Assignment: Object-Oriented Programming (OOP) in C++

Assignment Title: Smart University Management System

Assignment Description

You are required to design and implement a C++ application titled **Smart University Management System**. This project will test your knowledge of Object-Oriented Programming principles and recursion. You must apply class inheritance, polymorphism, virtual functions, operator overloading, friend and static members, and templates to complete the full system.

Assignment Questions

1. **User Class Hierarchy**
   - Create a base class `User` with the attributes: `name`, `id`, and a pure virtual function `displayInfo()`.
   - Create derived classes `Student` and `Professor` with their specific attributes.
   - Override the `displayInfo()` function in both derived classes.
2. **Friend Function Usage**
   - Implement a friend function `assignCourse(Student&, Course&)` that assigns a course to a student and accesses private members.
3. **Static Members and Functions**
   - Use static variables to keep count of total users and implement `getTotalUsers()`.
4. **Template-Based GPA Calculator**
   - Create a template class `Calculator<T>` that overloads arithmetic operators and includes a recursive GPA calculation method.
5. **Course Class with Operator Overloading**
   - Define a `Course` class and overload the == and << operators.

6. **Recursive Functions**
   - Write a recursive function `sumOfDigits(int)` to calculate the digit sum of a number.
   - Write a recursive function `power(float base, int exp)` to calculate exponential values using divide and conquer strategy.
7. **Virtual and Abstract Functions**
   - Declare a pure virtual function `calculatePerformance()` in the `User` class.
   - Implement it in `Student` and `Professor` classes differently.
8. **Virtual Base Class (Advanced)**
   - Create a base class `Person` and inherit it virtually into the `User` class to prevent ambiguity in multiple inheritance.

Bonus Task (Optional)

Develop a professor ranking system using recursive sorting and template functions for numeric score types.

Sample Console Output

```
Welcome to Smart University Management System
-----------------------------------------------
1. Create Student
2. Create Professor
3. Assign Course
4. Calculate GPA
5. Show User Info
...
```

Submission Guidelines

- Submit only one `.cpp` file.
- Comment your code meaningfully.
- Begin with a header section including your name, roll number, and a summary of your implementation.

Note: Late submissions may not be accepted without prior notice.

End of Assignment