# Campus Course & Records Manager (CCRM)

## Project Overview

Campus Course & Records Manager (CCRM) is a console-based Java SE application that allows institutes to manage:  
- Students (add, list, deactivate, transcript)  
- Courses (add, list, filter/search)  
- Enrollments and Grades (with GPA calculation)  
- File operations (import/export CSV, backup with recursion)  
- Menu-driven CLI workflow

## How to Run

1. Save the source code as CCRMApp.java.  
2. Compile:  
 javac CCRMApp.java  
3. Run:  
 java CCRMApp

## Evolution of Java

- 1995 – Java 1.0 released by Sun Microsystems  
- 1998 – Java 2 (split into J2SE, J2EE, J2ME)  
- 2004 – Java 5 (Generics, Enums, Annotations, Enhanced for-loop)  
- 2011 – Java 7 (NIO.2, try-with-resources)  
- 2014 – Java 8 (Lambdas, Streams, Date/Time API)  
- 2017–2023 – Java 9–21 (Modules, var, records, switch expressions, pattern matching)

## Java Editions – ME vs SE vs EE

|  |  |  |  |
| --- | --- | --- | --- |
| Feature | Java ME | Java SE | Java EE |
| Scope | Embedded/mobile devices | Desktop & console apps | Enterprise/web/distributed apps |
| APIs | Limited core APIs | Full core APIs (IO, Streams) | Adds Servlets, JSP, JPA, EJB |
| Target Device | IoT, mobiles | PC, desktop, servers | Large-scale enterprise servers |
| Example Use | Smartcards, IoT apps | CLI apps, Swing apps | Banking, e-commerce, portals |

## JDK, JRE, JVM

- JDK (Java Development Kit): Compiler (javac) + JRE + tools for development.  
- JRE (Java Runtime Environment): JVM + core libraries to run Java programs.  
- JVM (Java Virtual Machine): Executes bytecode, provides platform independence.  
Flow: .java (source) → javac → .class (bytecode) → JVM → Execution

## Install & Configure Java on Windows

1. Download latest JDK from Oracle.  
2. Install it → set installation path.  
3. Configure environment variables:  
 - Add JAVA\_HOME pointing to JDK folder.  
 - Add %JAVA\_HOME%\bin to PATH.  
4. Verify installation:  
 java -version  
 javac -version

## Eclipse IDE Setup

1. Open Eclipse → File > New > Java Project.  
2. Project name: CCRM.  
3. Create a new class CCRMApp.  
4. Paste the code and save.  
5. Run → Run As > Java Application.

## Mapping Table (Syllabus Topic → Code Location)

|  |  |
| --- | --- |
| Topic | Implemented In |
| Encapsulation | Student (private fields + getters/setters) |
| Inheritance | Person → Student, Instructor |
| Abstraction | Abstract class Person |
| Polymorphism | profile() method overridden in subclasses |
| Enums | Semester, Grade |
| Singleton | AppConfig |
| Builder Pattern | Course.Builder |
| Nested Class | Course.CourseCode |
| Interfaces | StudentService, CourseService |
| Lambdas & Streams | Sorting/filtering in services & reports |
| Anonymous Inner Class | SimpleFileVisitor in BackupService |
| Custom Exceptions | DuplicateEnrollmentException, MaxCreditLimitExceededException |
| NIO.2 | Files.walkFileTree, Files.copy, Files.write |
| Date/Time API | LocalDate, Instant |
| Control Structures | switch, while, break, continue |

## Usage Example

CCRM started. Data folder: C:\Users\YourName\ccrm\_data  
  
1) Students 2) Courses 3) Enrollment 0) Exit  
1  
Add student? (y/n): y  
RegNo: 101  
Name: Alice  
Email: alice@mail.com  
Student: Alice (101) - ACTIVE  
  
1) Students 2) Courses 3) Enrollment 0) Exit  
3  
Student regNo: 101  
Course code: CS101  
Enrolled.  
Marks? (y/n): y  
Enter marks: 85  
Transcript for Alice  
CS101 - Data Structures -> A  
GPA: 9.00

## Assertions

Assertions are used in constructors to ensure non-null IDs.  
Enable with:  
 java -ea CCRMApp

## Deliverables

- Source code: CCRMApp.java  
- README.md (this file)  
- Screenshots folder:  
 \* Java installation verification  
 \* Eclipse project setup & run  
 \* Program execution screenshots  
 \* Export/Backup folder structure  
- Optional demo video (2–5 mins walkthrough)

## Academic Integrity

This project is for educational purposes only. Submission must be your own work. Cite any references in acknowledgements.