SCIT

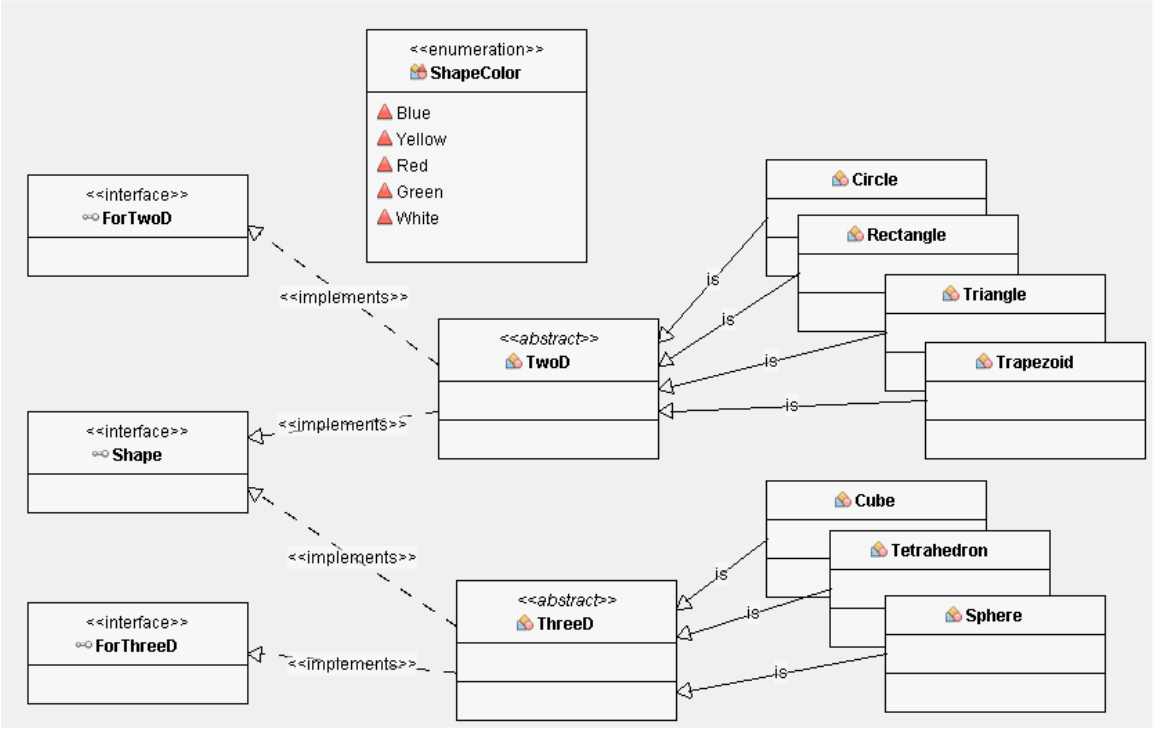
School of Computing and Information Technology
Faculty of Engineering & Information Sciences

CSIT121
Object Oriented Design and Programming
Assignment 2
File name: YourName_A2.java

Objectives:

Practice java programming with inheritance and polymorphism.

Task (5 marks)



A quick look to the UML diagram:

