2. Bank: Account Management System

Objective: Manage bank accounts, transactions, and balances using callback functions, error handling, async/await, and ES6 features.

Part 1: Basic Features

Instructions:

1. **Create a class Account** with properties `accountNumber`, `balance`, and `transactions`.
2. **Implement methods to:**
   - Deposit money (with error handling).
   - Withdraw money (with error handling).
   - Fetch account data from a simulated API (using async/await).
   - Save updated account data to a simulated API (using async/await).
3. **Use callback functions** to display updated balance and transactions after fetching and updating data.
4. **Add error handling** to manage simulated API request failures.

Part 2: Advanced Features

Objective: Enhance the account management system using advanced ES6 features, including modules, EventEmitter, and streams.

Instructions:

1. **Organize code using ES6 modules**:
   - Split the code into separate modules for Account, API calls, Logger, and Streams.
2. **Add an event-driven logging system** using EventEmitter:
   - Log significant actions such as fetching data, depositing money, and withdrawing money.
3. **Implement stream operations**:
   - Create readable, writable, duplex, and transform streams to handle account data.
   - Use `pipe()` for efficient data flow.
4. **Enhance error handling** to manage stream operations and API request failures.