**Objective**:

- Understand how cryptographic hash functions ensure data integrity and immutability in blockchain.
- Explore practical implementations of cryptographic hash functions to link and verify blocks.
- Demonstrate how small changes in data affect the cryptographic hashes in a blockchain.

**Lab Setup Requirements**:

- Python or any programming language with support for cryptography libraries (e.g., Python's hashlib).
- Access to an online IDE or local development environment (e.g., Jupyter Notebook, VS Code).
- Knowledge of basic programming concepts.

## Part 1: Introduction to Cryptographic Hash Functions

1. **Goal**: Understand how cryptographic hash functions work and how they ensure data integrity by generating unique, fixed-length outputs.
2. **Instructions**:
   - Write a Python program that uses the hashlib library to hash a string of text using the SHA-256 algorithm.
   - Generate hashes for different pieces of data and compare the results. Have them modify the input slightly and observe the large difference in the hash.

- **Expected Output**:
  - Show how a small change (like capitalization) produces a completely different hash, emphasizing the **sensitivity to input changes**.

## Part 2: Linking Blocks with Hashes

1. **Goal**: Simulate how blocks are linked together in a blockchain using hash values. Each block contains data and the hash of the previous block.
2. **Instructions**:
   - Implement a simple blockchain structure where each block includes a data field, a block number, and the hash of the previous block.
   - Add multiple blocks to the chain and observe how tampering with one block affects the rest of the chain.

- **Expected Outcome**:
  - You will see how each block is linked using the hash of the previous block. By modifying data in one block, you will see how the hash changes, which affects all subsequent blocks in the chain.

## Part 3: Understanding Block Tampering and Chain Integrity

1. **Goal**: Demonstrate how tampering with data in any block affects the integrity of the entire blockchain.
2. **Instructions**:

    - Modify the data in one of the existing blocks (e.g., change the data in the second block).
    - Recalculate the hash of the modified block and see how this affects the hash of all subsequent blocks.
- **Expected Outcome**:
    - You will see that when the second block is tampered with, the hash of the third block becomes invalid. You will need to update the third block's hash based on the tampered second block, demonstrating how tampering with one block compromises the entire chain.

## Part 4: Class Discussion and Review

1. **Goal**: Reinforce understanding by discussing the real-world implications of cryptographic hashing in blockchain.
2. **Instructions**:
    - Reflect on the exercises and discuss in small groups
        - Why are cryptographic hashes essential in blockchain?
        - How does blockchain achieve **security** and **trust** without a central authority?
        - How would tampering affect real-world applications like Bitcoin or Ethereum?