**AOP Aspect-Oriented Programming Overview**

**Aspect-Oriented Programming (AOP)**:

In computing, aspect-oriented programming (AOP) is a programming paradigm that aims to increase modularity by allowing the separation of cross-cutting concerns. Aspect-oriented programming entails breaking down program logic into distinct parts (so-called concerns, cohesive areas of functionality).

**Concern**:

A Concern is a term that refers to a part of the system divided on the basis of the functionality. Concerns are two types.

* Core concerns
* crosscutting concerns or system-wide concerns.

**Core concern**:

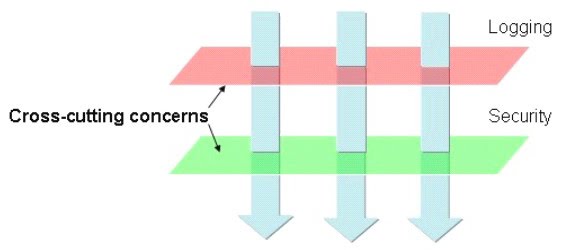
The concerns representing single and specific functionality for primary requirements are known as core concerns or primary functionality of the system is knows as core concerns.

For example: Business logic.

**Crosscutting Concern**:

The concerns representing functionalities for secondary requirements are referred to as crosscutting concerns or system-wide concerns. Or, the crosscutting concern is a concern which is applicable throughout the application and it affects the entire application.

For example: logging, security and data transfer are the concerns which are needed in almost every module of an application, hence they are cross-cutting concerns.



**Aspect**:

An aspect of a program is a feature linked to many other parts of the program, but which is not related to the program's primary function. An aspect crosscuts the program's core concerns, therefore violating its separation of concerns that tries to encapsulate unrelated functions. For example, logging code can crosscut many modules, yet the aspect of logging should be separate from the functional concerns of the module it cross-cuts.

**Keyword**:

AOP -> Aspect Oriented Programming

Aspect -> A Service

Advice -> Service Provider

Point Cut -> A point or condition to execute aspect for business method.

Adviser -> Point Cut with Advice Combination

AOP Aspect-Oriented Programming Overview