

Q. What is inheritance? full explanation.

Ans When we construct a new class from existing class in such a way that the new class access all the features & properties of existing class called inheritance.

**Note:-** 1) In java extends keyword is used to form inheritance.

2) It increases reusability.

1) It can access private members

SUBSCRIBE

Note:- I) In java extends keyword is used to perform inheritance.

II) It provides code reusability.

III) We can't access private members of class through inheritance.

IV) A sub class contains all the features of super class so, we should create the object of sub class.

III) We can't access private members of class through inheritance.

IV) A Sub class contains all the features of Super class So, we should Create the object of Sub class.

V) Method overriding only possible through inheritance.

# inheritance

Syntax :- class A

{

+, -

}

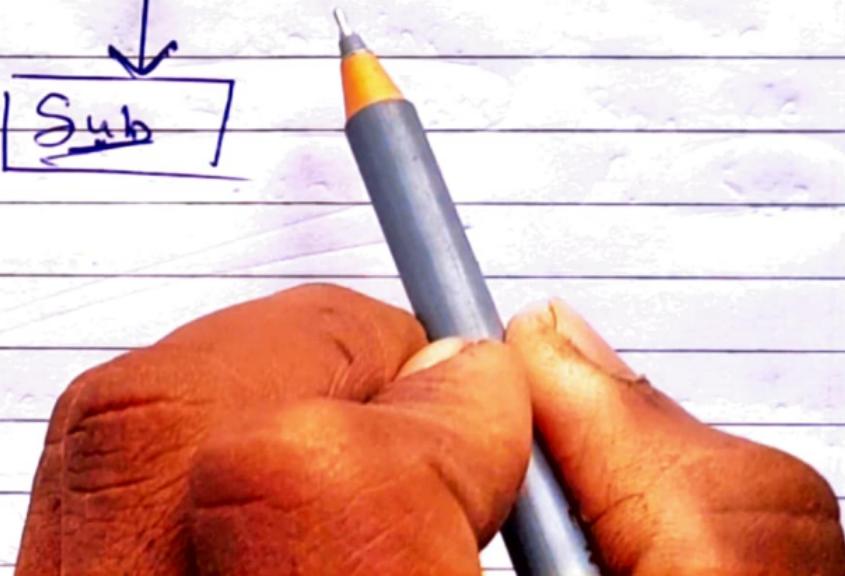
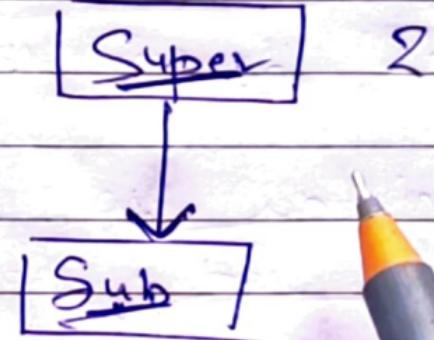
class B extends A

{

}

Type :-

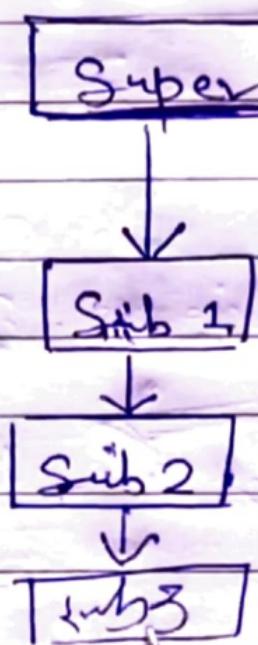
Types :-  $\rightarrow$  Single / Simple inheritance



SUBSCRIBE

# inheritance

ii) Multi-level inheritance :-



Sub 1



Sub 2

III) Multiple inheritance:-

(Super 1)

(Super 2)

