

SRP – the Single Responsibility Principle

A class should have only one reason to change

- This principle is also known as *cohesion*; how good reason the elements of a module have to be in the same module.
- SRP is a 'derivative' of cohesion, it relates cohesion to the forces that cause the module (class normally) to change.

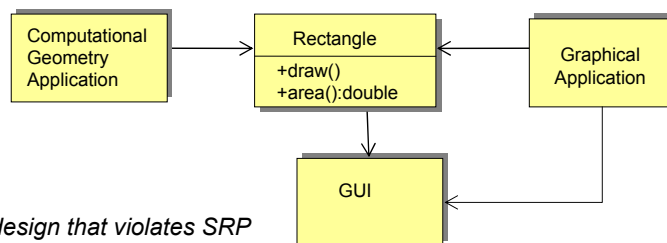
SRP – the Single Responsibility Principle

- The rationale behind the principle:
 - Changes in requirements are manifested as changes in class responsibilities
 - Therefore a 'cohesive' responsibility is a single axis of change – requirement changes often are restricted to a few cohesive responsibilities (In a reasonably designed system)
 - Thus, to avoid coupling responsibilities that change for different reasons, a class should have only one responsibility, one reason to change
- Violation of SRP causes spurious transitive dependencies between modules that are hard to anticipate, in other words *fragility*

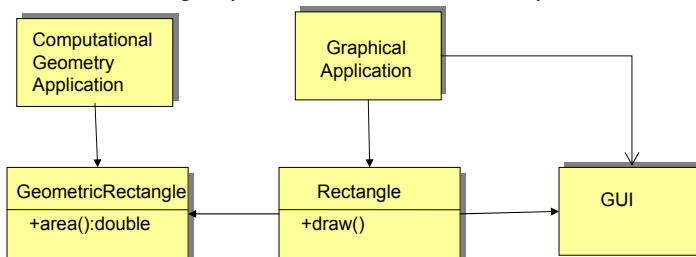
What is a 'responsibility'?

- A responsibility is thought of being a larger set of functionality that serves one purpose.
- Seeing the responsibilities of a class may be hard
 - You really can not understand the responsibilities of a class without understanding the needs of the clients of that class
 - In the context of SRP you should look at responsibilities as “reasons to change”, i.e. how are possible changes in the needs of the clients reflected to a single module or class.
- A responsibility is an axis of change if that change will really occur.
- Anticipating change is hard, SRP applied too often leads to needless complexity.
 - Do not apply SRP if there is no symptom.

SRP – Example



Corrected design, split the class with several responsibilities



Separating the responsibilities

- Often it is not possible to use the standard solution – splitting the class.
- Separating the responsibilities into two interfaces decouples the two responsibilities as far as rest of the application is concerned. This yields us most of the benefits of SRP

