This Keyword-

It is used to refer current class variables, method and constructor.

Note- this keyword is not used in static context

Why?

# Program-1 use of current class variable.

```
class Parent {

	int x = 20;
}
class Child extends Parent {

	int x = 25;

	void test() {
		int x = 30;
		Child c = new Child();
		//In Scenario 1
		System.out.println("By creating objects=" + c.x);
		//In Scenario 2
		System.out.println("By using this keyword=" + 
this.x);
	}
}
public class TestMain {

	public static void main(String[] args) {

		Child c= new Child();
		c.test();
	}
}
```

Output-

```
By creating objects=25
By using this keyword=25
```

In Scenario 1, we are calling the x variable of child class for that purpose we loading the whole class that is not good programmer approach. For use of single variable, we should go for this keyword in java.

In Scenario 2, we are trying to calling the x variable of current class by using the keyword that is the best approach because we are not wasting the memory here.

# Program 2-

2.1 program for use of current class method.

```
class Parent {

    void test() {
        System.out.println("Parent class method.");
    }

}
class Child extends Parent {

    void test() {
        System.out.println("Child class method");
    }

    void demo() {
        this.test();
    }

}
public class TestMain {

    public static void main(String[] args) {

        Child c= new Child();
        c.demo();
    }
}
```

2.2

Output-

```
Child class method.
```

Super Keyword-

It is used to refer immediate parent class object, method and constructor.

Why?

# **Program-1** for use of immediate parent class objects.

```
public class Parent {

        int x= 20;

}

class Child extends Parent {

        int x = 25;

        public void test() {
                int x= 30;

                //Scenario-1
                Parent p = new Parent();
                System.out.println("Parent class x variable=" + p. x);

                //Scenario-2
                System.out.println("Immediate super class of child class x variable" +
super. x);
        }
}


public class TestMain {

        public static void main(String[] args) {

                Child c= new Child();
                c.test();
        }
}
```

Output-

```
Parent class x variable=20
Immediate super class of child class x variable20
```

In Scenario 1, we are calling the roll no of parent class for that purpose we loading the whole class that is not good programmer approach. For use of single variable, we should go for super keyword in java.

In Scenario 2, we are trying to print the roll no of immediate super class by using the super keyword that is the best approach because we are not wasting the memory here.

# Program 2- program for use of immediate super class method.

```
class Parent {

    void test() {
        System.out.println("Parent class method.");
    }

}

class Child extends Parent {

    void test() {
        super.test();
    }

}

public class TestMain {

    public static void main(String[] args) {

        Child c= new Child();
        c.test();
    }
}
```

Output-

```
Parent class method.
```

Difference between super(),this() and super, this keyword:

super() this()	**super, this**
It is used for calling constructors	It is used to call method and variables
Using this, we can call immediate parent class and current class constructors	Using this, we can call immediate parent class and current class variables and methods.
It can be used only inside constructors	It can be used within instance(non-static) area only.