# Flutter Diploma

Become a Flutter Developer with just one course

1

Design a class named **GeometricObject**. This class contains:

- **color:** color name of the object (default value: “white”)
- **filled:** indicates if the object is filled with color or not (default value: false)
- **no argument constructor:** create a GeometricObject with default values for arguments
- **argument constructor:** create a GeometricObject with specified **color** and **filled** values
- **toString:** method returns a string description of the object

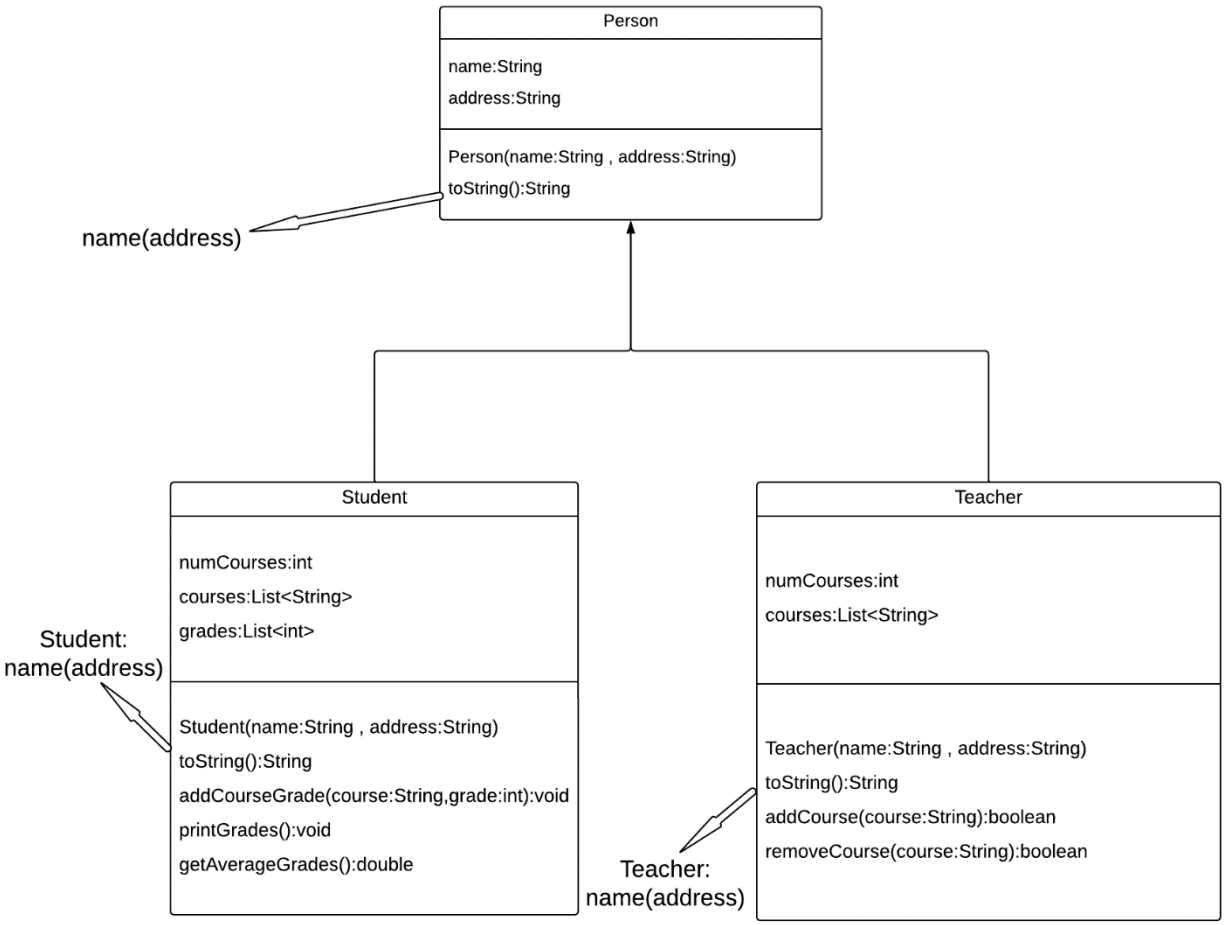
Design a class named **Triangle** that inherits (extends) from GeometricObject. This class contains:

- **side1:** first side of the triangle with default value 1.0
- **side2:** second side of the triangle with default value 1.0
- **side3:** third side of the triangle with default value 1.0
- **no argument constructor:** creates a triangle with default values
- **constructor:** creates a triangle with specified side1, side2, and side3 values
- **getArea():** a method that returns the area of the triangle
- **getPerimeter():** a method that returns the perimeter of the triangle
- **toString():** method returns a string description of the triangle object like that : “Triangle: side1 = $side1, side2 = $side2, side3 = $side3”

Design a class named Rectangle that inherits (extends) from GeometricObject. This class contains:

- **height:** the height of the rectangle with a default value 1.0
- **width:** the width of the rectangle with a default value 1.0
- **no argument constructor:** creates a rectangle with default values
- **constructor:** creates a rectangle with specified width, height, color, and filled values
- **getArea():** a method that returns the area of the rectangle
- **getPerimeter():** a method that returns the perimeter of the rectangle
- **toString():** method returns a string description of the rectangle object like that “Rectangle: height = $height, width = $width”

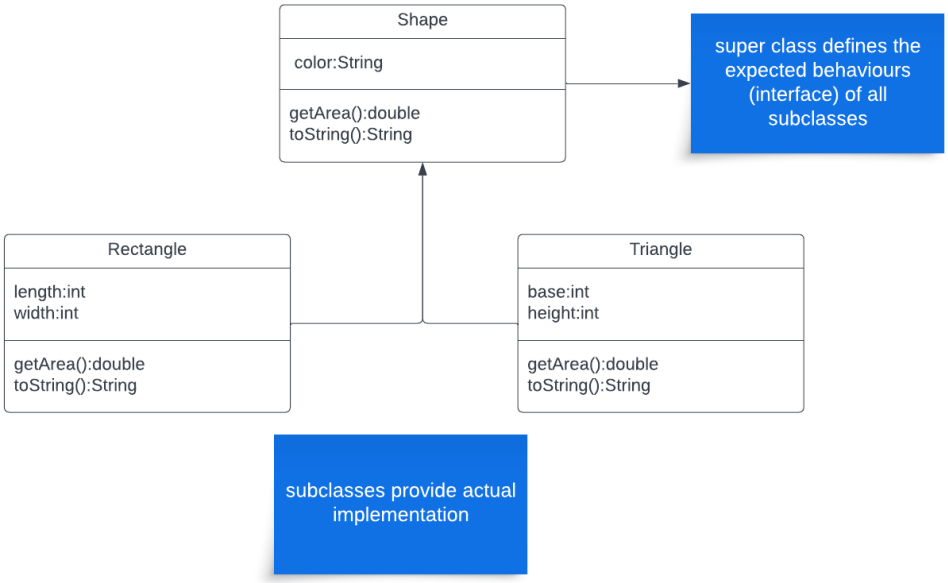
2




- In the Teacher class method addCourse returns false if a course already exists
- In the Teacher class method removeCourse returns false if a course doesn't exist

3

**(Polymorphism and abstraction)**




Suppose that our program uses many kinds of shapes, such as triangles, rectangles, and so on. We should design a superclass called **Shape**, which is an interface (abstract class with abstract methods) for all the shapes. For example, we would like all the shapes to have a method called getArea(), which returns the area of that shape. Implement getArea in subclasses.