

# **United International University**

## **CSE 1116 - Mid Exam - Set A**

**Write your name and ID on the question paper. Marks are on the right. Return question.**

## **University Management System**

A university wants to build a simple object-oriented system to manage its people and courses.  
You are asked to design classes following the principles of OOP.

### **Classes and Requirements**

#### **1. Person (Base Class) [3]**

- **Private fields:** String name, int age.
- Getters and setters for both private fields. For age, check if  $(0 < \text{age} < 120)$ .
- String getDetails() - returns a string in the format "Person: name, age".
- A constructor that takes 2 parameters name and age

#### **2. Student (Subclass of Person) [2]**

- **Field:** String major.
- A constructor that takes 3 parameters name, age, major
- Override the getDetails() method that returns a string in the format "Student: name, age, major".

#### **3. Professor (Subclass of Person) [6]**

- **Field:** String department, Students [ ] thesisStudents (size = 5)
- A constructor that takes 3 parameters name, age, department
- Override the getDetails() method that returns a string in the format "Professor: name, age, department".
- **Method:** addThesisStudent(Student s) - add the Student s object in the thesisStudents array
- **Method:** thesisStudentInfo() - Print details of all the thesis students

#### **4. Course [3]**

- **Private fields:** String code, String title.
- A constructor which accepts (code, title)
- Another constructor which accepts (code) with "TBD" as default title.
- String courseInfo() - returns a string in the format "Course: code - title".

## 5. Enrollment [4]

- **Fields:** Student student, Course course, Professor professor, double grade (default -1 for not graded).
- A constructor which takes (Student, Course, Professor)
- **Method:** void assignGrade(double g) (only valid if  $0 \leq g \leq 100$ , otherwise ignore).
- **Method:** String getReport() - returns a string in the format - "studentName enrolled in courseTitle under professorName: grade" (if grade not assigned, show "Pending").

## Main Tasks [7]

Create a **Main** class and do the following tasks in the **main** method:

1. Create at least **4 Students** and **2 Professors**, store them in a Person[], and print their details polymorphically.
2. Assign **2 Students** to each **Professor** as thesis students in any order.
3. Create **2 Courses** using different constructors and print their info.
4. Enroll the **students and professor** in different courses (at least **2 enrollments**).
5. Assign **grades** to some enrollments (show valid and invalid cases).
6. Print all **enrollment** reports.
7. Print the **information** of the thesis students of **both the professors**.