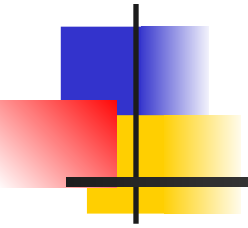

3 Fundamentals of Object-Oriented Programming

A Summary

3 Pillars of O-O Paradigm

1. Encapsulation
2. Inheritance
3. Polymorphism

Encapsulation

A class or object contains both data (state) and the methods that use the data.

Inheritance

One class can completely reuse (inherit) the structure of another class, including both attributes and methods.

What's Not Inherited

a) constructors - they are specific to one class

b) static methods? - you can invoke them via subclass name.

Polymorphism

We can invoke a method of an object without knowing the actual type of object that will perform the method.

Example: **`System.out.println( object )`**

`println` calls **`object.toString()`** without knowing which kind of object (which class) will perform toString().

Hence, println can print anything. Even objects it know nothing about.

Why Polymorphism is Useful

We can substitute one kind of object in place of a different kind, and the application still works!

This enables 1) reuse of software, 2) frameworks that we tailor for a specific app.