Deleting an Object

- Whenever an object is no more longer in use the it recommended to delete an object
- Advantages are memory optimization
- In order to delete an object simply we have to place "null" to the reference
- After placing null to the reference then the corresponding object becomes unused
- Garbage collection is the sub program of the JVM is always to manage the unused objects in the memory
- By chance if we are trying get something from the by reference , which is referred to "null" then JVM will raise "java.lang.NullPointerException"
- Garbage collector is the sub program of JVM , it will erased the object which are existed in the garbage collection, but we not guarantee in invocation of the Garbage collector soon after deleting an Object, Just because it is designed to run in different intervals
- Based on the application requirements ,we can also invoke the garbage collector by using the following method

```
java.lang.System
    public stativ void gc( );
        Eg: System.gc( )
```

- Note : Whatever the resources are allocated by time of object instantiation, Then it is our responsibility to perform resource DE allocation by the time deleting an Object

- In order to perform resource DE allocation then we have to make use of the following method

```
java.lang.Object
   protected void finalize( ){ }
```

This method will be executed automatically whenever the system.gc( ) is called.

```
class Sample{
       int x,y; //instance fields

       Sample( )
              { x=111; y=222; }

       void getData()
       { System.out.println("x val is : "+x);
         System.out.println("y val is : "+y); }

//java.lang.Object
       //protected void finalize();
       protected void finalize( )
```

```
      { System.out.println("Object is deleted ...");
        System.out.println("Res.D.A.Done....!!!!");}

       public static void main(String args[ ])
      {
              Sample s1=new Sample( );
              Sample s2=new Sample();
              s1.getData();
              s1=null;
        // s1.getData( );  //java.lang.NullPointerException

              //java.lang.System
                //public static void gc( );
                   System.gc( );

              System.out.println("Data from s2");
              s2.getData();
      }
}
```