**Native modifier**

Native is modifier ,applicable only for methods and we cant apply anywhere else

The methods which are implemented in non java (mostly c or c++) or called native methods or foreign methods

The main objectives of native keyword are 1) to imrove performance of the system 2) to achive machine level or memory level communication.3)to use already exsiting legacy non java code etc.

Pseudo code to use native keyword in java

Class native

{

Static

{

System.loadlibrary(“native library”) 1)Load native library

}

Public native void m1(); 2)Declare a native method

}

Class client

{

P S V M(S a)

{

Native n=new native(); 3)Invoke a native method

n.m1();

For native methods implementation is already available in old languages like c or c++.and we are not responsible to p’rovide implementation hence native method declareation should ends with ‘ ; ’ .

Public native void m1(); >>>>>>>>>> correct

Public native void m1() {}

Ce: native methods cannot have body

For native method implementation is alredy available in old languages but for abstract method ,implementation should not ;;be available hence we cant declare native method as abstract.i.e. native abstract combination is illegal combination for methods.

We cant declare native method as strictfp bcoj there is no guaranty that old languages follow IEEE 754 standard hence native strict fp combination is illegal combination for methods

The main advantage of native keyword is performance will be improved ,but main disadvantage of native keyword is it breaks platform independent nature of java.

**Transient keyword**

Transient is modifier applicable only for variable .we can use transient keyword in serialization context

At the time serialization if we don’t want to save value of particular variable to meet security constrint then we should declare that variable as transient.at the time of serialization jvm ignores original value of transient variable and save default value to file hence transient means not to serialize

**Volatile modifier**

Volatile is modifier applicable only for variables and we cant apply anywhere else

If the value of the variable is keep on changing by multiple threads then there may be a chance of data inconsistency problem,we can solve this problem by using volatile modifier

If variable is decalred as volatile then for every thrad jvm will create a separate local copy .every modification performed by that thread will takes place in local copy so that there is no affect on remaining threads.

The Main advantage of volatile keyword is we can overcome data inconsistency problem,but main disadvantage of volatile keyeord is creating and maintaining a separate copy for every thread increases complexity of prograsmig and create performance problem hence if there is no specific requirement then it is never recommended to ;use volatile keyword and it is almost deprecated keyword.

Final variable means the value never changes whereas volatile variable means value keep on changing hence volatile-final is illegal combination for variables.