<https://www.youtube.com/watch?v=XtrSoiFfTEQ>

Constructor means it is a class entity which is used to define some class features while creating the object. <https://www.youtube.com/watch?v=XtrSoiFfTEQ>

Constructor is used to create an object.

It is similar to a function but not a function.

It cannot return any value.

Constructor name should be same as the class name.

Constructor has no static or any return type, no datatype or anything in the signature just the name (which is same as the class name).

Default constructor is a constructor without any parameters.

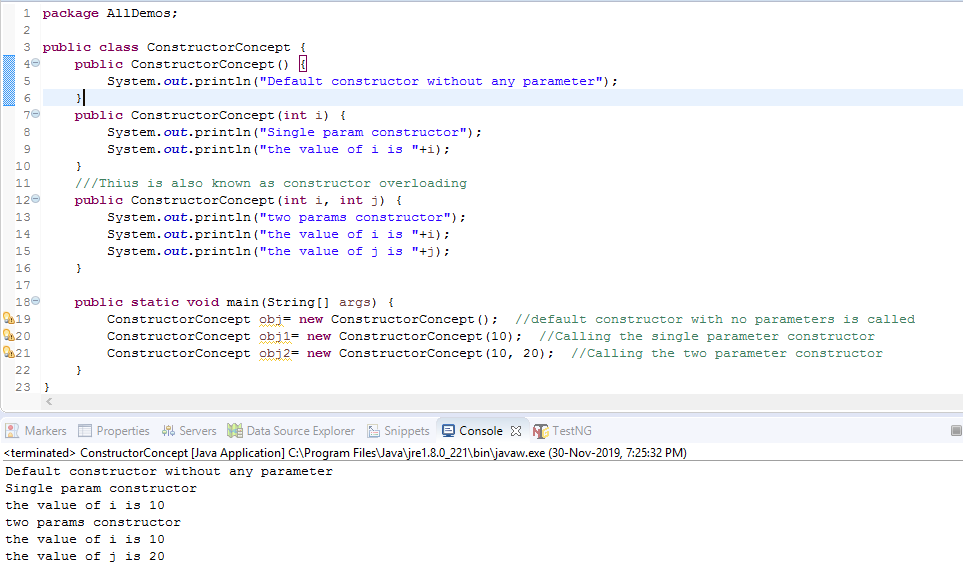
Constructor can be **overloaded** (same name with different datatypes, parameters, and order)

**No it is not possible to OVERRIDE a constructor.**

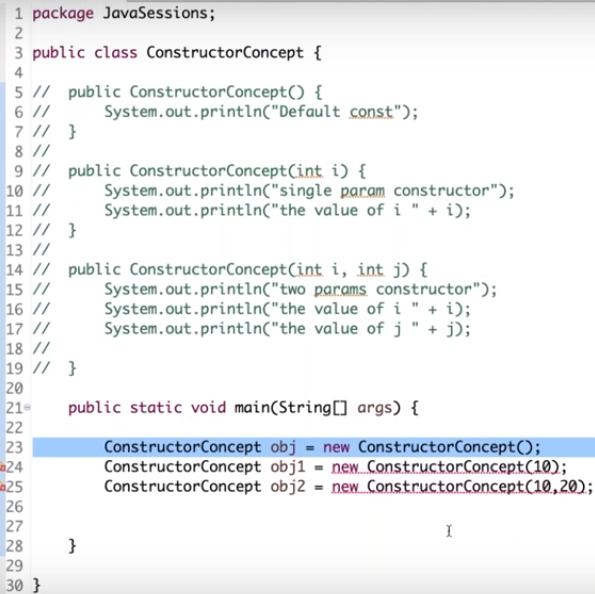
How to call a constructor?

Constructor will be called immediately when an object of the class is created.

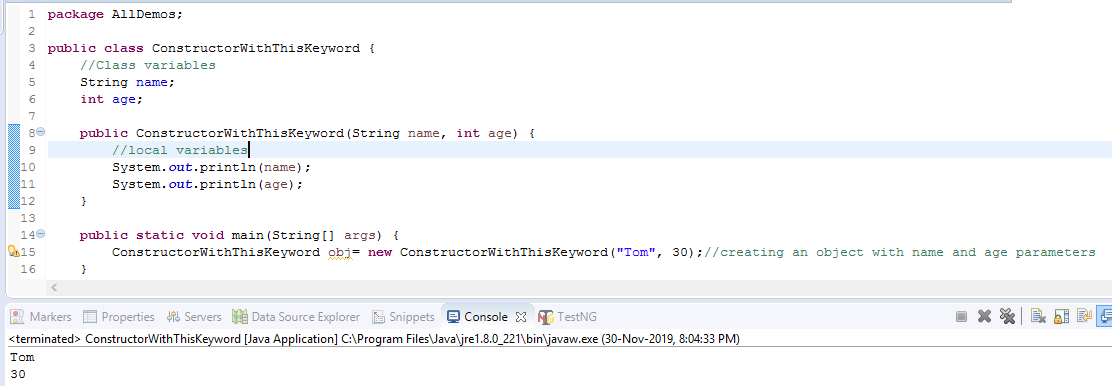
An object of the class is created in the main method using the new keyword



