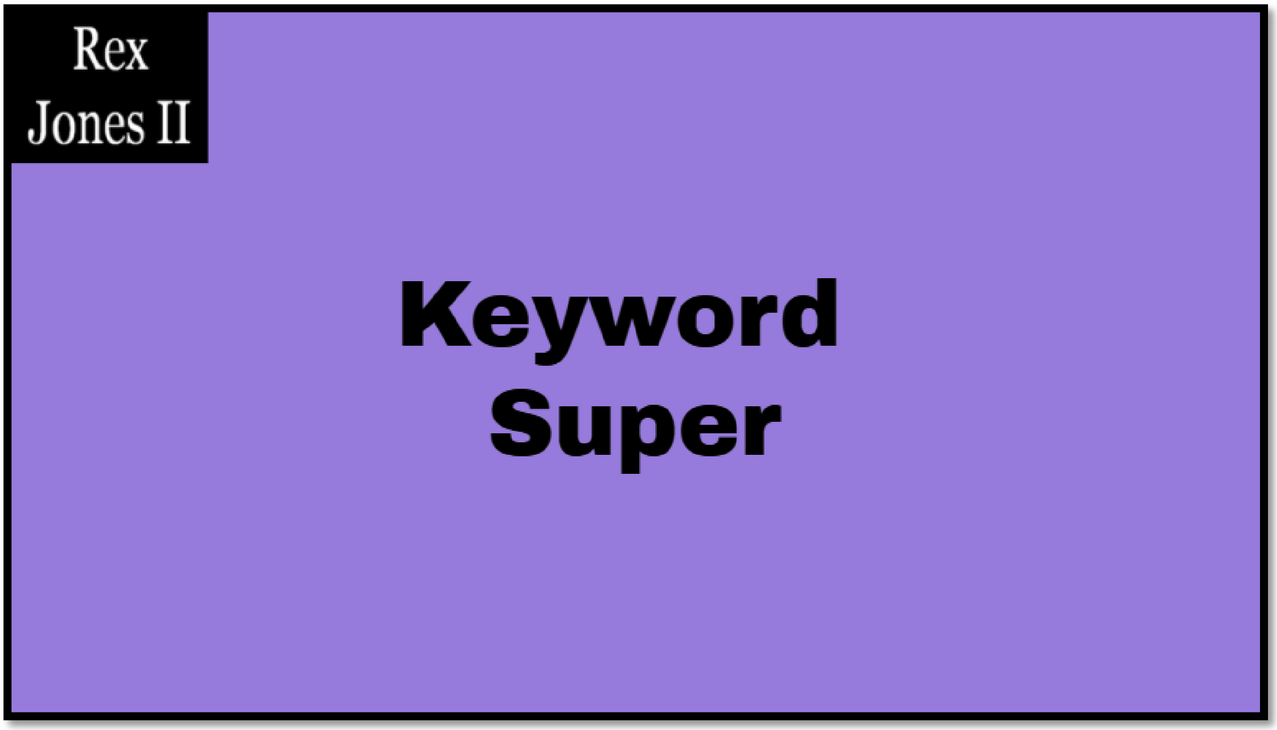
Rex
Jones II

(Transcript)
super Keyword



Rex
Jones II

Keyword
Super

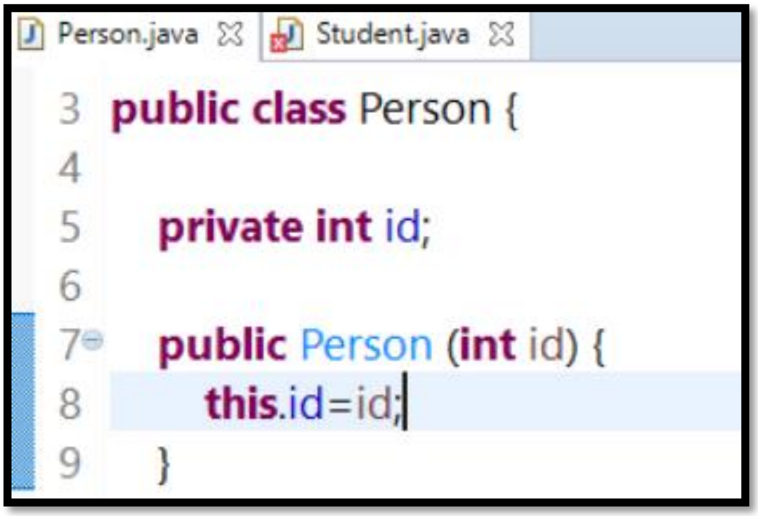
The keyword super refers to the superclass. It allows us to access the superclass variables, methods, and constructors. Here's a scenario where the superclass is inherited by the subclass. I like to say parent class and child class. The parent class does not have a defined constructor. Therefore, a default constructor is provided automatically by Java. You can watch my Constructor video for more information on Constructors.

If we look at the child class, it also does not have a defined constructor. A Constructor Rule states a subclass constructor must invoke the superclass constructor whether the superclass constructor is implicit or explicit. That means if there is a constructor in the parent class then the child class must call that constructor in the parent class. In this scenario, both the parent and child classes have implicit constructors because they are not defined.

