

Big Java Question – University Management System

Objective:

Design a Java-based University Management System that manages students, professors, courses, grades, salaries, attendance, and more, using core Java concepts and good programming practices.

Requirements and Features

1. Base Class: Person (Abstract)

Fields: id, name, gender, address

Constructor and displayDetails() method

Abstract void showRole();

Access Modifiers: Use private and protected for encapsulation

✓ 2. Inherited Classes:

Student extends Person

Fields: ArrayList<Course> courses, HashMap<Course, Integer> grades, int attendedClasses, int totalClasses

Methods: calculateGPA(), getAttendancePercentage(), checkScholarshipEligibility()

Professor extends Person implements Payable

Fields: ArrayList<Course> teachingCourses, double baseSalary, int extraClasses

Method: calculateSalary(), assignCourse(Course c)

✓ 3. Interface: Payable

Double calculateSalary();

✓ 4. Class: Course

Fields: String courseId, String courseName, int credits, Professor professor

Constructor and getter/setters

✓ 5. Class: University

Fields:

ArrayList<Student> students

ArrayList<Professor> professors

ArrayList<Course> courses

Features:

Add / remove / search student, professor, course

Assign students to courses

Assign professors to courses

Sort students by GPA

Filter students by attendance %

 Math Operations

GPA Calculation:

$$\text{GPA} = (\text{Sum of (credits * grade)}) / \text{Total Credits}$$

Attendance Percentage:

$$\text{Attendance\%} = (\text{attended} / \text{total}) * 100$$

Professor Bonus:

$\text{Salary} = \text{baseSalary} + (500 * \text{extraClasses})$

File Handling

Save and load student/professor/course data from .txt or .csv

Print summary reports for:

Top 3 students by GPA

Attendance defaulters

Salary Report for Professors

Data Structures to Use

ArrayList – to manage lists of entities

HashMap<Student, Integer> – for storing grades

HashMap<String, Student> – fast lookup by ID

Queue<Student> – for simulating course registrations

Collections.sort() – to sort students by GPA

! Exception Handling

Handle:

Division by zero (in GPA)

Invalid input (nulls, empty)

File not found errors

Duplicates on registration

CLI Menu

Use Java Scanner for a menu system like:

Welcome to University System

1. Add Student
 2. Add Professor
 3. Add Course
 4. Register Student to Course
 5. Calculate GPA
 6. Generate Reports
 7. Exit
-

BONUS

Add login roles (Admin / Professor / Student)

Passwords stored in `HashMap<String, String>`

Role-based access to functionalities