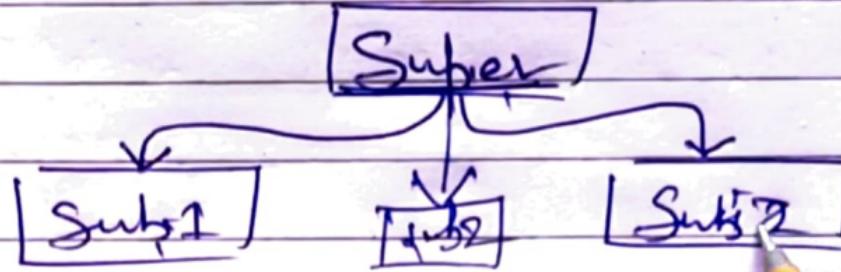
Sub

iv) Hierarchy :-

SUBSCRIBE

Sub'X

iv) hierarchical inheritance :-



# 1. Simple inheritance

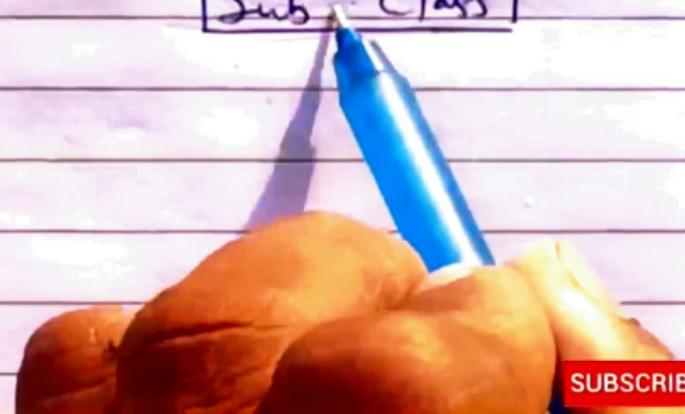
Q. What is simple inheritance? With example.

Ans → Simple inheritance nothing but which contain only one Super class and only one Sub class is called simple inheritance.

Super class

Sub class

Syntax :- class Super  
              S



Syntax:- class Super

}  
S

class Sub extends Super

S

SUBSCRIBE

untitled

```
class student
{
    int roll,marks;
    String name;
    void input()
    {
        System.out.println("Enter roll name & marks: ");
    }
}
```

```
class ankit extends student
```

```
{
    void disp()
```



```
untitled
{
    int roll,marks;
    String name;
    void input()
    {
        System.out.println("Enter roll name & marks: ");
    }
}
class ankit extends student
{
    void disp()
    {
        roll=1; name="ankit"; marks=89;
        System.out.println(roll+" "+name+" "+marks);
    }
}
```

```
untitled
/*
 * Simple Inheritance
 */
class student //super
{
    int roll,marks;
    String name;
    void input()
    {
        System.out.println("Enter roll name & marks: ");
    }
}
class ankit extends student //sub-class
{
    void disp()
    {
        roll=1; name="ankit"; marks=89;
        System.out.println(roll+" "+name+" "+marks);
    }
    public static void main(String[] args) {
        ankit r=new ankit();
```

WIN10\Desktop\ankit.java - Sublime Text (UNREGISTERED)

Selection Find View Goto Tools Project Preferences Help

ankit.java

```
/* Simple Inheritance */
class student //super
{
    int roll,marks;
    String name;
    void input()
    {
        System.out.println("Enter roll name & ma
    }
}
class ankit extends student //sub-class
{
    void disp()
    {
        roll=1; name="ankit"; marks=89;
        System.out.println(roll+" "+name+" "+mar
    }
    public static void main(String[] args) {
        ankit r=new ankit();
        r.input(); r.disp();
    }
}
```

20, Column 30

Type here to search

Select Command Prompt

Microsoft Windows [Version 10.0.18363.1000]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\WIN10>cd desktop

C:\Users\WIN10\Desktop>javac ankit.java

C:\Users\WIN10\Desktop>java ankit
Enter roll name & marks:
1 ankit 89

C:\Users\WIN10\Desktop>

Tab Size: 4

08:54

08-02-2021

ankit.java

```
/* Simple Inheritance */
class student //super
{
    int roll,marks;
    String name;
    private void input()
    {
        System.out.print]
    }
}
class ankit extends student
{
    void disp()
    {
        roll=1; name="ankit";
        System.out.println(roll+" "+name+" "+marks);
    }
    public static void main(String[] args) {
        ankit r=new ankit();
        r.input(); r.disp();
```