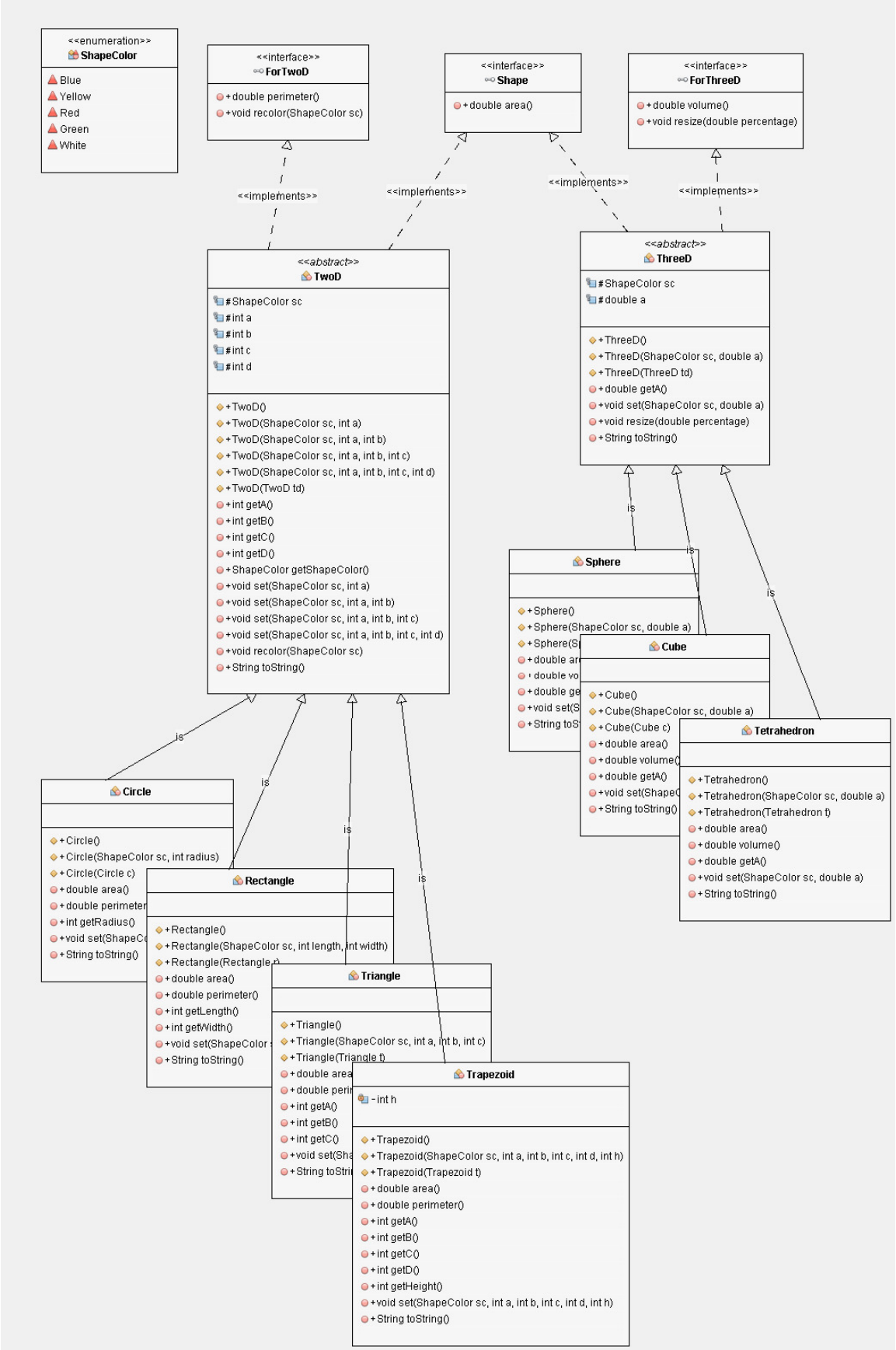
- Three interfaces ForTwoD, Shape and ForThreeD
- TwoD class implements ForTwoD and Shape interfaces
- ThreeD class implements ForThreeD and Shape interfaces
- Four subclasses to TwoD (Circle, Rectangle, Triangle and Trapezoid)
- Three subclasses to ThreeD (Cube, Tetrahedron, Sphere)
-

Implement the Shape hierarchy as shown in the above diagram

- Basically, each `two-dimensional shape` needs to compute its area and its perimeter.

