

# Craig Larman's "GRASP"

## General Responsibility Assignment Software Principles

Information Expert	Assign responsibility to the class that has the information needed to fulfill it.
Creator	Assign <code>class B</code> the responsibility to create an instance of <code>class A</code> if <code>class B</code> closely uses, aggregates, contains, or records <code>class A</code>
Controller	Assign the responsibility for handling a system event to a class representing one of these choices: the overall system, device, subsystem or represents a use case scenario where the system event occurs
Low Coupling	Assign responsibilities so that (unnecessary) coupling remains low.
High Cohesion	Assign responsibilities so that cohesion remains high.
Polymorphism	When related alternatives or behaviors vary by type (class), assign responsibility for the behavior (using polymorphic operations) to the types for which the behavior varies.
Pure Fabrication	Assign a highly cohesive set of responsibilities to an artificial or convenience "behavior" class that does not represent a problem domain concept (something made up)
Indirection	Assign the responsibility to an intermediate object to mediate between other components or services, so that they are not directly coupled.
Protected Variations	Identify points of predicted variation or instability; assign responsibilities to create a stable "interface" around them.

SOLID