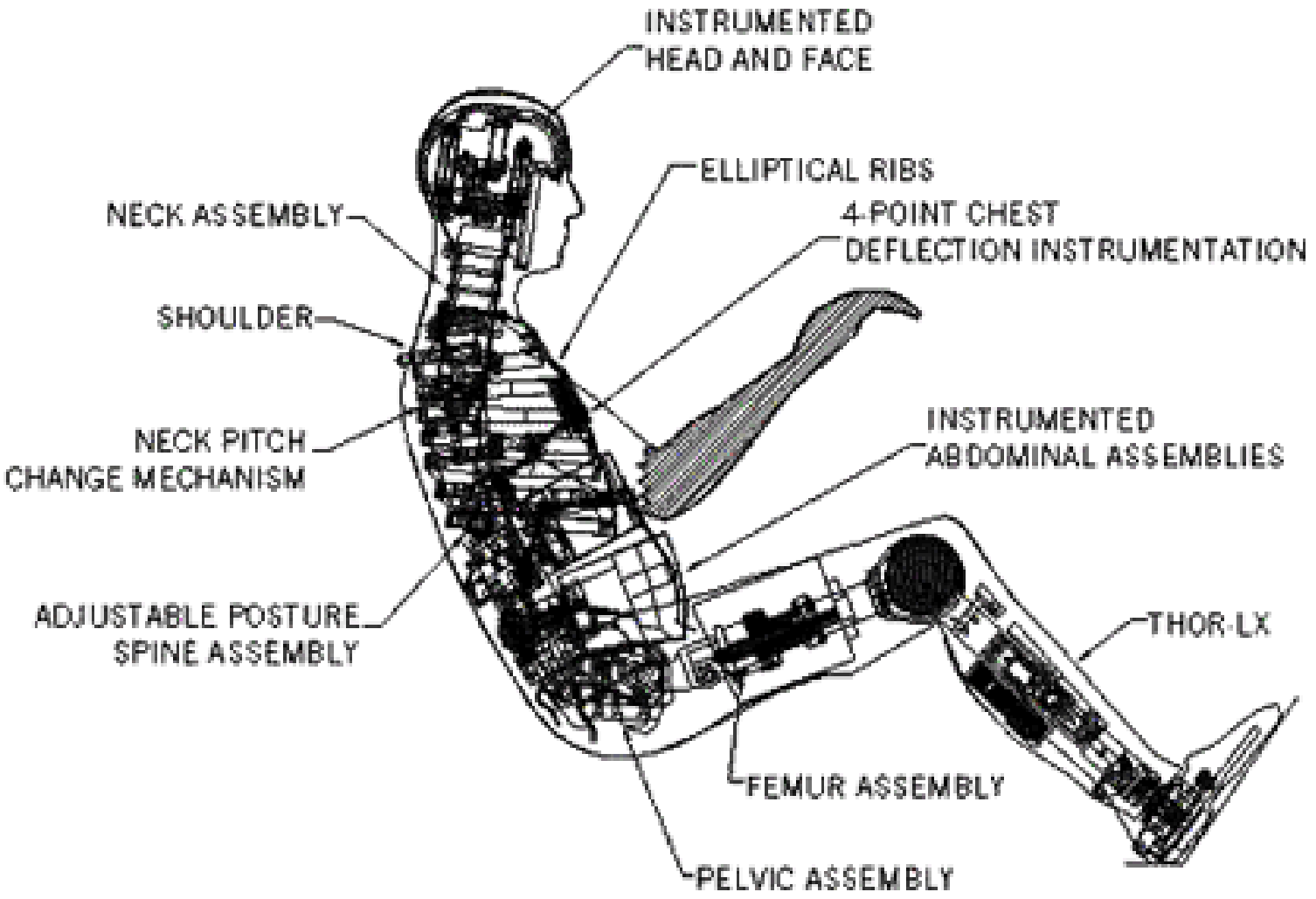