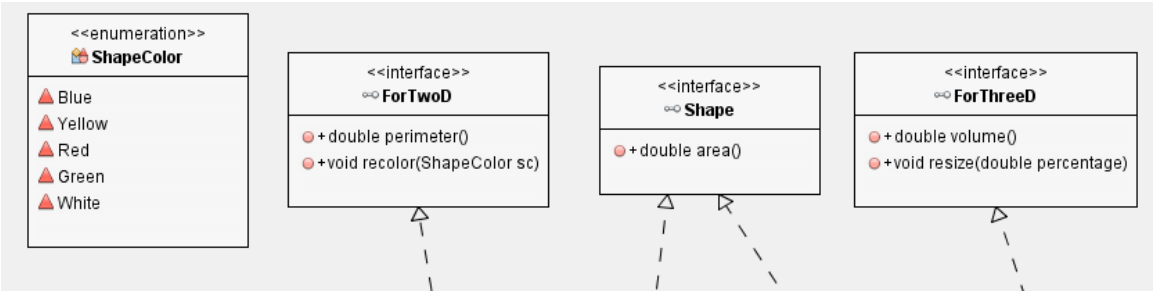
- Each `three-dimensional shape` needs to compute its `area` (also known as surface area) and its volume.
- Try to surf the internet to look for some formulas, for example, to compute the areas, the surface areas, and the volumes … I did that too ☺

A more detailed UML diagram is shown as follow: (Note that the #'s before the instance variables and the methods' names mean "protected")


<<enumeration>>
ShapeColor
Blue
Yellow
Red
Green
White
<<interface>>
ForTwoD
+double perimeter()
+void recolor(ShapeColor sc)
<<interface>>
Shape
+double area()
<<interface>>
ForThreeD
+double volume()
+void resize(double percentage)
<<implements>>
<<implements>>
<<implements>>
<<implements>>
<<abstract>>
TwoD
#ShapeColor sc
#int a
#int b
#int c
#int d
+TwoD()
+TwoD(ShapeColor sc, int a)
+TwoD(ShapeColor sc, int a, int b)
+TwoD(ShapeColor sc, int a, int b, int c)
+TwoD(ShapeColor sc, int a, int b, int c, int d)
+TwoD(TwoD td)
+int getA()
+int getB()
+int getC()
+int getD()
+ShapeColor getShapeColor()
+void set(ShapeColor sc, int a)
+void set(ShapeColor sc, int a, int b)
+void set(ShapeColor sc, int a, int b, int c)
+void set(ShapeColor sc, int a, int b, int c, int d)
+void recolor(ShapeColor sc)
+String toString()
<<abstract>>
ThreeD
#ShapeColor sc
#double a
+ThreeD()
+ThreeD(ShapeColor sc, double a)
+ThreeD(ThreeD td)
+double getA()
+void set(ShapeColor sc, double a)
+void resize(double percentage)
+String toString()
is
is
is
is
is
is
is
is
Sphere
+Sphere()
+Sphere(ShapeColor sc, double a)
Cube
+Cube()
+Cube(ShapeColor sc, double a)
+Cube(Cube c)
+double area()
+double volume()
+double getA()
+void set(ShapeC
+String toString()
Tetrahedron
+Tetrahedron()
+Tetrahedron(ShapeColor sc, double a)
+Tetrahedron(Tetrahedron t)
+double area()
+double volume()
+double getA()
+void set(ShapeColor sc, double a)
+String toString()
Circle
+Circle()
+Circle(ShapeColor sc, int radius)
+Circle(Circle c)
+double area()
+double perimeter
+int getRadius()
+void set(ShapeC
+String toString()
Rectangle
+Rectangle()
+Rectangle(ShapeColor sc, int length, int width)
+Rectangle(Rectangle r)
+double area()
+double perimeter()
+int getLength()
+int getWidth()
+void set(ShapeColor
+String toString()
Triangle
+Triangle()
+Triangle(ShapeColor sc, int a, int b, int c)
+Triangle(Triangle t)
+double area
+double peri
+int getA()
+int getB()
+int getC()
+void set(Sha
+String toStri
Trapezoid
-int h
+Trapezoid()
+Trapezoid(ShapeColor sc, int a, int b, int c, int d, int h)
+Trapezoid(Trapezoid t)
+double area()
+double perimeter()
+int getA()
+int getB()
+int getC()
+int getD()
+int getHeight()
+void set(ShapeColor sc, int a, int b, int c, int d, int h)
+String toString()

Wow … so difficult to see☺; no worry, I will break it down bit by bit and explain what you must do …

