**OBJECT-ORIENTED PROGRAMMING**

|  |  |
| --- | --- |
| Lab 12 | |
| **Topic** | Multilevel Inheritance |
| **Objective** | To text the understanding ofof inheritance and its types. |

**Case Study: ABC Online Learning Platform**

*Background:*

ABC Online Learning Platform is a modern and flexible e-learning system designed to facilitate education through a digital medium. It caters to a diverse range of users, including students and teachers, offering a variety of courses. The platform employs a sophisticated class hierarchy using multiple multilevel and hybrid inheritance to efficiently manage user roles and course interactions.

*Classes in the System:*

1. **User Class:**

The **User** class is the foundation of the hierarchy, representing a generic user on the platform.

* + *Attributes:*
    - **username** (string): holds the unique username of the user.
    - **userID** (int): a system-generated unique identifier for each user.
  + *Methods:*
    - **displayUserInfo()**: displays basic user information.

1. **Student Class:**

The **Student** class is derived publicly from the **User** class and represents individuals enrolled in courses.

* + *Additional Attributes:*
    - **coursesEnrolled** (\*string): stores the list of courses the student is currently enrolled in.
  + *Additional Methods:*
    - **enrollInCourse(const string& course)**: allows a student to enroll in a course.
    - **displayStudentInfo()**: extends **displayUserInfo()** to include enrolled courses.

1. **Teacher Class:**

The **Teacher** class is also derived publicly from the **User** class and represents instructors offering courses.

* + *Additional Attributes:*
    - **coursesTeaching** (\*string): keeps track of the courses the teacher is currently teaching.
  + *Additional Methods:*
    - **assignCourse(const string& course)**: enables a teacher to assign a course to themselves.
    - **displayTeacherInfo()**: extends **displayUserInfo()** to include taught courses.

1. **Course Class:**

The **Course** class is derived publicly from both **Student** and **Teacher**, demonstrating hybrid inheritance.

* + *Additional Attributes:*
    - **courseName** (string): denotes the name of the course.
  + *Additional Methods:*
    - **displayCourseInfo()**: displays comprehensive information about the course, including teacher and student details.

Usage Scenario:

Imagine a scenario where a new user, John, joins the ABC Online Learning Platform:

// Create instances of Student, Teacher, and Course

Student student1("JohnDoe");

Teacher teacher1("DrSmith");

Course course1("IntroductionToProgramming");

// John enrolls in a course

student1.enrollInCourse("IntroductionToProgramming");

// Dr. Smith assigns a course to himself

teacher1.assignCourse("IntroductionToProgramming");

// Display information

cout << "Student Information:\n";

student1.displayStudentInfo();

cout << "\nTeacher Information:\n";

teacher1.displayTeacherInfo();

cout << "\nCourse Information:\n";

course1.displayCourseInfo();

*Discussion:*

The ABC Online Learning Platform's class hierarchy demonstrates the use of multiple multilevel and hybrid inheritance effectively. The **User** class provides a common base for both students and teachers, allowing for shared functionalities. The **Course** class inherits publicly from both **Student** and **Teacher**, resulting in a hybrid inheritance scenario.

This design promotes code reusability and a clear representation of the relationships between different entities in the online learning system. The system allows students to enroll in courses, teachers to assign courses to themselves, and courses to display comprehensive information about enrolled students and assigned teachers.

The use of multiple multilevel and hybrid inheritance ensures a flexible and extensible system that can accommodate future changes and additions to the platform's features.