# Everything on Refactoring

Rewriting existing source code, with the intent of improving its design rather than changing its external behavior. Applicable at the design level too

## High level refactoring operations

### Collapsing Objects

If class B is a part of class A by aggregation or composition, and class B is associated with no other classes, we may be able to remove class B, absorbing its attributes into class A (private). Its operations might become private or disappear

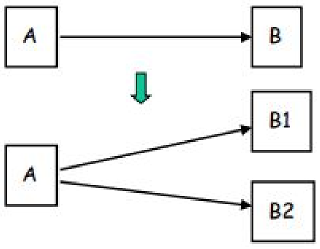
### Partial collapsing rules

Attributes of a class that are only used in getters and setters, are candidates from moving to the client class. They are moved  or in-lined and deleted.

The opposite of collapsing is splitting

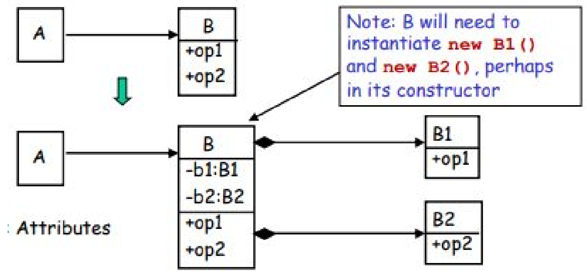
### Splitting classes

Splitting complex classes into two or more new independent classes results to new classes with more focused identity. This is possible if the class is poorly cohesive.

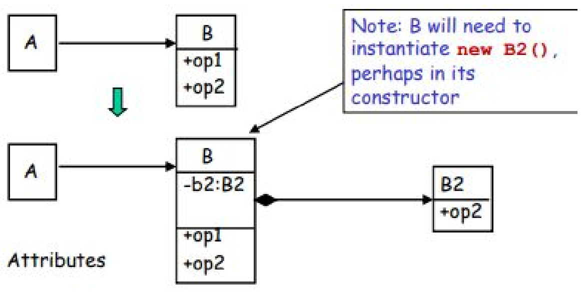


This is not good refactoring. Changes to A need to be made.

In general it may be best to retain B as a container for instances of the new classes. B would keep the same public operations, but their bodies would delegate the calls to appropriate instances of the new classes. A won't change either.


A
B
+op1
+op2
Note: B will need to instantiate new B1() and new B2(), perhaps in its constructor
A
B
-b1:B1
-b2:B2
+op1
+op2
: Attributes
B1
+op1
B2
+op2

The splitting technique can also be used for cohesive parts of the class, in order to produce a new class with a more focused identity. An instance of the new class would be aggregated/composed into the remains of the original class.


A
B
+op1
+op2
Note: B will need to instantiate new B2(), perhaps in its constructor
A
B
-b2:B2
+op1
+op2
B2
+op2
Attributes

During development there may be classes that share attributes and operations but with specialised components. Introducing superclasses and inheriting from them may enable more and beneficial code reuse.

Benefits may be short term if this gives implementation inheritance. If so class splitting and delegation may work better.