

Faculty of Engineering, University of Jaffna,

Department of Computer Engineering.

EC5080: Software Construction

Lab 05

Duration: 3 Hours

Lecturer: Ms. Sujanthika M

Input/output, error handling and parsing textual formats

Objectives:

- Handle command-line arguments and file I/O operations in Java.
- Parse and validate structured text input with appropriate formatting.
- Implement and manage custom and built-in exceptions for robust error handling.

You are developing a command-line utility to manage a text-based catalog of books in a library. The catalog stores records in the following format:

Book Title: Author Name : ISBN : Number of Copies

Build a Java program that:

- Accepts a .txt file as the first argument.
- Based on following arguments, performs:
 - **Title keywords** → Search for books by title (case-insensitive match).
 - **ISBN (13-digit)** → Print matching book (throws DuplicateISBNException if more than one found).
 - **New book record (title:author:isbn:copies)** → Add to catalog in alphabetical order by title.
- Include the following features:
 - Auto-create files/directories if missing.
 - Output search/add results in well-aligned columns.

Create a structured exception system with a base class BookCatalogException.

Derive the following: You are expected to handle all exceptions in a meaningful order with user-friendly messages

- `InvalidISBNException` – ISBN not 13 digits or contains non-numeric characters
- `DuplicateISBNException` – Duplicate ISBN found
- `MalformedBookEntryException` – Book entry has missing/empty fields
- `InsufficientArgumentsException` – Fewer than two arguments given
- `InvalidFileNameException` – First argument not ending in `.txt`

After execution, your program must display the,

- Number of valid records processed
- Number of search results
- Number of books added
- Number of errors encountered

And always print this message at the end (even after exceptions):

Thank you for using the Library Book Tracker.

Log each error or invalid entry to a file named `errors.log` which includes the offending line, the error message, and a timestamp

Submission guidelines:

Submit a zip folder which includes

- All `.java` files
- A sample `catalog.txt` with test data
- `errors.log` (if generated)
- A `README.txt` with overview of each feature, and assumptions made

Marking Scheme

Criteria	Marks
Correct reading & interpretation of command line arguments	5
File and directory creation if not present	5
Search	10
Insertion of new book	10
Proper implementation of exception hierarchy	30
Stats & guaranteed final output	20
Logging & timestamp	10
Code structure & quality	10

