**Static modifier**

Static is modifier applicable for methods and variables but not for classes .we cant declare top level class with static modifiers but we can declare inner class as static (such type of inner class is called Static nested classes)

In the case of instance variable for every object ,a separate copy will be created,but in the case of static variables ,a single copy will be created at class level and shared by every object of that class

Eg

Class test {

Static int x=10;

Int y=20;

P S V M(S [] a) {

Test t1 = new Test();

T1.x = 888;

T1.y= 999;

Test t2 = new test ();

Sop(t2.x……….t2.y);

} }

o/p>>>>>>>>>> **888……………..20**

we cant acces instance member directly from static area but we can access from instance area directly.

We can access static member from both Instance and static directly.

Caase 1

Overloading concept applicable for static methods including main method ,but jvm will call string array argument main method only.

e.g.

class test{

P S V M(String [] args)

{ sop (“String []”); }

P S V M(int [] args)

{ sop (“int []”);}

}

o/p >>>>>>>>>>>> String []

Other overloaded method ,we have to called just like a normal method call..

Case 2

Inheritance concept applicable for static method including main method hence while executing child class ,if child ddoesn contain main method then parent class main method will executed.

Case 3

Class p{

P S V M(String [] args)

{sop(parent);

}

}

Class c extends P

{

P S V M(String [] args)

{

Sop(child class);

}

}

o/p

Java p >>>>>>>>>>>> parent

Java c >>>>>>>>>>>>> child class

It seems overriding concept applicable for static methods,but it is not overriding but is method hiding.

Note-for static method overloading and inheritance concepts are applicable but overriding concept are not applicable but instead of overriding method hiding is applicable

Inside method implementation ,if we are using atleast one instance variable then that method talks about a particular object ,hence we should declare method as instance method.

Inside method implementation if we are not using any inastance variable then this method no way related to a oerticaluar object hence we have to declare such tyoe method as static method irrespective of wheatherr we r using static variable or not.

For static method implantation should ;be available but for abstract method imolementation is not available hence abstract static combination is illegal for methods.

**Synchronised Modifier**

Synchronised is modifier applicable for methods and blocks but not for classes and variable.

If multiple threads trying to operate simultaneously on the same java object then there may be chance of data inconstancy problem.this is caleed race condition.we can overcome this problem by using synchonised keyword.if a method or block declare as synchronized then at a time only one thread is allowed to execute that method or block on given object so that data inconstistancy problem wiil be resolved,but main disadvantage of synchronized keyeord is it increases waiting time of threads and creates performance problems hence if there is no specific requirement then it is not recoomended to use synchronized keyword

Synchronized method should compulsory contain implementations whereas abstract method doesn’t contain any implementations hence abstrsact synchronized is illegal combination of modifiree for methods