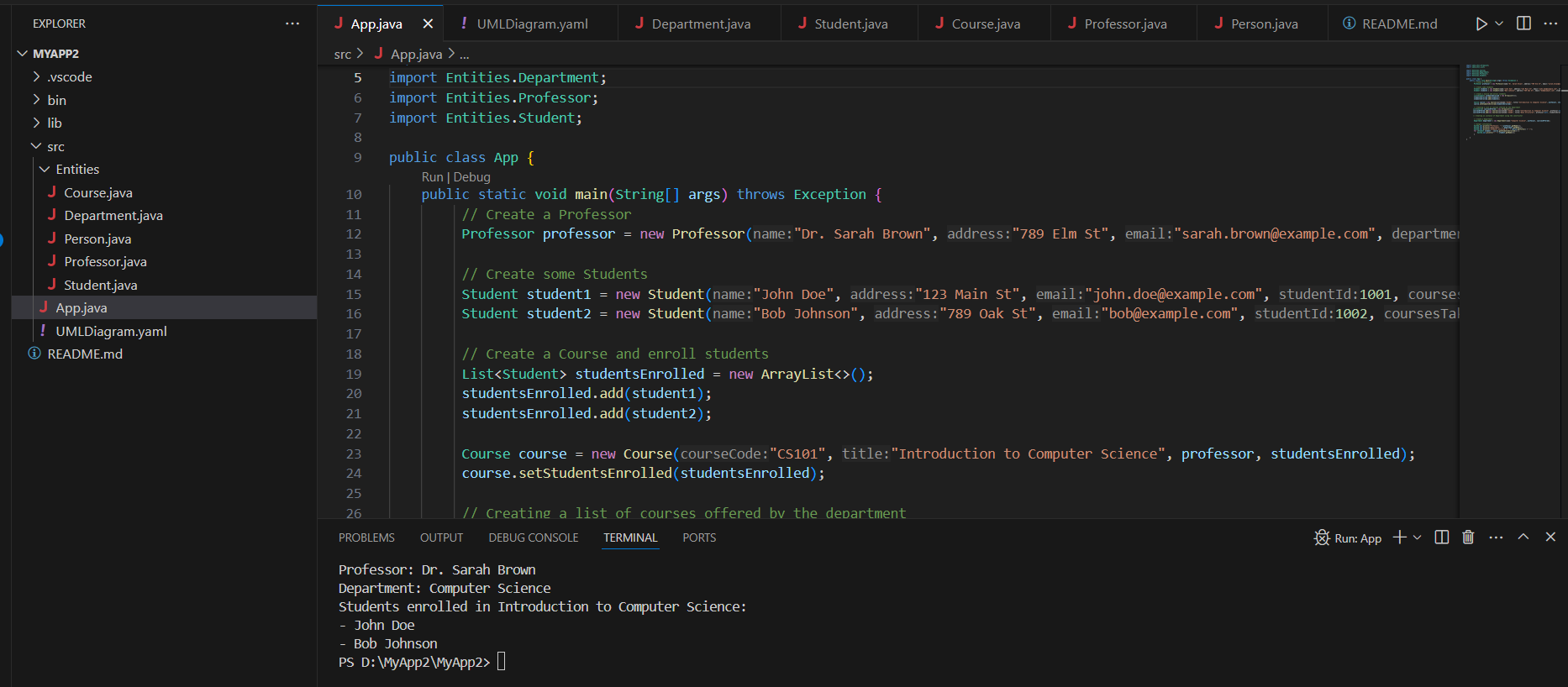
1. **Write, compile and run programs in Java**



Refer to the Java Program shared with this document

1. **Describe object hierarchy structure and how to design such a hierarchy of related classes.**

**Answer:**

**What is Object hierarchy structure?**

Object hierarchy structure refers to the arrangement of classes and their relationships based on inheritance.

**How to design a hierarchy of related classes?**

Step 1: Identify Classes and Their Relationships

* **Identify Entities:** Determine the main entities or objects in your system. For instance, in a university system, entities might include Person, Student, Professor, Course, Department, etc.
* **Understand Relationships:** Define how these entities relate to each other. For example, a Student enrolls in many Courses, a Course is taught by a Professor, a Department offers many Courses, etc.

Step 2: Define Base and Derived Classes

* **Base Classes:** These are more general entities that can serve as the foundation for other, more specific classes. For example, Person as an entity.
* **Derived Classes:** These classes inherit attributes and behaviors from their base classes and may add their own specific attributes and methods. For example, Student as an entity inherits attributes from Person as an entity.

Step 3: Implement Interfaces for Common Behaviors

* **Interfaces:** Define interfaces to specify common behaviors that classes can implement. For example, attendance can be calculated for both Student and Professor.

Step 4: Use Composition for "Has-a" Relationships

* **Composition:** Use composition to model "has-a" relationships, where one class contains another as a member. For example, a Course has Students.

