**Design Pattern Java Companion**

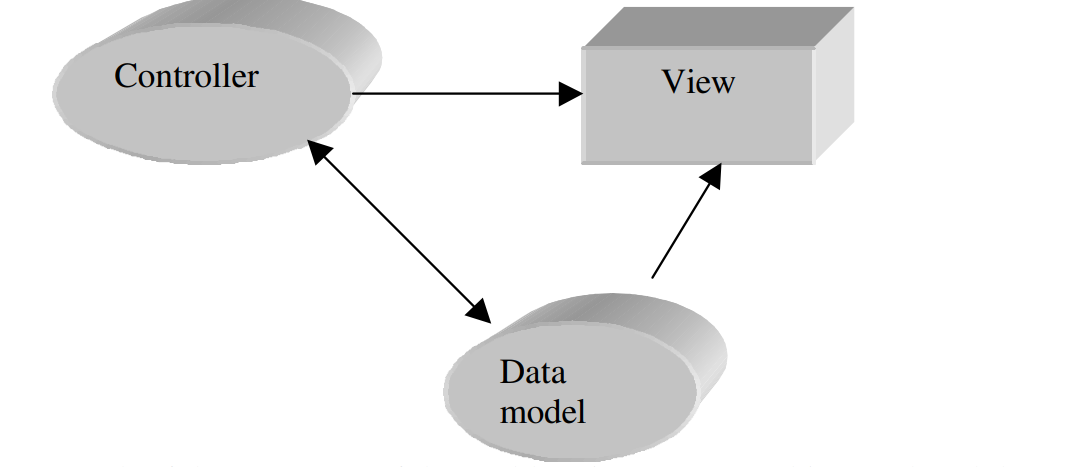
**- James W. Cooper**

Notes by Hitesh Ahuja

**Introduction**

**Design Patterns** are just convenient ways of reusing object-oriented code between projects and between programmers.

One of the earliest framework which creates a structure to separate different part of code to reduce complexity was MVC framework.This framework divides the user interface problem into three parts. The parts were referred to as a ***data model*** which presented the user interface, and the ***controller***, which interacted between the user and the view.



Each of these aspects of the problem is a separate object and each has its own rules for managing its data.  
  
So based on the above example, design patterns describe how objects communicate without become entangled in each other’s data models and methods.

**Defining Design Patterns**

Some useful definitions of design patterns :

* “Design patterns constitute a set of rules describing how to accomplish certain tasks in the realm of software development.” (Pree, 1994)
* “A pattern addresses a recurring design problem that arises in specific situations and presents a solution to it” ” (Buschmann, et. al. 1996)

**Note**

Design patterns are not just about the design of objects, but about the *communication between objects.*

Based on the original *Design Patterns* book consisting of 23 design patterns, they are divided into three types

* **Creational pattern** are the ones that create objects for youm rather than having you instantiate objects directly. This gives your program more flexibility in deciding which objects need to be created for a given case.
* **Structural patterns** help you compose groups of objects into larger structures, such as complex user interfaces or accounting data.
* **Behavioral patterns** help you define the communication between objects in your system and how the flow is controlled in a complex program.