Campus Course & Records Manager (CCRM)

1. Project Overview

CCRM is a console-based Java SE application to manage campus students, courses, grades, CCRM is a **console-based Java application** designed to help educational institutes efficiently manage their courses, students, and academic records. The system allows administrators to:

- **Students:** Create, update, and manage student profiles; enroll or unenroll students in courses.
- **Courses:** Create, update, list, and search courses; assign instructors to courses.
- **Grades & Transcripts:** Record student marks, compute GPA, and print detailed transcripts.
- **File Utilities:** Import and export data using CSV/JSON-like plain files; archive and backup course data.

Technical Details:

- Built using **Java SE** and run locally.
- Structured with clear packages to maintain modularity and readability.
- Demonstrates **core Object-Oriented Programming concepts**:
    - **Encapsulation, Inheritance, Abstraction, Polymorphism**
- Includes **robust exception handling** to manage errors gracefully.
- Uses modern Java features:
    - **NIO.2 for file operations**
    - **Streams API for data processing**
    - **Date/Time API for handling academic records**
- Utilizes **interfaces, abstract classes, nested classes, enums, lambdas, recursion**.
- Implements **design patterns**: Singleton (for central services) and Builder (for complex object creation).

Purpose

- For **developing standalone applications**, not web servers or enterprise systems.
- Examples: desktop apps, console apps, small utilities, and local tools.

Components

Java SE includes:

- **Java Development Kit (JDK):** For writing, compiling, and running Java programs.
- **Java Runtime Environment (JRE):** For running Java programs.
- **Core Libraries:** java.lang, java.util, java.io, java.nio, java.time, etc.
- **APIs:** Collections, Streams, Date/Time, Networking, File I/O, etc.

Runnable main class: edu.ccrm.cli.MainCLI.java

JDK Version: 17

How to run:
javac -d out src/edu/ccrm/**/*.java
java -cp out edu.ccrm.cli.MainCLI

2. Evolution of Java

- 1995: Java 1.0 – Platform-independent programming
- 1998: Java 2 – J2SE, J2EE, J2ME
- 2004: Java 5 – Generics, annotations
- 2014: Java 8 – Lambda expressions, streams
- 2021: Java 17 – LTS, modern features

3. Java ME vs SE vs EE

Feature	Java ME	Java SE	Java EE / Jakarta EE
Target Platform	Mobile/Embedded	Desktop/Standard	Enterprise / Web
APIs	Lightweight	Standard APIs	Enterprise APIs
Example Apps	Mobile apps	Desktop tools	Web servers, APIs

---

4. JDK / JRE / JVM

- JVM: Executes Java bytecode
- JRE: JVM + standard libraries, runs programs
- JDK: JRE + compiler, for developing programs

5. Windows Installation & Eclipse Setup

JDK Installation:
1. Download and install JDK from Oracle/AdoptOpenJDK
2. Set JAVA_HOME environment variable
3. Verify installation:
java -version
javac -version
Screenshot: screenshots/jdk_install.png

Eclipse Setup:
1. Open Eclipse → Create New Java Project
2. Import src folder
3. Set output folder to out/
4. Run MainCLI.java
Screenshot: screenshots/eclipse_setup.png

6. Mapping Table: Syllabus → Code

Syllabus Topic	File / Class / Method
OOP Concepts	Course.java, Enrollment.java
File I/O / CSV	io/CSVReader.java, io/CSVWriter.java
Exception Handling	exceptions/*.java
Collections	service/StudentService.java
CLI Interface	cli/MainCLI.java

7. Enabling Assertions

Run with -ea flag:
java -ea -cp out edu.ccrm.cli.MainCLI
Sample assertion in code:
assert studentId != null : "Student ID cannot be null"