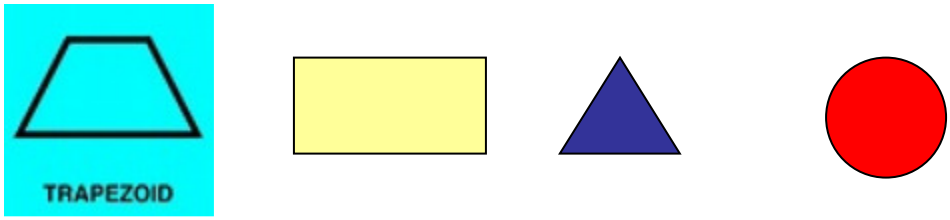
Let us explore the above UML diagram at the highest hierarchy; you can see we have an enum class and three interfaces:



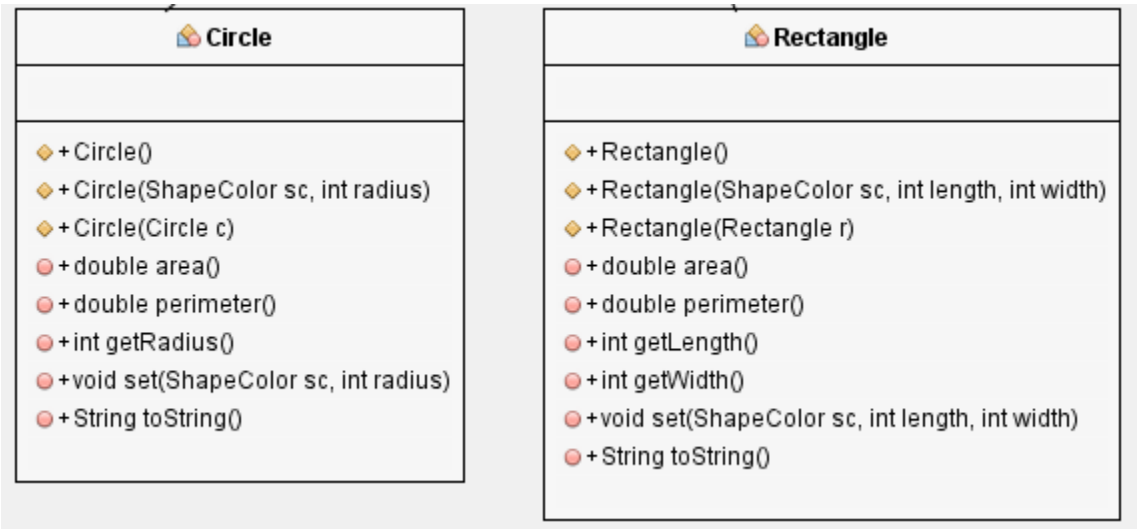
- Enum class defines some enum constants (various colors, feel free to change)
- The main interface should be the Shape interface consists of only one abstract method, the area method. You can put some useful constant in this interface, for example the PI.
- The interface ForTwoD consists of two abstract methods
- The interface OnyThreeD consists of two abstract methods

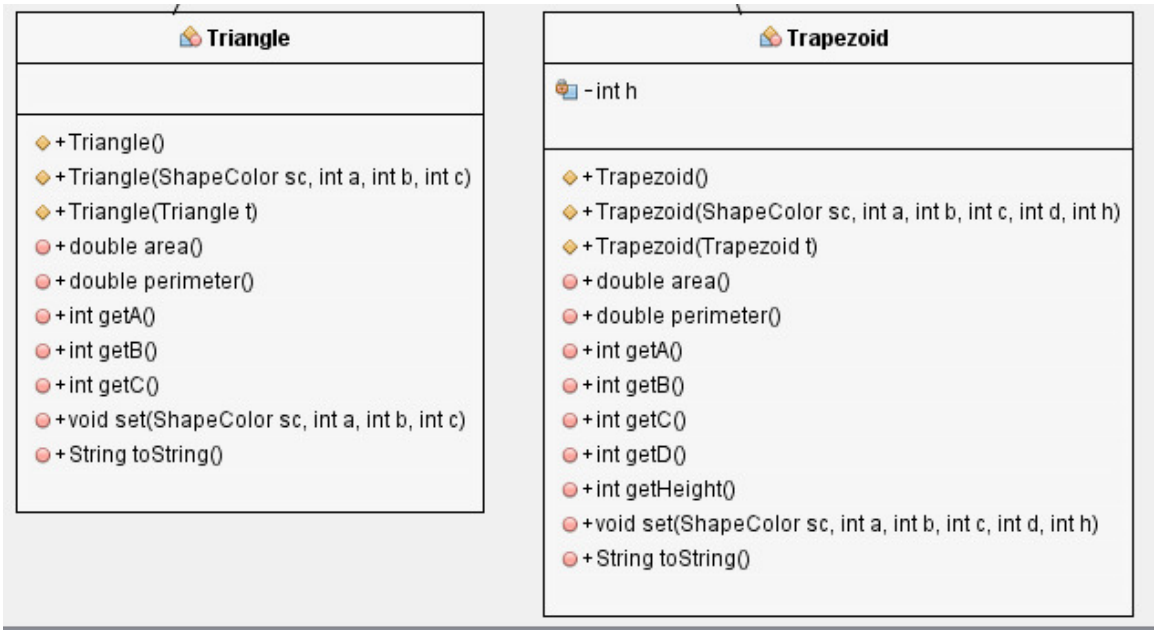
Let us now explore the abstract class TwoD which implements the Shape and the ForTwoD interfaces:

