Solution Review: Implement the Complete Student Class

This review provides a detailed analysis to solve the 'Implement the Complete Student Class' challenge.

Solution

```
class Student // Student Class
{
  // Fields
  private string _name;
  private string _rollNumber;

  // Properties
  public string Name
  {
    set
    {
      this._name = value;
    }
    get
    {
      return this._name;
    }
  }

  public string RollNumber
  {
    set
    {
      this._rollNumber = value;
    }
    get
    {
      return this._rollNumber;
    }
  }

}
```

```
class Program
{
  public static void Main()
  {

     Student student = new Student();
     student.Name = "John";
     student.RollNumber = "20";
     System.Console.WriteLine(student.Name + " " + student.RollNumber);
    // In an Encapsulated implementation the following should return an error
    // student._name = "John";
    // student._rollNumber = "20";
  }
}
```

Explanation

- **Line 4-5:** We have implemented the `Student` class which has the **private** fields `_name` and `_rollNumber.`
- **Line 8-18:** Implemented `Name`, a **public** *property* which returns the name of a student.
- **Line 20-29:** Implemented `RollNumber`, a **public** *property* which returns the roll number of a student.
- **Line 36-46:** In the `Main()` method the fields are accessed using the implemented properties.

The essence of this example is to secure the class fields by declaring them as **private**. In order to access the fields, **public** properties or getters/setters should be implemented.

In the next section, you’ll learn about a very important pillar of object-oriented programming paradigm: Inheritance.