***Question#1:-***

Design and implement a program that shows the relationship between person, student and professor. Your person class must contain two pure virtual functions named getData() of type void and isOutstanding() of type bool and as well getName() and putName() that will read and print the person name. Class student must consist of function name getData (), which reads the GPA of specific person and isOutstanding() function which returns true if the person GPA is greater than 3.5 else should return false. Class professor should take the respective persons publications in getData() and will return true in isOutstanding() if publications are greater than 100 else will return false . Your main function should ask the user either you want to insert the data in professor or student until and unless user so no to add more data.

***Question#2:-***

Imagine you're developing software for a newly opened gourmet restaurant called "Savory Delights." As part of the restaurant management system, you're tasked with implementing a module for handling menu items. The head chef at Savory Delights has provided detailed instructions for preparing and serving various types of dishes, ranging from appetizers to main courses and desserts. To facilitate this, you decide to create an abstract class called `MenuItem`, which defines pure virtual functions for `prepare()` and `serve()`. Additionally, you plan to create derived classes such as `Appetizer`, `MainCourse`, and `Dessert`, each tailored to the specific requirements of its category. For example, an `Appetizer` might involve assembling delicate hors d'oeuvres with precision, a `MainCourse` could require complex cooking techniques and plating arrangements to showcase the chef's culinary expertise, and a `Dessert` might involve intricate decoration and garnishing to delight the diner's senses. To ensure the software meets the chef's exacting standards, you'll need to test it rigorously by creating instances of each menu item type and verifying that the `prepare()` and `serve()` methods are executed flawlessly, replicating the meticulous attention to detail required in a high-end restaurant kitchen. This scenario-based approach not only demonstrates your ability to translate culinary concepts into software design but also ensures that the restaurant management system enhances the dining experience at Savory Delights by delivering impeccably prepared and presented dishes to discerning customers.