Four possible shapes for `TwoD`: one value, for example the radius, is a circle shape; two values, for example the length and the width, is a rectangle; three values for example the three sides of a triangle (provided it can be formed), five values for trapezoid (one additional value was the height, as we need that to compute the area). In this class, you can see that we have four important constructors to describe the four shapes, a default constructor, a copy constructor, some accessor and mutator methods and a `toString` method. Each 2D shape also has a color and the color can be changed (recolor method) during runtime. You can see we also define an enumeration type to specify the color of the shapes.



The following UML diagram shows the four concrete subclasses of `TwoD`:



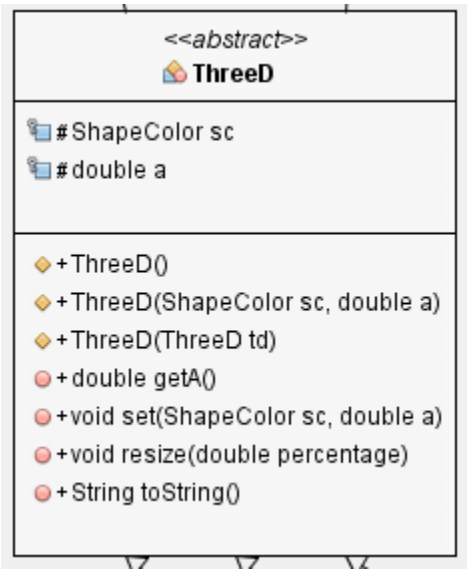


Some methods just override the super class methods (same implementations)

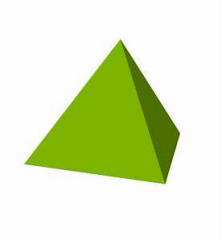
Information defined in each of the subclasses should be obvious in definitions.

Note that in the Trapezoid class, the sides c and d will not be used in the computation, You can assume that a and b are the two parallel sides; and the height is the instance variable h defined in the Trapezoid class.

Next, we explore the abstract class ThreeD which implements the Shape and ForThreeD interfaces:



