Part 1: Book Store

10: .contracts.BookContract class
- 5: Defines column names
- 5: Defines getter (Cursor) and putter (ContentValues) operations

10: .entities.Book entity class
- 5: Book(Cursor) constructor
- 5: writeToProvider(ContentValues) method

35: .databases.CartDbAdapter class
- 5: Useful literals: DATABASE_NAME, BOOK_TABLE, AUTHOR_TABLE,
- CREATE_DATABASE, DATABASE_VERSION
- 5: CREATE_DATABASE follows spec, and is constructed using column names from contract
- 5: Use of SQLiteOpenHelper (subclass) to create/open database
- 10: Definition of adapter operations:
    - `public void open() throws SQLException;`
    - `public Cursor fetchAllBooks();`
    - `public Book fetchBook(long rowId);`
    - `public void persist(Book book) throws SQLException;`
    - `public boolean delete(Book book);`
    - `public boolean deleteAll();`
    - `public void close();`
- 10: Authors in separate table in database, use of join and aggregation (GROUP-BY with GROUP-CONCAT) to return authors in query

5: SimpleCursorAdapter in main activity to display contents of book store

40: Functionality (with video demos)
- 10: Main activity displays books
- 5: Main activity supports deletion of a single book (via context action bar)
- 10: activity to add book, inserts the book into the database
- 5: Sub-activity for viewing details of a book
- 10: exiting app and restarting shows shopping cart still available

Total: 100

Part 2: Chat App

20: Design
10: entity classes for Messages and Peers (entities package)
   (Must define Constructor(Cursor, writeToProvider(ContentValues) for both

5: contract class (contracts package) defines getter and putter operations
5: database adapter (databases package)

30: Functionality (video demonstration)
- 5: Client sent message is received at server
- 5: Server displays a list of messages received so far
- 5: Option menu provides activity for displaying peers
- 5: Selecting a peer provides info about that peer
- 10: Exiting app and restarting shows messages still available

Total: 50