Let me Explain the code briefly:

1. The code contains three classes, all defined in the ClassLibrary1 namespace: student, teacher, and subject. These classes are used to represent students, teachers, and subjects.

2. The student class:

- It has three public fields: name, clas, and sec, representing the name, class, and section of a student, respectively.
- It has two virtual methods: gettingdetails() and printthedetails(). A virtual method can be overridden in derived classes (teacher and subject classes).
- gettingdetails() method prompts the user to enter student details such as name, class, and section, and reads the input from the console.
- printthedetails() method prints the student details to the console.

3. The teacher class:

- It inherits from the student class (using the : student syntax after the class name).
- It adds an additional field subject to represent the subject taught by the teacher.
- It overrides the gettingdetails() and printthedetails() methods to gather and print teacher-specific details in addition to the student details.

4. The subject class:

- It also inherits from the student class.
- It adds a field subcode to represent the subject code.
- It overrides the gettingdetails() and printthedetails() methods to gather and print subject-specific details in addition to the student details.

In the Program.cs file:

5. The Main method is the entry point of the program.

- An instance of student class is created and its gettingdetails() and printthedetails() methods are called.
- An instance of teacher class is created and its gettingdetails() and printthedetails() methods are called.
- An instance of subject class is created and its gettingdetails() and printthedetails() methods are called.

Overall, the code showcases basic inheritance and polymorphism. The teacher and subject classes inherit the properties and methods of the student class, and they also override the base class methods to provide their specific details. This allows you to use a single variable of type student to hold instances of both teacher and subject classes due to the polymorphic behavior. However, note that in the current code structure, the student class is not meant to be instantiated directly, as it is a base class for other specialized classes.

GIT Link: https://github.com/sundar2568223/studentProject-oops.git