**SingletonDesign pattern:-**

Ways to break singleton:

1)cloning

2)multithreading

3)serialization and Deserialization

4)custom classloader

5)reflection api

Breaking Singleton design pattern by Clone();

public final class SingeltonCloneTest implements Cloneable {

/\*\*

\* @param args

\* @return

\*/

private static SingeltonCloneTest instance = null;

private SingeltonCloneTest() {

System.out.println("Rahul Tripathi");

}

public static SingeltonCloneTest getInstance() {

if (instance == null) {

instance = new SingeltonCloneTest();

return instance;

}

return instance;

}

@Override

protected Object clone() throws CloneNotSupportedException {

// TODO Auto-generated method stub

/\*

\* Here forcibly throws the exception for preventing to be cloned

\*/

throw new CloneNotSupportedException();

// return super.clone();

}

public static void main(String[] args) {

// TODO Auto-generated method stub

SingeltonCloneTest test1 = SingeltonCloneTest.getInstance();

try {

SingeltonCloneTest test2 = (SingeltonCloneTest) test1.clone();

} catch (CloneNotSupportedException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}