***SINGLETON CLASS***

WHAT DO YOU MEAN BY SINGLETON CLASS?

A Class that has only one instance and provides a global point of access to it.

WHAT IS USE OF SINGLETON CLASS?

Object creation is costly process in java. When objects gets created it has occupies the memory. So, we need to stop the process of creating the objects at all the time and restrict them use at single instance at a time. So it produce some advantages.

1. Save memory – Cause single instance we are using again and again.
2. It is used in the multithreading context, for data consistency.
3. Used in data configuration and setting.
4. Used when connecting to database.

TELL ME ABOUT SINGLETON CLASS?

Defination : **"define a class that has only one instance and provides a global point of access to it"**

**There is two form of singleton design patterns**

1. **Early instantiation :**

**Create a class instance a class loading time.**

1. **Lazy instantiation**

**Define class reference at class loading time and create a class instance when we required.**

HOW TO CREATE SINGLETON CLASS?

1. **Static Members: Gets memory only once because of static, Its contains the instance of single class.**
2. **Private Constructor: It will prevent to instantiate the singleton class from outside the world.**
3. **Static factory Method: Provide a global point of access to the singleton object from method “getInstance()”, and return from its instance to the caller.**

Example of SINGLETON CLASS?

**// We have to create singleton class using two ways 1. Early and 2. Lazy instantiation.**

**// following the example for Early instantiation.**

**Class Singleton{**

**private static Singleton obj = new Singleton();**

**private Singleton(){}**

**public Singleton getInstance(){**

**return obj;**

**}**

**// Another methods**

**}**

**// following the examples for Lazy instantiation.**

**Class Singleton{**

**private static Singleton obj;**

**private Singleton(){**

**}**

**public static Singleton getInstance(){**

**if(null==obj){**

**obj = new Singleton();**

**}**

**return obj;**

**}**

**//Another Methods**

**}**