**Introduction to Design Patterns**

**In this tutorial we will discuss**  
1. What are Design Patterns  
2. Evolution of Design Patterns   
3. The three categories of design patterns    
  
In many interviews you might have encountered lot of questions related to interfaces, abstract classes, delegates and other features related to object oriented programming along with the design pattern related questions.    
  
  
  
Design patterns solutions are evolved from the  features of object oriented programming.    
  
Once you understand the design patterns, it makes you very comfortable in attending interviews as well as applying these features with confidence in your projects or applications.  
    
Implementing design patterns in the applications are proven and tested. Writing the code aligning with design patterns will make your applications Reliable,Scalable and Maintainable.    
  
  
  
**What are Design Patterns**   
Design patterns are reusable solutions to the problems that we encounter in the day to day programming. They are generally targeted at solving the problems of object generation and integration. In other words, Design patterns acts as templates which can be applied to the real-world programming problems.  
  
**History and evolution of design Patterns**  
The four authors of the book famously know as Gang of four are the ones who brought the concepts of design patterns in their book “Elements of reusable Object-Oriented software” .   
   
Gang of Four has divided the book into two parts with first part explaining about the pros and cons of object oriented programming and the second part describes the evolution of 23 classic software design patterns.  
   
For more details, please refer to the following wikipedia article  
<https://en.wikipedia.org/wiki/Design_Patterns>  
  
**Types of Design Patterns**  
Gang of Four have categorised the design patterns in to 3 types based on different problems encountered in the real world applications. They are Creational, Structural and Behavioural.  
   
**Creational design patterns :**These patterns deal with object creation and initialization. Creational pattern gives the program more flexibility in deciding which objects need to be created for a given case.  
   
**Examples of Creational design patterns category :**Singleton , Factory and Abstract Factory etc.  
  
**Structural design patterns :**This pattern deals with class and object composition. In simple words, This pattern focuses on decoupling interface, implementation of classes and its objects.   
  
**Examples of Structural design patterns category :**Adapter,  Facade and Bridge etc.  
  
**Behavioural design patterns :**These patterns deal with communication between Classes and objects.   
  
**Examples of Behavioural design patterns :** Chain of Responsibility, Command and Interpreter etc.  
   
For understanding design patterns better it is very important to have basic knowledge about the following object oriented concepts

* Abstraction
* Inheritance
* Polymorphism
* Encapsulation
* Interfaces
* Classes
* Abstract classes

In Part 2 we will focus on the Singleton design pattern in detail with a simple example.