(4) We can't declare interface variables with the following modifiers.
private, protected, transient, volatile.

(5) There are no restrictions.

(6) For interface variables, compulsary we should perform initialization at the time of declaration otherwise we will get CT Error.

(6) not required to initialize.

(7) Inside Interface, we can't declare the instance and static block

(7) We can declare no problem.

(8) We can't declare constructors.

(8) We can declare but executed at the time of child object creation.

Q:- Diff b/w Overloading and Overriding :→

Sol^n:- Overloading :→

Two methods are said to be overloaded iff both methods having the same name but different argument types.

```
class Test
{
    public void m1(int i)
    {
    }
    public void m1(long l)
    {
    }
}
```

→ overloaded methods.

Overriding :→

Whatever methods parent has bydefault available to the child through inheritance. sometimes child may not satisfy with parent methods implementation. Then child is allow to redifine that method based on it's requirement. This process is called overriding.

* The parent class method which is overriden is called overridden methods.
* The child class method which is overriding is called overriding method.

Ex:-

```
class Parent
{
    public void property()
    {
        s.o.p("cash + land + gold");
    }
    public void marry()
    {
        s.o.p("x-girl");
    }
}
class child extends property
{
    public void marry()
    {
        s.o.p("y- girl");
    }
}
```

Overridden methods ← `public void marry()`

Overriding methods. ← `public void marry()`

Method signature :-

combination of method name and argument types of the method.