Command Prompt

C:\Users\WIN10\Desktop>java ankit

Enter roll name & marks:

1 ankit 89

C:\Users\WIN10\Desktop>javac ankit.java

ankit.java:20: error: cannot find symbol
 r.input(); r.disp();
 ^
symbol: method input()
location: variable r of type ankit

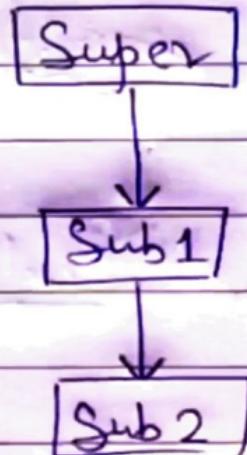
1 error

```
/* Simple Inheritance */  
class student //super  
{  
    protected int roll,marks;  
    String name;  
    protected void input()  
    {  
        System.out.println("Enter roll name & marks: ");  
    }  
}  
class ankit extends student //sub-class  
{  
    void disp()  
    {  
        roll=1; name="ankit"; marks=89;  
        System.out.println(roll+" "+name+" "+marks);  
    }  
    public static void main(String[] args) {  
        ankit r=new ankit();  
        r.input(); r.disp();  
    }  
}
```

## 2. Multilevel inheritance V.V.T

Q. What is multilevel inheritance? With example.

Ans In multi-level inheritance we have only one Super class and multiple Sub classes called multi-level inheritance.



Syntax:- class Super

Syntax:- Class super

Sub 1

}  
S

Class sub1 extends Super

}  
S

}  
S

Class sub2 extends Sub1

}  
S

SUBSCRIBE

```
5     void add()
6     {
7         a=10; b=20;
8         c=a+b;
9         System.out.println("Sum of two Numbers: "+c);
10    }
11    void sub()
12    {
13        a=200; b=100;
14        c=a-b;
15        System.out.println("Sub of two Numbers: "+c);
16    }
17 }
```

```
11     void sub()
12     {
13         a=200; b=100;
14         c=a-b;
15         System.out.println("Sub of two Numbers: "+c);
16     }
17 }
18 class B extends A
19 {
20     void multi()
21     {
22         a=10; b=20;
23         c=a*b;
24         System.out.println("Multip| of two Numbers: "+c);
25 }
```

untitled

```
14         c=a-b;
15         System.out.println("Sub of two Numbers: "+c);
16     }
17 }
18 class B extends A
19 {
20     void multi()
21     {
22         a=10; b=20;
23         c=a*b;
24         System.out.println("Multiplication of two Numbers: "+c)
25     }
26     void div()
27     {
28         a=10; b=2;
```

```
23         c=a*b;
24         System.out.println("Multiplication of two Numbers: "+c);
25     }
26     void div()
27     {
28         a=10; b=2;
29         c=a/b;
30         System.out.println("Division of two Numbers: "+c);
31     }
32 }
```

33 class C extends B //sub2

34 {

35

36 }

37

```
1 // Multi-level Inheritance
2 class A// Super
3 {
4     int a,b,c;
5     void add()
6     {
7         a=10; b=20;
8         c=a+b;
9         System.out.println("Sum of two Numbers: "+c);
10    }
11    void sub()
12    {
13        a=200; b=100;
14        c=a-b;
15        System.out.println("Sub of two Numbers: "+c);
16    }
17 }
18 class B extends A//sub1
19 {
20     void multi()
21     {
22         a=10; b=20;
23         c=a*b;
24         System.out.println("Multiplication of two Numbers: "+c);
25     }
26     void div()
27     {
28         a=10; b=2;
29         c=a/b;
30         System.out.println("Division of two Numbers: "+c);
31     }
32 }
33 class C extends B//sub2
34 {
35     void rem()
36     {
37         a=10; b=3;
38         c=a%b;
39         System.out.println("Remainder of two Numbers: "+c);
40     }
41 }
```

```
29         c=a/b;
30         System.out.println("Division of two Numbers: "+c);
31     }
32 }
33 class C extends B//sub2
34 {
35     void rem()
36     {
37         a=10; b=3;
38         c=a%b;
39         System.out.println("Remainder of two Numbers: "+c);
40     }
41 }
42 class Test
43 {
44     public static void main(String[] args) {
45         C r=new C();
46         r.add(); r.sub(); r.multi(); r.div(); r.rem();
47     }
48 }
49 }
```

