**Abstraction in Java**

**Abstraction** is the concept of exposing only the required essential characteristics and behavior with respect to a context.

Hiding of data is known as **data abstraction**. In object oriented programming language this is implemented automatically while writing the code in the form of class and object.

**Real Life Example of Abstraction**

Abstraction shows only important things to the user and hides the internal details, for example, when we ride a bike, we only know about how to ride bikes but can not know about how it work? And also we do not know the internal functionality of a bike.

**Note:**Data abstraction can be used to provide security for the data from the unauthorized methods.

**Note:**In Java language data abstraction can achieve using class.

**Example of Abstraction**

**class** Customer

{

**int** account\_no;

**float** balance\_Amt;

String name;

**int** age;

String address;

**void** balance\_inquiry()

{

/\* to perform balance inquiry only account number

is required that means remaining properties

are hidden for balance inquiry method \*/

}

**void** fund\_Transfer()

{

/\* To transfer the fund account number and

balance is required and remaining properties

are hidden for fund transfer method \*/

}

**How to achieve Abstraction ?**

There are two ways to achieve abstraction in java

* Abstract class (0 to 100%)
* Interface (Achieve 100% abstraction)

Read more about Interface and Abstract class in the previous section.

**Difference between Encapsulation and Abstraction**

Encapsulation is not providing full security because we can access private member of the class using reflection API, but in case of Abstraction we can't access static, abstract data member of a class.