Difference between

1. **this() and super() constructor >> used for constructor call**
2. **this and super keyword >> used for variable call**

—----------

1. **This() : this() is used to call same class constructor**

E.x.

```
package testJava;
public class aclass
{
	aclass()                                // no parameter constructor
	{
		System.out.println("no argument constructor");
	}

	aclass(int i)                           // single parameter constructor
	{
		this();
		System.out.println("parameterised constructor");
	}

	public static void main(String[] args)
	{
		aclass ac = new aclass(1);
	}
}
```

Output : no argument constructor
parameterised constructor

—----------

2. **Super() : super() is used to call super class constructor**

E.x.

Super class >>

```
package testJava;
public class aclass
{
	aclass()                                // no parameter constructor
	{
		System.out.println("no argument constructor from super class");
	}

}
```

Sub class >>

```
package testJava;
public class sclass extends aclass
{
	sclass()
	{
		super();
		System.out.println("no argument constructor from sub class");
	}
	public static void main(String[] args) {

		sclass sc = new sclass();
	}
}
```

Output : no argument constructor from super class
no argument constructor from sub class

==========

There are few cases which are showing the rules for using super() and this()

Case 1 : using “super()” inside constructor after execution statement

```
public class aclass
{
	aclass()                                // Constructor
	{
		System.out.println("constructor");
		super();
	}
}
```

Output : compile time error
Note : Take the super() only in first line
——————

Case 2 : using “this()” inside constructor after execution statement

```
public class aclass
{
	aclass()                                // Constructor
	{
		System.out.println("constructor");
		this();
	}
}
```

Output : compile time error
Note : Take the this() in first line
——————

Case 3 : using “super()” and ‘this()” inside constructor after/before execution statement

```
public class aclass
{
	aclass()                                    // Constructor
	{
		System.out.println("constructor");
		super();
		this();
	}
}
```

Output : compile time error
Note : We can not use super() and this() combinly.
—----------
Case 4 : using super() and this() inside method

```
package testJava;

public class aclass
{
	public void Test()                          // method
	{
		super();
		System.out.println("method");
	}
}
```

Output : compile time error
Note : We can use super() and this() only inside constructor

==========

2. **this and super keyword >> used for instance variable**

this : this is used to call same class instance variable

```
package Testing_Package;

public class NewClass
{
	int a = 30;
}
```

—----------

super : super is used to call super class instance variable

```
package Testing_Package;

public class Javatesting extends NewClass
{
	int a = 10;

	public void method1()
	{
		int a = 20;
		System.out.println(a);		// called local variable
		System.out.println(this.a);	// called instance variable from current class
		System.out.println(super.a);	// called instance variable from super class
	}

	public static void main(String[] args)
	{
		Javatesting ref = new Javatesting();
		ref.method1();
	}
}
```

Output : 20
10
30