1. **Describe the concept of object polymorphism in theory and demonstrate this concept in practice.**

**Answer:**

**Object polymorphism** refers to the ability of different objects to be treated as instances of the same class (or superclass) but behave differently based on their actual implementation. It allows objects of different classes to be processed uniformly if they belong to a common superclass or implement a common interface.

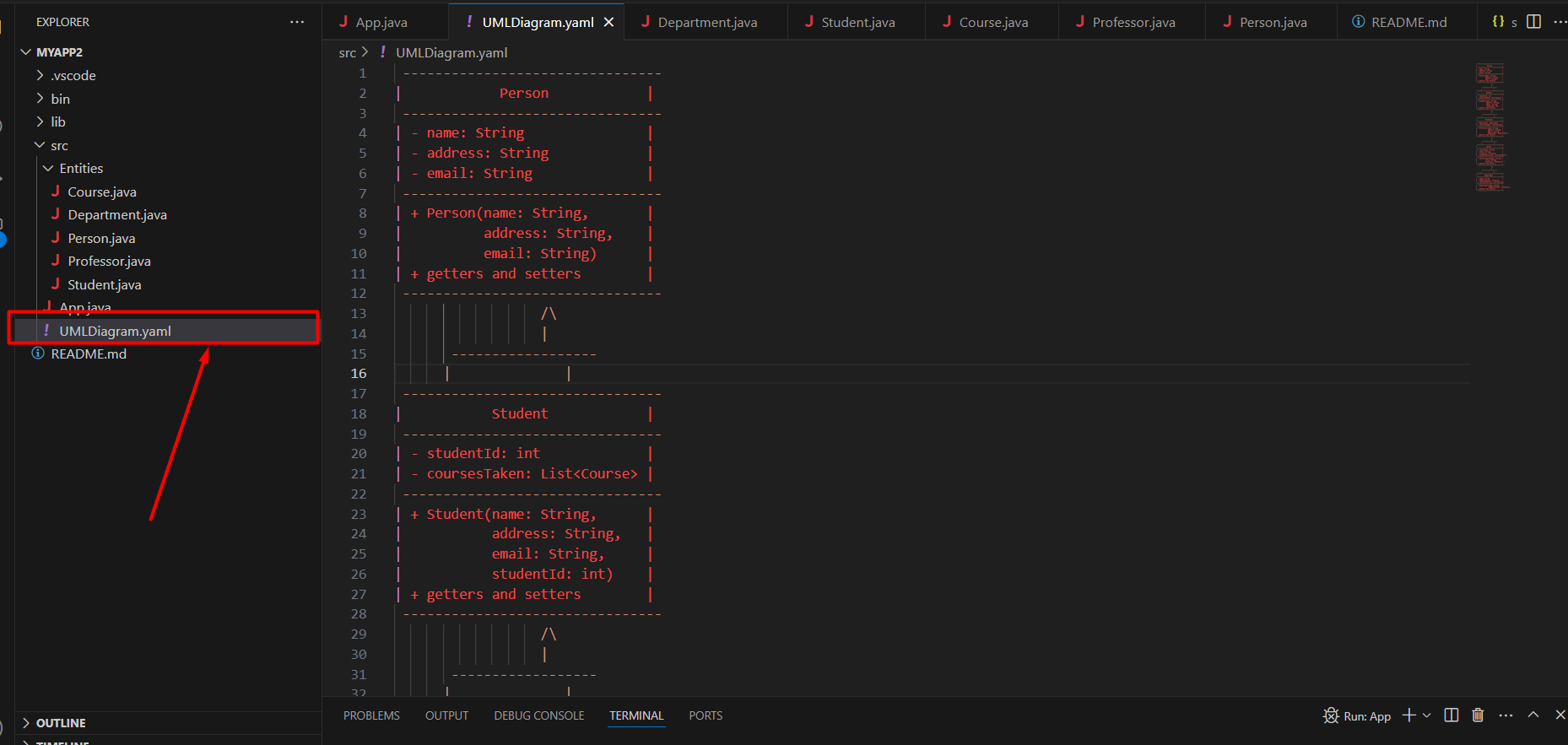
Polymorphism relies on inheritance, where subclasses inherit behaviors and properties from their superclass.

In practice, this has been implemented in the Java Program shared with this document.

1. **Make use of Unified Modelling Language (UML) diagrams and test code. Use common design patterns in your programs**

**Answer:**

**Unified Modeling Language (UML**) diagrams visually represent the design of the object hierarchy in a standardized format.



Refer to the **UMLDiagram.yaml** file which part of the Java Program.

### Explanation of the Test Code

**Creating Instances:** The code creates instances of Professor, Student, Course, and Department classes, initializing them with sample data.

**Relationships:** It demonstrates how a Professor is associated with Course and Department, and how Students are enrolled in a Course.

**Output:** Finally, the code outputs information about the professor and students enrolled in a course, demonstrating the usage of getters to retrieve data.