Now in the below case, when we comment all the constructors, still there is no error for the default constructor, because while creating the class a default, hidden constructor is already created, whereas for the single param and double param constructor, we have to manually create the constructors



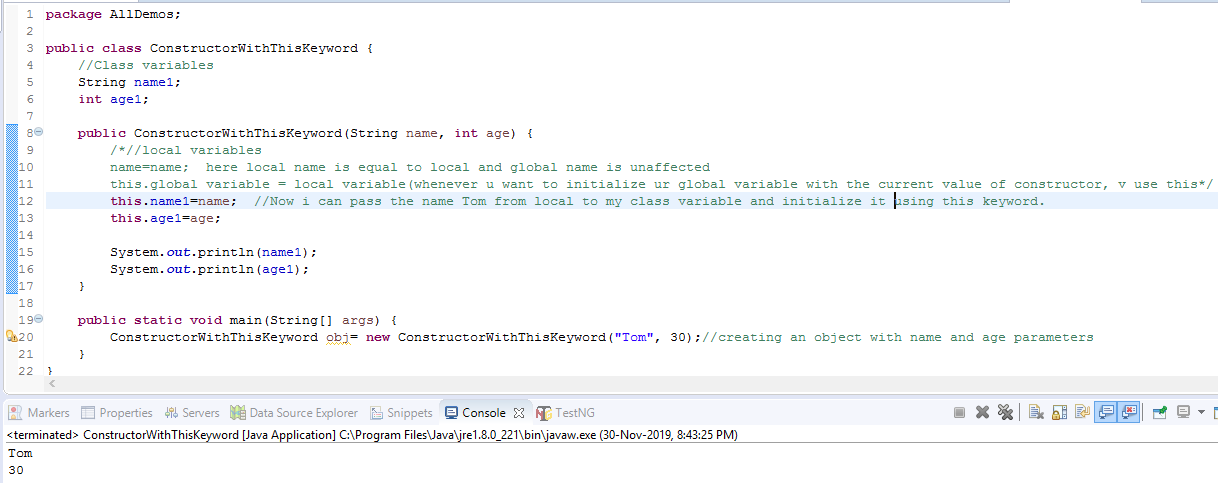
But when we run the program with the constructors; obj1 and obj2 commented, the obj constructor program will just run without any output

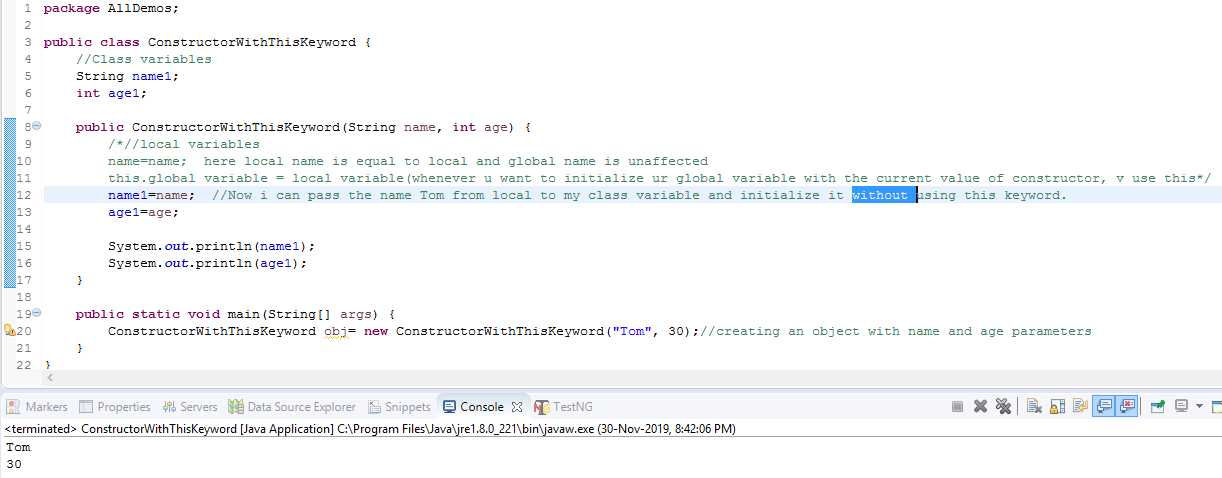


Now when Tom and 30 are passed as arguments, the local variables will be initialized,

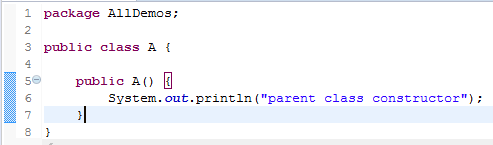
THIS Keyword is used to initialize the class/global variables.