Now, look what happens when I add a defined constructor to the parent class.

```
Public Person (int id) {
  this.id = id;
```

}



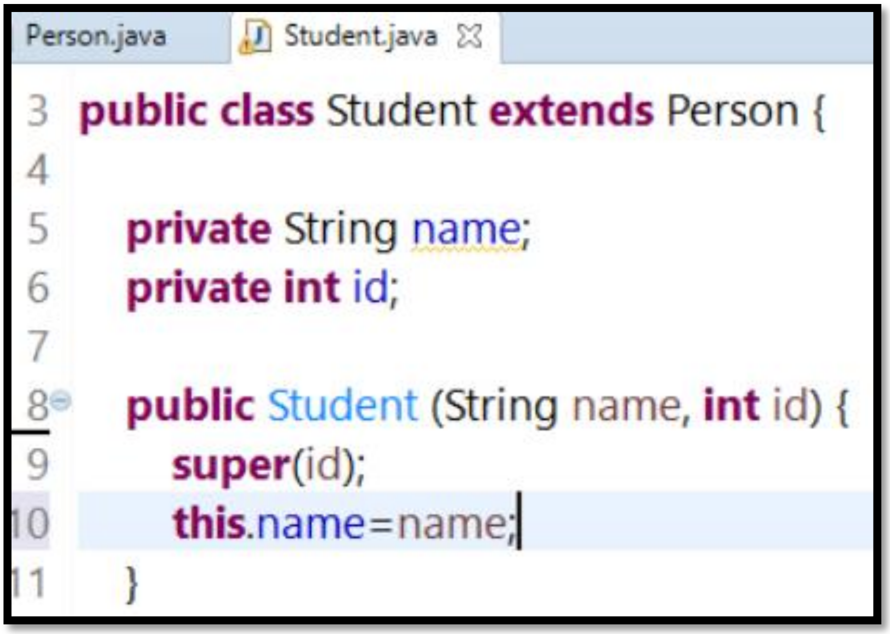
```
public class Person {

    private int id;

    public Person (int id) {
        this.id=id;
    }
```

Save and there’s an error message in the child class. The error message says “Implicit super constructor Person() is undefined for default constructor. Must define an explicit constructor”. That means the child class must have a defined constructor because the parent class has a defined constructor.

In the child class, let’s write public Student (String name, int id) {

}

Now, we have different compiler error. The error states “Implicit super constructor Person() is undefined. Must explicitly invoke another constructor” This is where the keyword super comes into play. We must invoke the superclass constructor using the super keyword. So, we write super, CTRL + SPACE, and we see a parameter for (int id) and select that option and the error goes away. If I write, this.name = name before super(id). We have a different error. Hover super(id) and the error states “Constructor call must be the first statement in a constructor”. Whenever we call a constructor, it must be the first statement. So we add super (id) before this.name = name.



```
public class Student extends Person {

    private String name;
    private int id;

    public Student (String name, int id) {
        super(id);
        this.name=name;
    }
```

There are 2 ways to use the super keyword. This is 1 way where we call the superclass constructor. The 2nd way is to access members of a superclass that has been hidden by the subclass. Now, we see 2 methods: getter and setters. The getter method - getID() and setter method – setID() in the subclass hide the getter and setter methods in the superclass.

Let me show you how to access the superclass getter and setter methods. Go back to the child class and write a method.

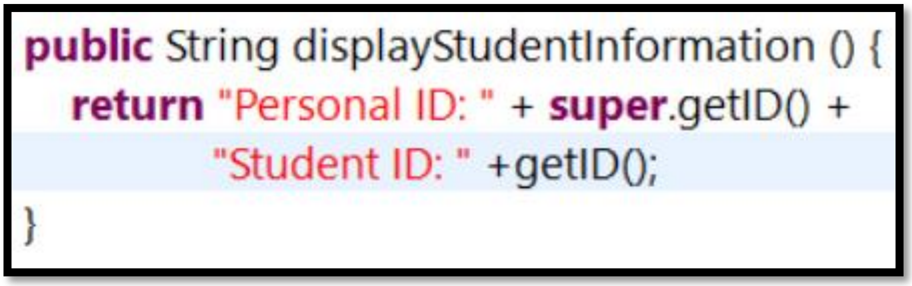
```
public String displayStudentInformation () {
  return “Personal ID: ” +

}
```

getID. Do you see Student? This will call the subclass Student method. We want the superclass Person method. Therefore, we write super dot and now we see getID for the superclass Person method.

```
public String displayStudentInformation () {
  return “Personal ID: ” + super.getID() +

}
```

We can also add a method from the sub class/child class. This example, can have Personal ID and Student ID.

```
public String displayStudentInformation () {
  return “Personal ID: ” + super.getID() +
           “Student ID: ” + this.id;
}
```



```
public String displayStudentInformation () {
    return "Personal ID: " + super.getID() +
             "Student ID: " +getID();
}
```

A superclass variable operates just like a superclass method. However, the variable and method must be a public member. That’s it for the super keyword. You can follow me on Twitter, connect with me on LinkedIn, and subscribe to my YouTube channel. See you next time

Social Media Contact

- YouTube https://www.youtube.com/c/RexJonesII/videos
- Twitter https://twitter.com/RexJonesII
- LinkedIn https://www.linkedin.com/in/rexjones34/
- GitHub https://github.com/RexJonesII/Free-Videos
- Facebook http://facebook.com/JonesRexII