C:\Users\WIN10\Desktop\Testjava - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

Test.java

```
1      /* Multi-level Inheritance */
2  class A// Super
3  {
4      int a,b,c;
5      void add()
6      {
7          a=10; b=20;
8          c=a+b;
9          System.out.println("Sum of two Numbers: "+c);
10     }
11     void sub()
12     {
13         a=200; b=100;
14         c=a-b;
15         System.out.println("Sub of two Numbers: "+c);
16     }
17 }
18 class B extends A//sub1
19 {
20     void multi()
21     {
22         a=10; b=20;
23         c=a*b;
```

Select Command Prompt

```
C:\Users\WIN10\Desktop>javac Test.java

C:\Users\WIN10\Desktop>java Test
Sum of two Numbers: 30
Sub of two Numbers: 100
Multiplication of two Numbers: 200
Division of two Numbers: 5
Remainder of two Numbers: 1

C:\Users\WIN10\Desktop>
```

Line 46, Column 33

Type here to search

Tab Size: 4 Java

18:28 08-02-2021

Q. Why Java doesn't support multiple inheritance?

Ans → Whenever a Sub class wants to inherit the property of two or more Super classes that have same method, Java Compiler can't decide which class method it should inherit.

Then there might be a chance of memory duplication. This is the reason Java doesn't support multiple inheritance through classes.

Syntax :-

the property of two or more Super classes  
that have same method, java compiler  
can't decide which class method it should  
inherit.

Then there might be a chance of memory  
duplication i.e. a reason java doesn't  
support multiple inheritance through classes.

Syntax:-

Class A

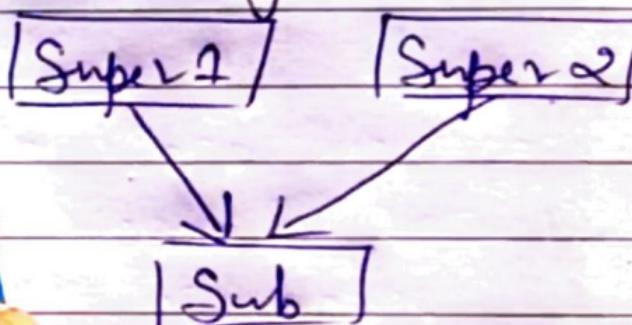
{

Class B

{

void m

{



SUBSCRIBE

that have same method, Java compiler can't decide which class method it should inherit.

Then there might be a chance of memory duplication i.e. a reason Java doesn't support multiple inheritance through classes.

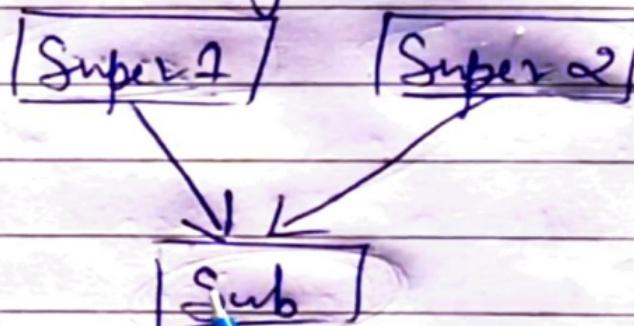
Syntax:-

Class A  
S

void m1()  
S

Class B  
S

void m1()  
S



class A

class B

↓  
Sub

void m1()

void m1()