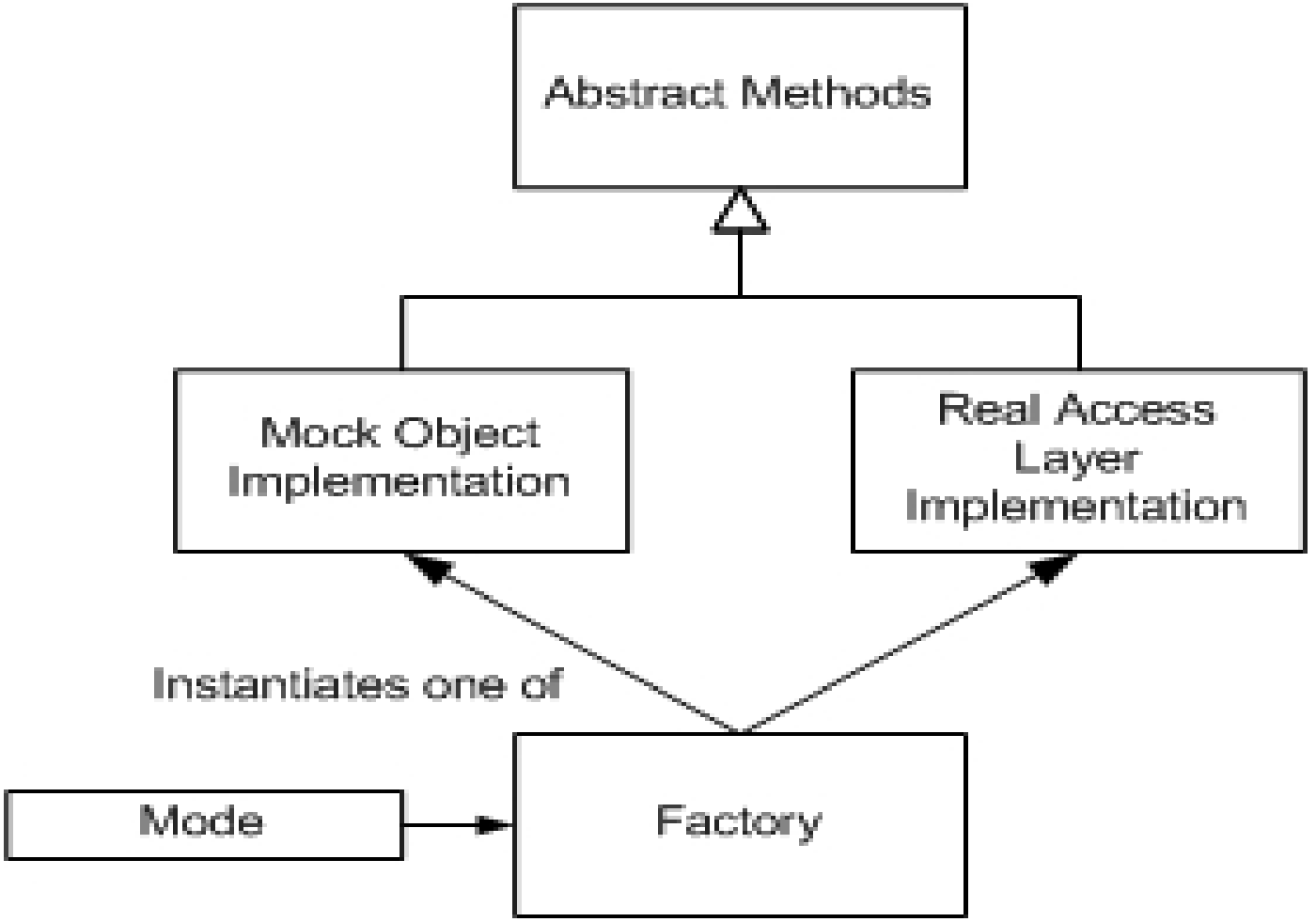
Mock Objects

