Advantages of Inheritance

In this lesson, you'll get to know about the advantages of using Inheritance.
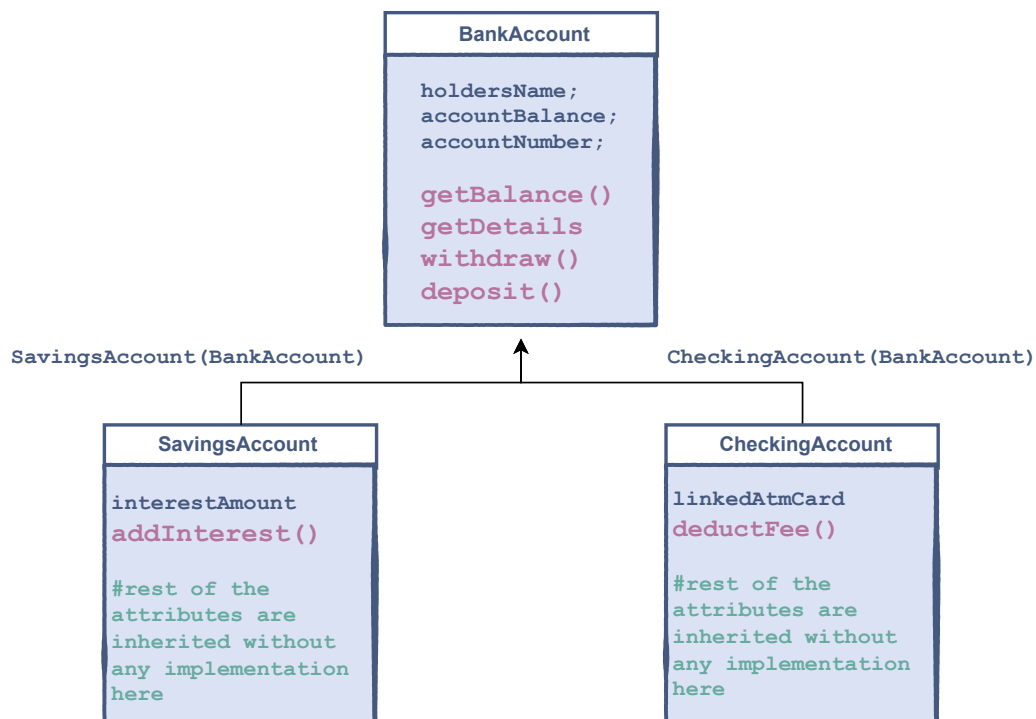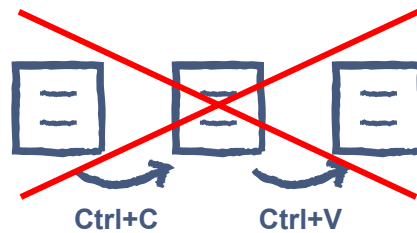
Re-usability

Inheritance makes the code reusable. Consider that you are up for designing a banking system using classes. Your model might have these:

- A parent class: `BankAccount`
- A child class: `SavingsAccount`
- Another child class: `CheckingAccount`

In the above example, you **don't need to duplicate** the code for the `deposit()` and `withdraw()` methods inside the *child* classes, namely `SavingsAccount` and `CheckingAccount`. In this way, you can avoid the duplication of code.
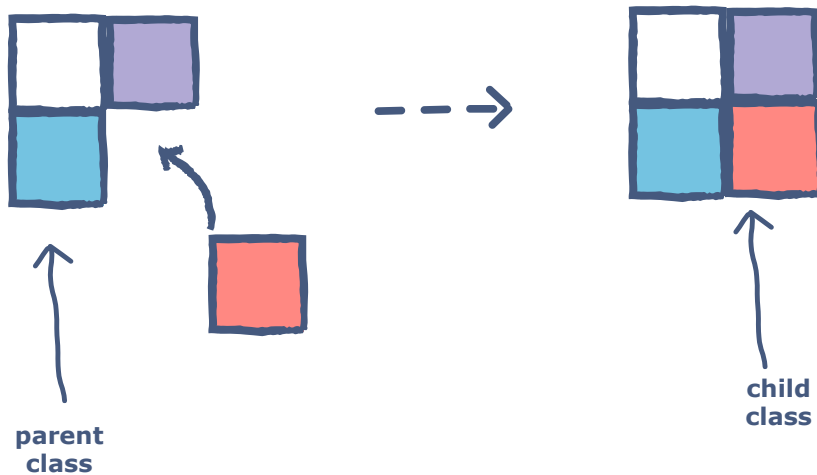

Ctrl+C   Ctrl+V

Code Modification

Suppose you put the same code in different classes all over the place, but what happens when you have to make a change to a function? You have to make changes in several places. Chances are, you forget some places and bugs are introduced. With inheritance, all changes will be localized, and inconsistencies will be avoided.

Extensibility

Using inheritance, one can extend the base class as per the requirements of the derived class. It provides an easy way to upgrade or enhance specific parts of a product without changing the core attributes. An existing class can act as a base class from which a new class with upgraded features can be derived.
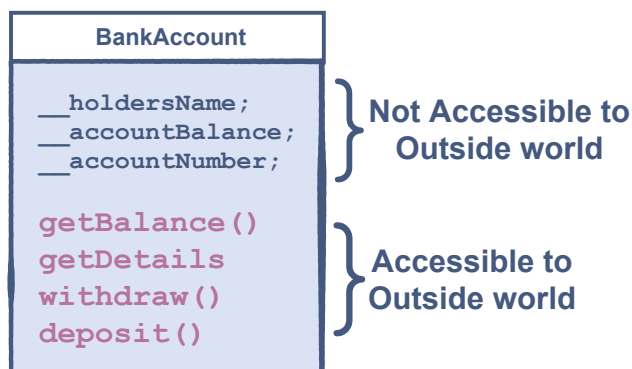
In the example above, you realize at a later point that you have to diversify this banking application by adding another class for `MoneyMarketAccount`. So, rather than implementing this class from scratch you can extend it from the existing `BankAccount` class as a starting point. You can also reuse its attributes that are common with `MoneyMarketAccount`.

Child class can add on to the Parent class

Data Hiding

The base class can keep some data private so that the derived class cannot alter it. This concept, encapsulation, has already been discussed in the previous chapter.

So these were some of the key advantages of inheritance. In the next lesson, attempt a quick quiz to test your understanding of the concepts related to inheritance.