AIM: To demonstrate the usage of keyword super and final

THEORY

Keyword super

Whenever a subclass needs to refer to its immediate superclass, it can do so by use of keyword 'super'.

super has two general forms. The first calls the superclass's constructor. The second is used to access a member of the superclass that has been hidden by a member of a subclass.

1) Using super to call Superclase Constructors

A subclass can call a constructor defined by its superclasses by use of the following form of super:

Super (orgs-list);

Here angs-list specifies any arguement needed by the constructor in superclass. super () must always be the first statement executed inside a subclass's constructor.

For eg.

class Test extends Exam 1

Test (int grades, int porcent) 1

Super (grades, prevant);

class Exam &

E int percent, grade;

Exam (inta, int b)

d percent = 10; grade = a; }

3

Here Test calls super 1) with arguments grades and percent. Which intializes the Examil constructor values. Now, since constructors can be overloaded, super () can be used called using any form defined by the superdays. Using super to access member of superclass The 'super' acts like this heyword, expect that it always refers to superclass of subclass in which it is used super member where, member can either be a method or instance variable. This form of super is most applicable to situations in which member names of a subclass hide members by the same name in superclass. For eq class A d int i: class B extends A & int i: B lint q int b) of Farby super. i = a; i=b; } 3 void show d System.out. print. ln (" ; in subclass : " + i); System out println (" i'm super class: " + super.i); 3 class Test d public static void main (8tring args (3) & B ob= new B(2,3); ob. show U;

FOR EDUCATIONAL USE

Sundaram)

Output:

i in subclass : 3

i in superclass: 2

· Regword final

The final keyword has three uses: first, it can be used to create exthe equivalent of a named constant, second is to prevent overriding and third is to prevent inheritance

1) To create an equivalent of a named constant

When a voriable is declared with final keyword, its value can't be modified, thus a constant. This also means that you must intialize a final variable.

For eg

class Test 1

final int CAP = 3;

public static void main (Stringe angul)

2 CAP = 5; 3

3

The above code will to throw a compile time error.

2) To prevent overiding

To disallow a method from being overridden, specify final as a modifier at the start of its declaration. Methods declared as final cannot be overriden

for eq class Ad final and meth() 1 3 System out print to 1" This is last "); class B extends Ad void meth Ud \$ 3 System. out printer (" No"); The above code will throw an compile time error 3) To prevent inheritance By declaring a class as final it prevents it from being inherited. However since an abstract class is incomplete by itself and relies upor its subclaces to provide complete implementations, final and abstract cannot be used together to declare a class For eq final chass A1 int a; vaid show () of system out print In (" Great show"); 3 class B extends A 1 void en () of System out println ("From"); } The above code will show a compile time error.

(Jundaram)

	CONCLUSION:
	Errors encountered:
>	Incorrect call to superclass constructor:
	Child (int a , int b)
	å i=1 - i=2.
	Super (a,b);
0	3
Solution	Call to super must be the first statement in constructor:
	Child (int a, int b)
	2 super (a,b);
	7
	}
- 1	
2)	
0	iz it super(i);
011	
Solution	Correct Syntax: i= i + super.i;

Sundarum

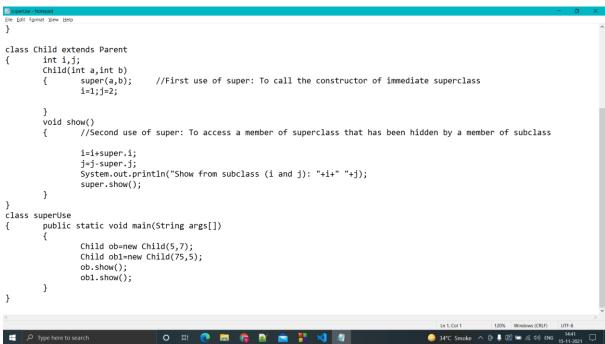
LAB 8: DEMONSTRATE USAGE OF KEYWORD SUPER AND FINAL

Name: Shreyas Sawant Div: D7A Roll No.: 55

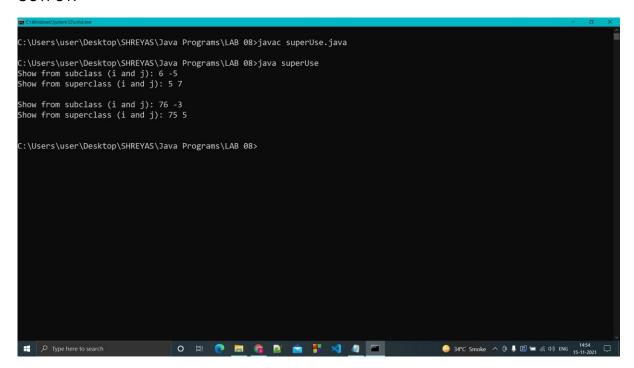
Q.1 Demonstrate any two uses of keyword super.

CODE:

```
public int i,j;
        Parent(int a,int b)
                i=a;
                j=b;
        void show()
                System.out.println("Show from superclass (i and j): "+i+" "+j+"\n");
        }
class Child extends Parent
        int i,j;
        Child(int a,int b)
                super(a,b);
                                //First use of super: To call the constructor of immediate superclass
                i=1;j=2;
        void show()
                //Second use of super: To access a member of superclass that has been hidden by a member of subclass
                i=i+super.i;
                j=j-super.j;
                System.out.println("Show from subclass (i and j): "+i+" "+j);
                super.show();
        }
                                                                                             ු 34°C Smoke ∧ ලි 🌡 😭 ≔ /ය අා) ENG 15-11-2021
```



OUTPUT:



Q.2 Demonstrate the importance of keyword final in inheritance and method overriding.

CODE:

```
class One
       //Methods declared as final cannot be overriden
       final public void skill()
               System.out.println("Skills are required to access this." );
}
//classes declared as final cannot be inherited
final class Two extends One
       //Show error as method cannot be overriden
       void skill()
       {
               System.out.println("Cannot access");
}
class Three extends Two
{
        void skill()
               System.out.println("Cannot access it.");
       {
               super.skill();
                                                                                 Type here to search
       void skill()
               System.out.println("Cannot access");
}
class Three extends Two
       void skill()
               System.out.println("Cannot access it.");
               super.skill();
//final
class finalUse
        public static void main(String args[])
               Three ob=new Three();
               ob.skill();
```

OUTPUT

