1. **Shape Abstract Class:**
    - Fields: **l** (length) and **h** (height).
    - **input** method to get user input for length and height.
    - Abstract method **computeArea()**.
2. **Triangle Class (extends Shape):**
    - Implements the **computeArea** method to calculate the area of a triangle using the formula **0.5 * length * height**.
3. **Rectangle Class (extends Shape):**
    - Implements the **computeArea** method to calculate the area of a rectangle using the formula **length * height**.
4. **DynamicBinding Class (main class):**
    - Creates instances of **Triangle** and **Rectangle**.
    - Declares a reference variable **s** of type **Shape**.
    - Assigns the reference of the **Triangle** object to **s**, demonstrating dynamic binding.
    - Calls **input** and **computeArea** methods through the **s** reference, which is dynamically bound to the appropriate method based on the actual object type.

Explanation of Operations:

- The program creates instances of **Triangle** and **Rectangle**.
- A reference variable **s** of type **Shape** is declared.
- The **s** reference is assigned the reference of the **Triangle** object, demonstrating dynamic binding.
- The **input** and **computeArea** methods are called through the **s** reference, and the appropriate methods in the **Triangle** class are invoked.
- The **s** reference is then assigned the reference of the **Rectangle** object, and the **input** and **computeArea** methods are called, invoking the appropriate methods in the **Rectangle** class.

In summary, this program showcases dynamic binding through polymorphism, where the behavior of the **computeArea** method is determined at runtime based on the actual type of the object referenced by the **s** variable