Final Keyword

Final keyword can use with variable, with function and with class also.

Final variable: final variable means variable cannot modify its value once we assign it In Short we can say final variable is used for declare the constant in java.

```
public class FinalVarApplication {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
      final int a=10;
        ++a;
       Sys
    }

}
```

The final local variable a cannot be assigned. It must be blank and not using a compound assignment

1 quick fix available:

Remove 'final' modifier of 'a'

Press 'F2' for focus

If we think about above code it will generate the error to us because we try to modify the value of variable a and we declare the variable a as final variable. We cannot modify the value of final variable.

Final method

Final method means method cannot override in child class called as final method.

Before learn the final method we have to know what the method is overriding.

Method overriding

Method overriding means if we define the method in parent class and redefine the same method in child class called as method overriding.

```
class A
{
   void show()
   {
     System.out.println("I am method A");
   }
}
class B extends A
{
   void show()        Here we override the show() method
   {
      System.out.println("I am method B");
   }
}
```

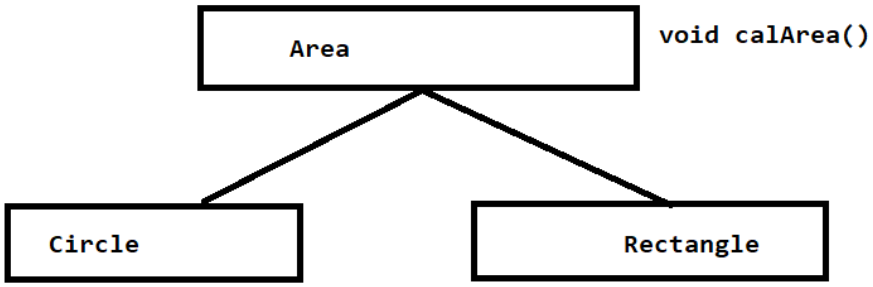
Note: in the case of method overriding if we create the object of child class and try to call the overridden method then by default child logic get executed.

```java
package org.techhub;                    output : ----->

class A
{   void show()
    {   System.out.println("I am method of A");
    }
}
class B extends A
{   void show()
     { System.out.println("I am method of B");
     }
}
public class FinalVarApplication {

     public static void main(String[] args) {
       B b1 = new B();
        b1.show();
     }
```

```
<terminated> FinalVarApplication [Java Applicat
I am method of B
```

Q. Why we need to perform method overriding or what is the benefit of method overriding?

Method overriding is used for normally when we want customize the parent logics in child class according to requirement of child Then we can use the method overriding.




```
void calArea()
{
//here we can write the
 //logic of circle area
}
```

```
void calArea()
{
   //here we can write the logic of
  //Rectangle Area
}
```

Note: when we create the object of Circle class
and call the calArea() method then we get
result of circle area.
When we create the object of Reactangle class
and call the calArea() method then we get
result of Reactangle Area

Example2

```
class Music
{
    void play()
    {
      System.out.println("playing music");
    }
}
class Sony extends Music
{
   void play()
   {
     System.out.println("Here sony music playing logic");
   }
}
class Samsung extends Music
{
   void play()
   {
     System.out.println("Here samsung music playing logic");
   }
}
```

Example3

```
class Bank
{
   int getInterestRate()
   { return  0;
   }
}
class ICICI extends Bank
{
   int getInterestRate()
   { return  8;
   }
}
class HDFC extends Bank
{
   int getInterestRate()
   { return  9;
   }
}
public OverridingApp
{  public static void main(String x[])
    {  ICICI i = new ICICI();
       System.out.println("ICICI Interest rate is  "+i.getInterestRate()); // 8
       HDFC h = new HDFC();
       System.out.println("HDFC Interest rate is "+h.getInterestRate());// 9
   }
}
```

Note: every bank charge interest rate but every bank having different rate so we create one generalize class name as Bank and customize the interest according to bank and its expenses.
means we customize the getInterestRate() method in ICICI with 8 percent and we customize the getInterestRate() method in HDFC with interest rate is 9

Example4

```
class Company
{   int per;
    void perCriteria()
    {  per=60;
    }
}
class FresherHiring extends Company
{
   void perCriteria()
   {
     per = 50;
   }
}
public class OverridingApp
{  public static void main(String x[])
    {
         FresherHiring f = new FresherHiring();
           f.perCriteria();
    }
}
```

Note: here in Company class we set the perCriteria() as 60% but we customize the freshere criteria with 50%
means we customize the Company class in FreshereHiring class

But some company want to fix the percentage and they are won't give permission to modify its logic in child class then we can declare the parent method as final.

```
    class Company
    {   int per;
final   void perCriteria()
        {  per=60;
        }
    }
    class FresherHiring extends Company
    {
       void perCriteria()
       {
         per = 50;
       }
    }
    public class OverridingApp
    {  public static void main(String x[])
        {
             FresherHiring f = new FresherHiring();
               f.perCriteria();
        }
    }
```

Here compile will generate the error we cannot override the final method in child class and we try to override.

Q. How to avoid the method overriding or how to prevent the method overriding?

If we want to avoid the method overriding we can declare the method as final in parent class.

Final keyword is used for restrict the method overriding means it is used for avoid the method overriding means it is used for parent logic modification from child class.

Final class

If we use the final keyword with class then we cannot inherit the class in any another child class

Final class normally use for create the immutable classes in java.

```
final class A
{

}
class B extends A
{

}
```

Here Compiler will generate the error at compile time because we try to inherit the class A in class B