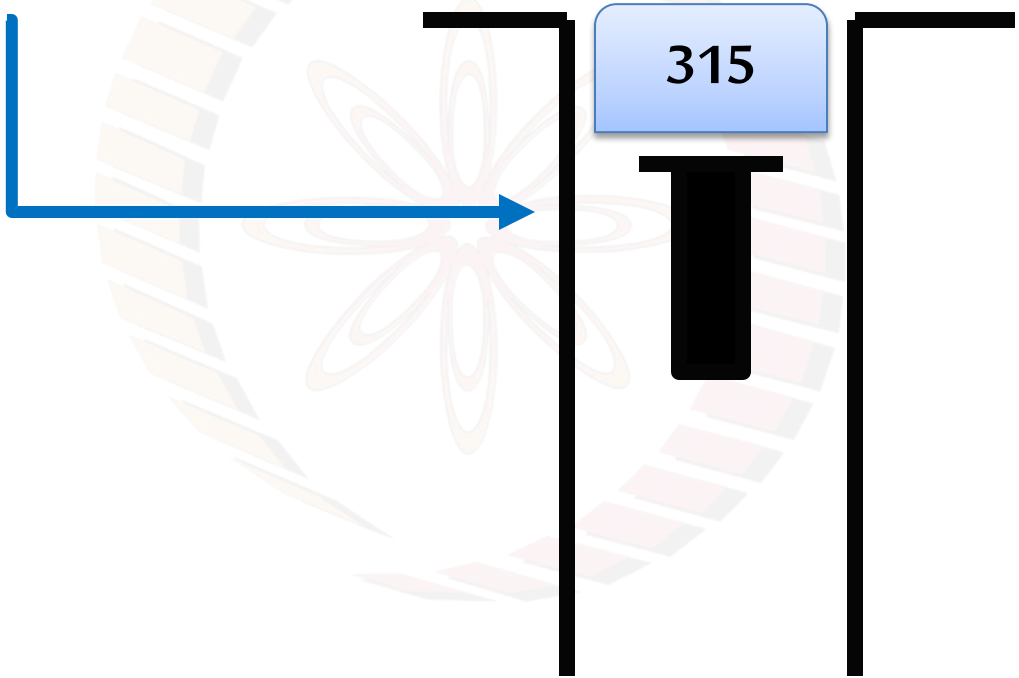
# FORTH – Sample Execution using RPN

**30 10 * 15 + CR .**



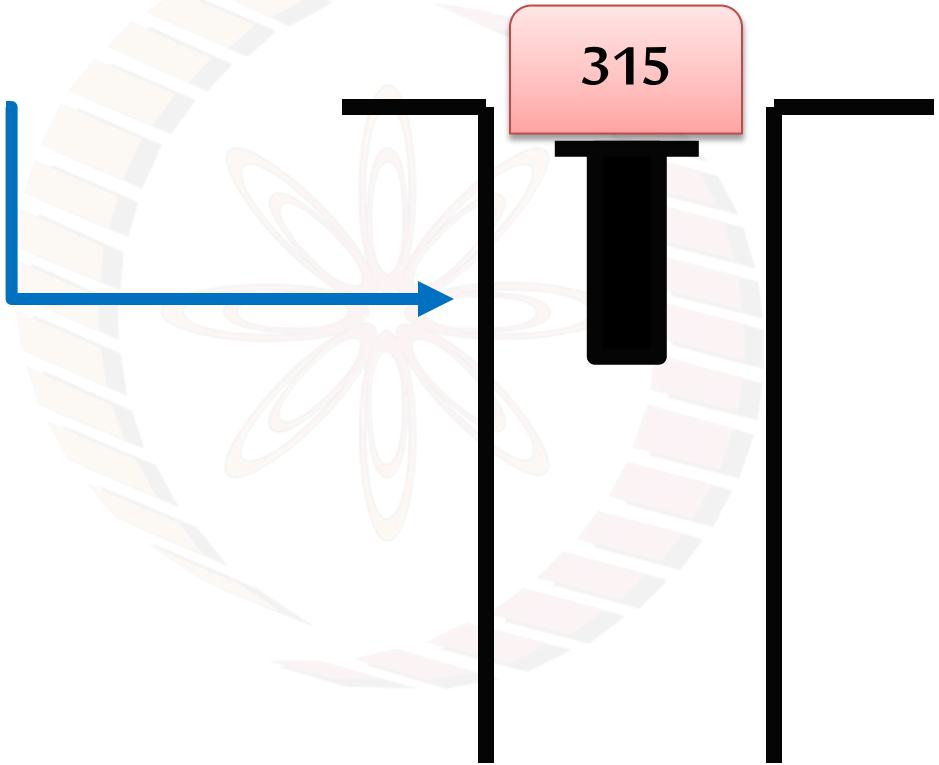
# FORTH – Sample Execution using RPN

**30 10 * 15 + CR .**



# FORTH – Sample Execution using RPN

**30 10 * 15 + CR .**



# FORTH – Sample Code

## FORTH Code:

```
:FLOOR5 (n--n') DUP 6 < IF DROP 5 ELSE 1 – THEN;
```

## Equivalent C Code:

```
int floor5(int v){
        return (v<6)?5:(v-1);
}
```

- Defines a new word (a subroutine) called FLOOR5

**Code Source:** https://en.bitcoin.it/wiki/


BITCOIN • DIGITAL • DECENTRALIZED • PEER TO PEER
1 TROY OZ .999 FINE COPPER
MY MONETARY MEDIUM

thank you!