```
=======================================
***State three uses of final keyword***
=======================================
- final is a predefined keyword.
- Following are the uses of final keyword:
1) To make constant variable:
-----------------------------
- If we declare the variable using final keyword then that variable become constant
in java.
- Constant variable means, once it is created then we can not change its value
later.
- Constant us a variable which can not change its value during the execution of
program.
Example:
class finalKeywordDemo
{
        public static void main(String args[])
        {
                final float PI=3.14f;           //constant variable
                int radius=2;
                float area;
                area=(PI*radius*radius);
                System.out.println("Area of Circle="+area);
        }
}
/*
Area of Circle=12.56
*/

2) To avoid method overriding:
------------------------------
- Suppose, base class method name and derived class method name are same then base
class method overriden by derived class method.
- If we want to avoid method overriding then we can use final keyword before the
base class method declaration.
- Example:
//To avoid method overriding
class Base
{
        final void display()
        {
                System.out.println("display method of base class");
        }
}
class Derived extends Base
{
        void display()
        {
                System.out.println("display method of derived class");
        }
```

```
}
class AvoidMethodOverriding
{
        public static void main(String args[])
        {
                Derived d1=new Derived();
                d1.display();
        }
}
/*
AvoidMethodOverriding.java:11: error: display() in Derived cannot override
display() in Base
        void display()
             ^
  overridden method is final
1 error
*/
```

3) To avoid inheritance:

- To avoid inheritance then we can declare the base class using final keyword.
- If we declare base class using final keyword then we can not create subclass from it.
- Example:

```
//To avoid method overriding
final class Base
{
        void display()
        {
                System.out.println("display method of base class");
        }
}
class Derived extends Base
{
        void show()
        {
                System.out.println("show method of derived class");
        }
}
class AvoidInheritance
{
        public static void main(String args[])
        {
                Derived d1=new Derived();
                d1.display();
                d1.show();
        }
}
/*
AvoidInheritance.java:9: error: cannot inherit from final Base
```

```
class Derived extends Base
                      ^
1 error
*/
```