- They are simulated objects that mimic the behavior of real objects in controlled ways
- 'Mocks' are typically used to test the behavior of some other object, much like a crash test dummy is used to simulate the dynamic behavior of a human in vehicle impacts
- They contain enough state and/or behavior to subsitute for the real object
- They are extremely useful in unit testing


INSTRUMENTED
HEAD AND FACE
ELLIPTICAL RIBS
NECK ASSEMBLY
4-POINT CHEST
DEFLECTION INSTRUMENTATION
SHOULDER
INSTRUMENTED
ABDOMINAL ASSEMBLIES
NECK PITCH
CHANGE MECHANISM
THOR-LX
ADJUSTABLE POSTURE
SPINE ASSEMBLY
FEMUR ASSEMBLY
PELVIC ASSEMBLY

Mock Object UML via Factory

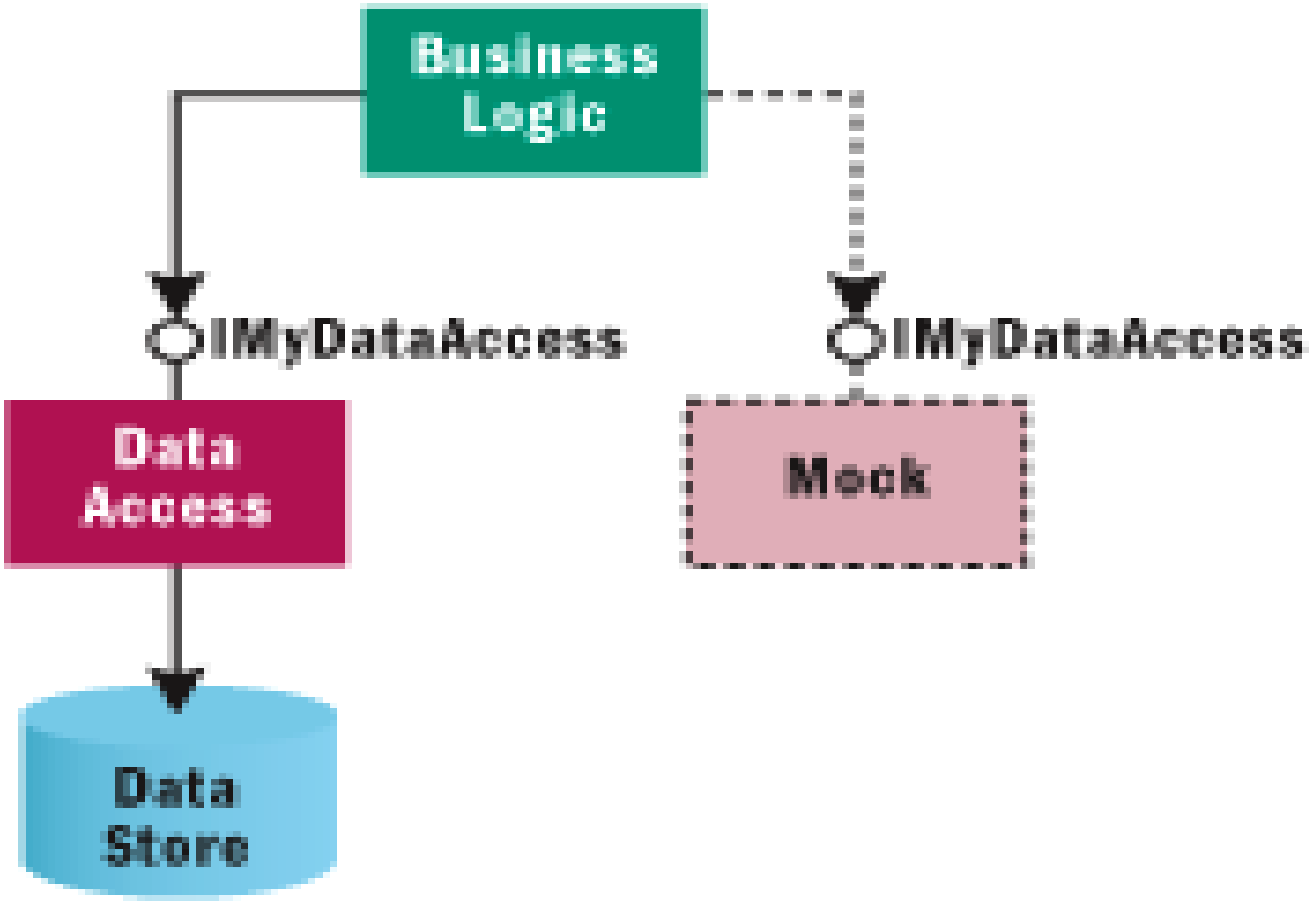


Mock Objects and Patterns

- Mocks can be used in place of real objects in a variety of design patterns
  - Factory
  - Adapter
  - Proxy
  - Dependency Injection
  - Strategy
  - Observer
  - State
  - The list goes on!

When Mocks are Most Useful

- Substitute for real object during tests when real object
  - Supplies non-deterministic results (results that are somewhat or completely unpredictable) (temperature, current time, etc.)
  - Has states that are difficult to create or reproduce (e.g., a network error)
  - Is slow
  - Does not yet exist or may change behavior
  - Would have to include information and methods exclusively for testing purposes (breaks SRP)


Business Logic
IMyDataAccess
IMyDataAccess
Data Access
Mock
Data Store

Mocks implement same interface as real object

- This allows client to remain unaware of whether object is real or mock. Mocks are thus an application of Dependency Injection!
- Mock object frameworks allow the programmer to specify which, and in what order, methods will be invoked on a mock object and what parameters will be passed to them, as well as what values will be returned.
- Complex behaviors such as network sockets can be mimicked by a mock object.

Mocks, Fakes, and Stubs

- You will sometimes see these terms used interchangeably, but there are differences
- Fakes are simpler than mocks, simply implementing the same interface as the real object. Fakes typically supply only method stubs, which yield Null Object behavior
- Mocks have method implementations that contain assertions of their own. The methods can examine the context of each call, perform tests on the data passed in as arguments, and return an appropriate value
- Generally speaking, 'Fakes' are defined as anything that is not real. Depending on usage, they are either stubs or mocks.

Mock Object Final Thoughts

- Used primarily for testing other objects in a system
- Can provide pre-determined behavior so outcome of tests can be easily evaluated
- Mocks feel like a fancier version of Null Object
- Good use of abstraction in your code makes it (very) easy to plugin a Mock Object in the place of a real one
- Use Mock Objects liberally to test your code. They are lightweight and easy to inject into your existing OO framework