PROJECT OUTLINE

1. main.py

- **Purpose**: The entry point of the application.
- **Responsibilities**:
    - Initializes the peer process.
    - Reads configuration files (Common.cfg, PeerInfo.cfg).
    - Starts the communication with other peers.

2. peer.py

- **Purpose**: Handle peer-specific operations.
- **Responsibilities**:
    - Manage the peer's state (has full file, missing pieces, bitfield).
    - Establish TCP connections to other peers.
    - Send and receive messages (handshake, bitfield, choke/unchoke).
    - Coordinate piece downloading.
- **Why**: This file encapsulates everything related to how a peer behaves and interacts.

3. file_manager.py

- **Purpose**: Manage file operations.
- **Responsibilities**:
    - Split the file into pieces based on PieceSize from Common.cfg.
    - Store file pieces in the appropriate directory (e.g., peer_[peerID]).
    - Reassemble file pieces once all are downloaded.
    - Handle file I/O (read/write operations).
- **Why**: Separating file management ensures a clear distinction between communication and file handling logic.

4. message.py

- **Purpose**: Handle message creation, parsing, and validation.
- **Responsibilities**:
    - Create the messages (handshake, choke, unchoke, etc.).
    - Parse incoming messages from other peers.
    - Ensure message formats follow the protocol (message length, type, payload).
- **Why**: This modular design ensures you can easily extend message handling without cluttering other parts of the system.

5. config.py

- **Purpose**: Load and manage configuration files.
- **Responsibilities**:
    - Read Common.cfg and PeerInfo.cfg to get parameters like file name, piece size, and peer details.
    - Provide helper functions to access these configurations during peer operations.
- **Why**: Having a separate file for configuration management makes it easier to change or add configuration options without affecting other parts of the system.

6. logger.py

- **Purpose**: Manage logging for each peer.
- **Responsibilities**:
    - Create log entries for events like connections, choke/unchoke messages, downloading pieces, etc.
    - Write logs to files like log_peer_[peerID].log.
- **Why**: Centralized logging will make it easier to track events and debug issues.