

Abstract Base Classes

In this lesson, we'll study abstract base classes.

WE'LL COVER THE FOLLOWING ^

- Inheritance
- Abstract base classes
 - Rules

Inheritance

When one class inherits from another class, it gets all its attributes and members from the parent class.

- During inheritance, the access specifier of the inherited class and the access specifier of the inheritance must be considered.
- The inherited classes use the attributes and methods from the parent class and can add new ones.
- Inheritance is the base for polymorphism. One of the three key characteristics of object orientation.

Polymorphism: when the characteristics of an object behave differently at run time.

Abstract base classes

These are classes that have at least one pure virtual method. By declaring a method virtual and adding `= 0` to the method declaration, we can make a method pure virtual.

```
class Account{
```

```
class Account{  
    public:  
        virtual double estimateReturn() = 0;  
    ...  
}
```

Abstract Base Classes are typically used as interfaces for class hierarchies because they determine the implementation of the derived classes.

Rules

1. A class that has a pure virtual method cannot be instantiated.
2. A derived class has to implement the pure virtual method in order to get instantiated.
3. A pure virtual method can be implemented in a class.
4. We must implement a pure virtual destructor. Idiom in C++ is used to declare an abstract base class.

In the next lesson, we'll study the access rights of inheritance that are available to classes.