

Inheritance

- **A contract can inherit from other contracts** which are known as **base contracts**;
- When a contract inherits from multiple contracts, **only a single contract is created** on the blockchain, and the code from all the base contracts is copied into the created contract;
- General inheritance system is very similar to **Python's**, especially concerning multiple inheritance.
- Solidity supports multiple inheritance including **polymorphism**. Multiple inheritance introduces problems like the **"diamond problem"**;
- All function calls are virtual, which means that the most derived function is called, except when the contract name is explicitly given;
- When deploying a derived contract the base contract's constructor is automatically called;
- **is** keyword is used when declaring a new derived contract;

Abstract contracts & Interfaces

- An **abstract contract** has at least a function that lacks implementation;
- **An abstract contract cannot be deployed;**
- Setting a contract **constructor as internal** makes that an abstract contract;
- **Interfaces** are similar to abstract contracts, but they cannot have any functions implemented;
- **Interfaces can be inherited;**
- Interfaces have further restrictions:
 - Cannot inherit other contracts or interfaces
 - Cannot define constructor
 - Cannot define variables
 - Cannot define structs
 - Cannot define enums
- Contracts are allowed to inherit interfaces from other contracts;
- An interface is created using the **interface** keyword instead of contract;