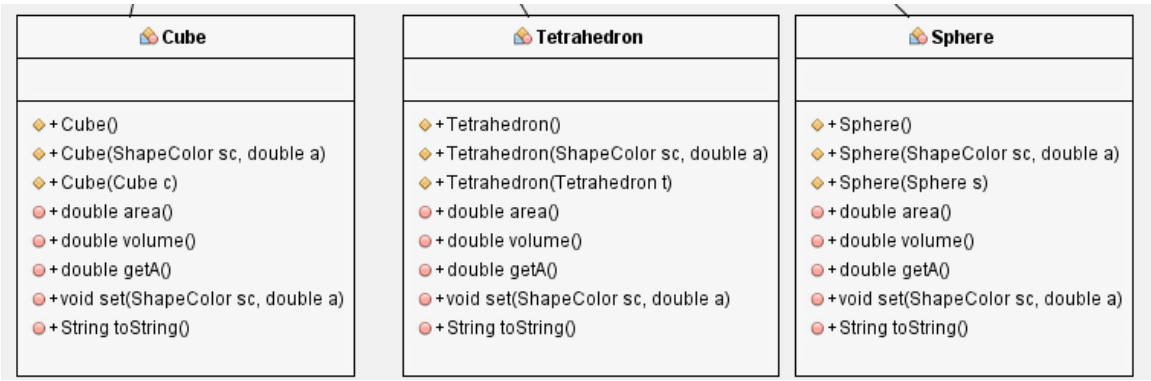




Three possible shapes for `ThreeD`: Just one value can determine the shapes of a sphere, a cube and or a tetrahedron. In this class, we only also have constructors, copy constructor, accessor methods, mutator methods and a `toString` method. For a 3D object, we can compute and return the volume too.

Note that `ThreeD` class also implements `ForThreeD` interface class. The method `resize` is to reduce the size by certain *percentage*.

The following UML diagram shows the three concrete subclasses of ThreeD:



Look for the surface area and volume formulas somewhere in internet to compute and to return their values.

Let us now explore the main class, i.e. main method is defined in this class

All shapes (2D or 3D) should be *randomly generated* and store them in an ArrayList of `Shape's`.

You can see a few private class methods are defined in this class:

- a method generates and returns a positive integer, not too large
- a method generates and returns a positive real number, not too large
- a method generates and returns a color.
- a method to test if three sides can form a triangle.
- a method generates and returns a TwoD shape
- a method generates and returns a ThreeD shape
- the two process methods of 2D and 3D can be called in the displayList (optional)
- a method to display the objects stored in the ArrayList. Note that in the display method, you display the details of each shape object, i.e., the toString methods for each of the classes only display a "brief" object info, display of area / volume / resizable/ recolor should be done in this method.

Convenient to your design, minor updates to methods or additional methods are allowed. The name of classes / methods in the UML diagram cannot be changed

In the main method, you *repeatedly* generate an integer `k` (0 or 1 or 2). If k is 1 you construct a 2D object; if k is 2, you construct a 3D object and k is 0, you end the task. The following shows one of the possible displays:

```
Shape 1: Circle (2D (Green, 7))
Updated color: Red
Area = 153.938
I am a Circle shape with color changed to Red
-------------------------------------------------
Shape 2: Triangle (2D (Red, 8, 7, 5))
Updated color: White
Area = 17.321
I am a triangle shape with color changed to White
-------------------------------------------------
Shape 3: Trapezoid (2D (Red, 5, 5, 7, 9), 2)
Updated color: Green
Area = 10.000
I am a trapezoid with color changed to Green
-------------------------------------------------
Shape 4: Tetrahedron (3D (Yellow, 3.371))
Surface area = 19.681
Volume = 4.514
Size reduced by 17.5%: Tetrahedron (3D (Yellow, 2.779))
Updated surface area = 13.379
Updated volume = 2.530
I am a tetraheron shape
-------------------------------------------------
```

Four objects were generated and stored in an array list and you displayed the list. You can see Shapes 1 to 3 are 2D objects, their colors are changed during runtime (you must make sure that the color is really changed to a different color); Shape 4, a tetrahedron, its sizes, area, volume were reduced by 17.5 % (this percentage was randomly generated).

Note that the list may be empty …

IMPORTANT

Put all your classes in a file called **`YourName_A2.java`** and make sure that this file can be compiled and can be executed. Upload **ONLY** this file to Moodle**. ALL ZIP FILE SUBMISSION WILL BE REJECTED.**

No re-submission will be allowed after grading.

In the above file, remember to put down your name and the following declaration (some similar contents):

// Tell me if it is your own work, and whether you
// have passed your program to your friends etc etc etc
// and willing to accept whatever penalty given to you.

- **Wrong file name -0.5 mark**
- **No declaration, no name etc -0.5 mark**
- **Failing to demo -1 mark**
- **Program's indentations and alignment of statements -0.5 mark**
- **Late penalty: -0.1 mark per hour.**