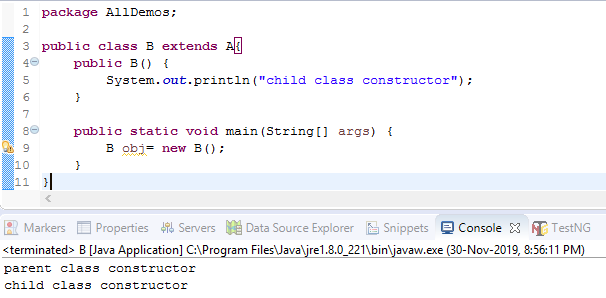
If I want the class/global variables; name and age to be initialized, then I have to use the, **this** keyword

Another way to initialize global/class variables without using the, this keyword



Create a class A and a constructor within it

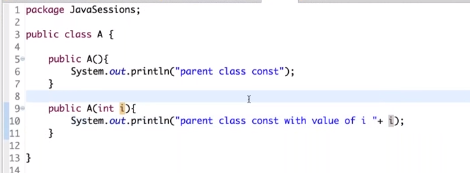


and similarly create another class B which inherits A and run the program as below 

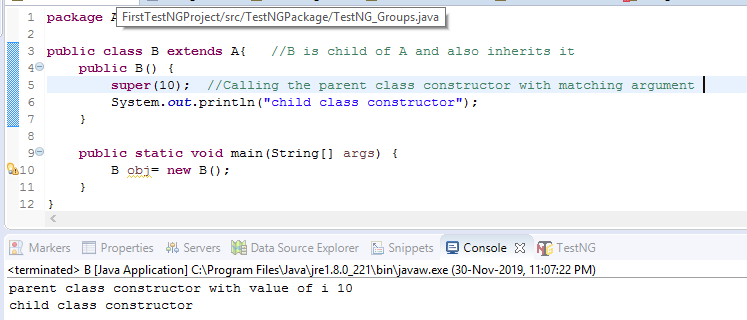
The **super keyword** refers to superclass (parent) objects. It is **used** to call superclass **methods**, and to access the superclass constructor. The most common **use** of the **super keyword** is to eliminate the confusion between superclasses and subclasses that have **methods** with the same name.

Now to understand the concept of Super() keyword. The Super() keyword is used to call parent class constructor with matching arguments.

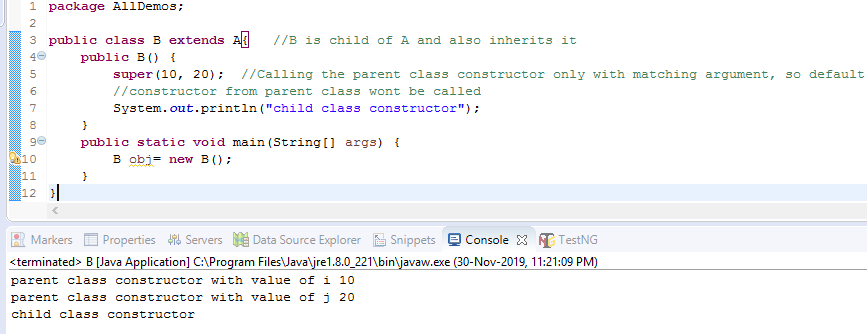
Lets create a class A as below; here v created a 2nd constructor **with int i as** the argument



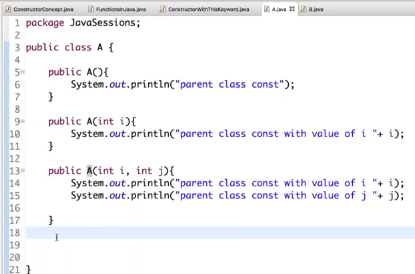
Now if I want to call the parent class constructor A(int i) from B Class, we use the Super keyword

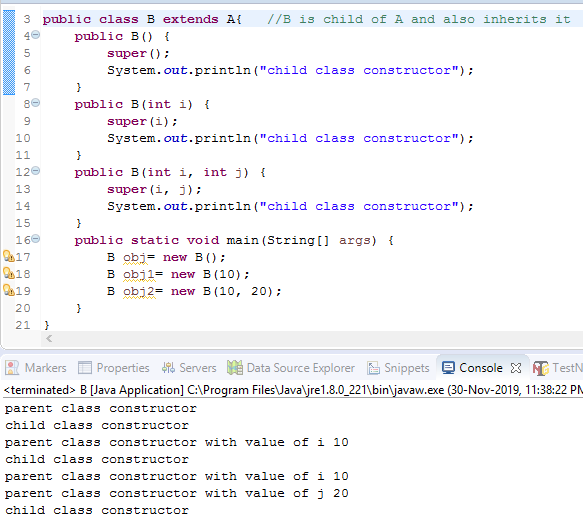


Super keyword is used to call the parent class constructor with matching arguments. Super keyword should always be written as the first statement in the child class constructor. Since it is always the first statement, there can be no 2nd Super() keyword in the child class.



Now suppose I want to call all the constructors from Class A, then the below code will be used





We cannot have more than one Super() keyword in a constructor and it is always the first statement.

Pending from 49:03 onwards