

3.2 Super Keyword in Java

- The super keyword in java is a reference variable that is used to refer parent class objects.
- The keyword “super” came into the picture with the concept of Inheritance.
- It is majorly used in the following contexts:

1. Use of super with variables:

- This scenario occurs when a derived class and base class has same data members.
- In that case there is a possibility of ambiguity for the JVM. We can understand it more clearly using this code snippet:

```
/* Base class vehicle */
class Vehicle
{
    int maxSpeed = 120;
}
/* sub class Car extending vehicle */
class Car extends Vehicle
{
    int maxSpeed = 180;
    void display()
    {
        /* print maxSpeed of base class (vehicle) */
        System.out.println("Maximum Speed: " + super.maxSpeed);
    }
}
/* Driver program to test */
class Test
{
    public static void main(String[] args)
```

```
{
    Car small = new Car();
    small.display();
}
}
```

Output:

Maximum Speed: 120

In the above example, both base class and subclass have a member maxSpeed.

We could access maxSpeed of base class in subclass using super keyword.

2. Use of super with methods:

- This is used when we want to call parent class method.
- So whenever a parent and child class have same named methods then to resolve ambiguity we use super keyword.
- This code snippet helps to understand the said usage of super keyword.

```
class Person /* Base class Person */
{
    void message()
    {
        System.out.println("This is person class");
    }
}
class Student extends Person /* Subclass Student */
{
    void message()
    {
        System.out.println("This is student class");
    }
    void display() // Note that display() is only in Student class
```

```

    {
        message();           // will invoke or call current class message()
method
        super.message();    // will invoke or call parent class message()
method
    }
}

class Test    /* Driver program to test */
{
    public static void main(String args[])
    {
        Student s = new Student();
        s.display();    // calling display() of Student
    }
}

```

Output:

This is student class

This is person class

In the above example, we have seen that if we only call method `message()` then, the current class `message()` is invoked but with the use of `super` keyword, `message()` of superclass could also be invoked.

3. Use of super with constructors:

- `super` keyword can also be used to access the parent class constructor.
- One more important thing is that, ‘`super`’ can call both parametric as well as non parametric constructors depending upon the situation.
- Following is the code snippet to explain the above concept:

```
class Person          /* superclass Person */
{
    Person()
    {
        System.out.println("Person class Constructor");
    }
}

class Student extends Person /* subclass Student extending the
Person class */

{
    Student()
    {
        super();      // invoke or call parent class constructor
        System.out.println("Student class Constructor");
    }
}

class Test          /* Driver program to test*/
{
    public static void main(String[] args)
    {
        Student s = new Student();
    }
}
```

Output:

Person class Constructor
Student class Constructor

In the above example we have called the superclass constructor using keyword ‘super’ via subclass constructor.

3.2.1 Other Important points:

- Call to `super()` must be first statement in `Derived(Student)` Class constructor.
- If a constructor does not explicitly invoke a superclass constructor, the Java compiler automatically inserts a call to the no-argument constructor of the superclass. If the superclass does not have a no-argument constructor, you will get a compile-time error. Object *does* have such a constructor, so if Object is the only superclass, there is no problem.
- If a subclass constructor invokes a constructor of its superclass, either explicitly or implicitly, you might think that a whole chain of constructors called, all the way back to the constructor of Object. This, in fact, is the case. It is called *constructor chaining*.