{

{

}

}

{

{

Class C extends A, B

{

C is in confusion

{

SUBSCRIBE

Q. What is Hierarchical inheritance?

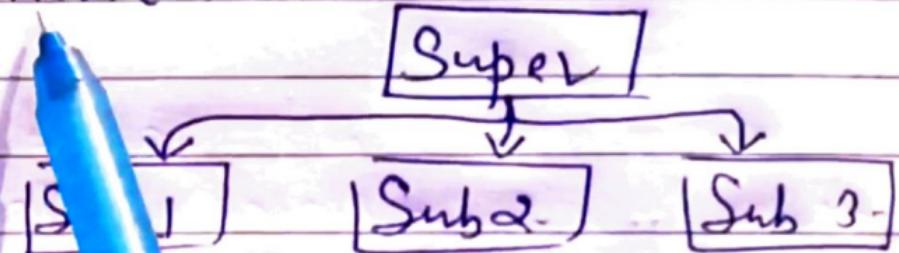
Ans → A inheritance which contain only one Super class and multiple Sub class and all Sub class directly extends Super class Called hierarchical inheritance.

Syntax:-

class A

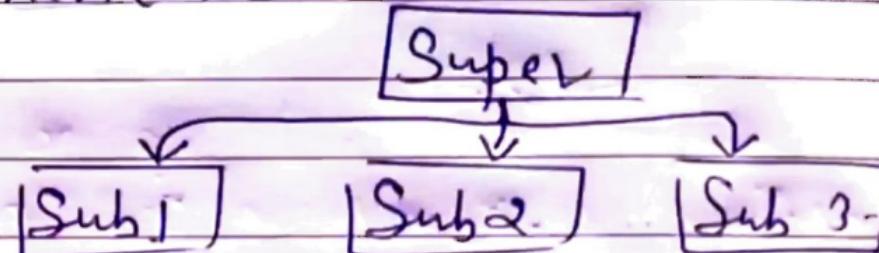
{

}



class and multiple Sub class' and all  
Sub class directly extends Super class Called  
hierarchical inheritance.

Syntax:-



class A

{

}

class B extends A

{

class C extends A

{

## Syntax:-

[Sub 1]

[Sub 2]

[Sub 3]

class A

{

}

class B extends A

{

}

class C extends A

{

}

```
1         /* Hierarchical Inheritance */  
2 class A  
3 {  
4     void input()  
5     {  
6         System.out.println("Enter Your Name: ");  
7     }  
8 }  
9 class B extends A  
10 {  
11     void Show() {  
12     }  
13     System.out.println("My Name is Ankit");  
14 }  
15 }
```



```
1          /* Hierarchical Inheritance */  
2 class A  
3 {  
4     void input()  
5     {  
6         System.out.println("Enter Your Name: ");  
7     }  
8 }  
9 class B extends A  
10 {  
11     void Show()  
12     {  
13         System.out.println("My Name is Ankit");  
14     }  
15 }  
16 class C extends A  
17 {  
18     void Disp()  
19     {  
20         System.out.println("My Name is Ankush");  
21     }  
22 }
```

```
13     System.out.println("My Name is Ankit");
14 }
15 }
16 class C extends A
17 {
18     void Disp()
19     {
20         System.out.println("My Name is Ankush");
21     }
22 }
23 class Demo
24 {
25     public static void main(String[] args) {
26         B r=new B();
27         C r2=new C();
28
29         r.input(); r.Show();
30         r2.input(); r2.Show();
31     }
32 }
```

Command Prompt

```
C:\Users\WIN10\Desktop>javac Demo.java
Demo.java:30: error: cannot find symbol
                      r2.input(); r2.Show
                                         ^
symbol:   method Show()
location: variable r2 of type C
1 error
```

```
C:\Users\WIN10\Desktop>
```



```
10 {
11     void Show()
12     {
13         System.out.println("My Name is Ankit");
14     }
15 }
16 class C extends A
17 {
18     void Disp()
19     {
20         System.out.println("My Name is Ankush");
21     }
22 }
23 class Demo
24 {
25     public static void main(String[] args) {
26         B r=new B();
27         C r2=new C();
28
29         r.input(); r.Show();
```

Select Command Prompt

```
symbol: method Show()
location: variable r2 of type C
1 error
```

```
C:\Users\WIN10\Desktop>javac Demo.java
```

```
C:\Users\WIN10\Desktop>java Demo
Enter Your Name:
My Name is Ankit
Enter Your Name:
My Name is Ankush
```

```
C:\Users\WIN10\Desktop>
```

