

What is Inheritance?

Solidity's inheritance feature enables programmers to extend contractor attributes and properties to derived contracts, allowing developers to modify them through overriding.

This sets Solidity apart from Java, as it allows multiple inheritances, allowing a derived contract to have multiple parent contracts simultaneously. This allows a single contract to inherit from multiple contracts.

Key Points:

1. A derived contract can access all non-private members including state variables and internal methods. But using this is not allowed.
2. Function overriding is allowed provided function signature remains the same. In case of the difference of output parameters, the compilation will fail.
3. We can call a super contract's function using a super keyword or using a super contract name.
4. In the case of multiple inheritances, function calls using super gives preference to most derived contracts.



Using the “is” Keyword in Solidity

To create a derived (or inheriting) contract, simply use the is keyword, as demonstrated in the example code below:

```
# A is a derived contract of B
contract A is B{
    //Code
}
```

As mentioned earlier, Solidity allows for multiple inheritances. You can implement multiple inheritances in solidity as shown in this sample code:

```
contract A{
    //Code
}

# single inheritance
contract B is A{
    //Code
}

#multiple inheritance
contract C is A,B {
    //Code
}
```



The implementation above has been carefully chosen to demonstrate a particularly interesting case of multiple inheritances in Solidity. Take note that one of the contracts that C is deriving from is also a derived contract. That is, contract B is also derived from contract A.

This is not an error – Solidity allows this type of multiple inheritances as well, and your code should compile without any errors.

Types of Inheritance in Solidity

1. **Single Inheritance:** In Single or single level inheritance the functions and variables of one base contract are inherited to only one derived contract.
2. **Multi-level Inheritance:** It is very similar to single inheritance, but the difference is that it has levels of the relationship between the parent and the child. The child contract derived from a parent also acts as a parent for the contract which is derived from it.
3. **Hierarchical Inheritance:** In Hierarchical inheritance, a parent contract has more than one child contracts. It is mostly used when a common functionality is to be used in different places.
4. **Multiple Inheritance:** In Multiple Inheritance, a single contract can be inherited from many contracts. A parent contract can have more than one child while a child contract can have more than one parent.



YouTube Link:



<https://www.youtube.com/embed/aEL8IpsCB10>

Solidity Source File

Single Inheritance

{CODE}

Multi-level Inheritance

{CODE}

Hierarchical Inheritance

{CODE}

Multiple Inheritance

{CODE}

