Abstract Classes and Interfaces

1. Write an abstract class `Shape`
    - Data members: `numSides`
    - Constructor: initialize `numSides`
    - Concrete method: get method for `numSides`
    - Abstract methods: `getArea()`, `getPerimeter()`
    - Write a concrete subclass `Rectangle` – Data members: `width`, `height`
    - Write a concrete subclass `RtTriangle` – Data members: `width`, `height`

- In another class, write a main method to define a `Rectangle` object and an `RtTriangle` object.

- Write an interface `Resizable`
    - Has a method `resize(double x)` that resizes a Shape's dimensions by factor `x`
- Make `Rectangle` implement `Resizable`
- Write a main method to:
    - Define a Rectangle (width = 2, height = 3)
    - Print the Rectangle's area & perimeter
    - Resize the Rectangle by factor of 2
    - Re-print the Rectangle"s area & perimeter

How would you design this code to include a new shape named Circle?

2. What is constructor chaining in JAVA?

Additional Reading if you are interested:

From Java 9 onwards, interfaces can also contain the following:

a. `Static` methods
b. `Private` methods
c. `Private` Static methods

Research the above three things. Try and understand why `static` methods, `private` methods and `private static` methods were included for Interfaces.