IS A RELATIONSHIP

It is also known as inheritance

Advantage : code reusability

By using extends keyword we can implement IS A Relationship

**class** P {

**public void** p1() {}

}

**class** C **extends** P {

**public void** c1() {}

}

Very important conclusion

P p = **new** C();

p.p1();

*//p.c1(); we can't*

C c = **new** C();

c.c1();

c.p1();

Conclusion 1: Whatever methods parent has by default available to the child and Hence on the child reference we can call both parent and child class methods.

Conclusion 2: Whatever methods child has by default not available to parent, and hence on the parent reference we can’t call child specific methods

Conclusion 3: Parent reference can be used to hold child object but by using that reference we can’t call child specific methods. But we can call the methods present in parent class.

Conclusion 4: Parent reference can be used to hold child object but child reference can’t be used to hold parent object.

**Multiple Inheritance**

A Java class can’t extend more than one class at a time, hence java won’t provide support for multiple inheritance in class.

class A extends B, C {} //invalid

But interface can extends any number of interfaces simultaneously, hence java provide support for multiple inheritance with respect to interface.

Interface A extends B, C { } //valid

Why ambiguity will not occur: Even though multiple method declarations are available but implementation is unique and hence there is no chance of ambiguity problem in interfaces.

Strictly speaking through interface we wont get any inheritance.

**Cyclic Inheritance:**

Cyclic inheritance is not allowed in java. Of course it is not required.

**class** C **extends** C {} //invalid