|  |
| --- |
| Circle Language Spec: Inheritance |

## Inheritance Introduction

This documentation folder covers the concept of inheritance. This concept has been present inside object oriented programming languages for decades. But the new computer language takes it a little bit further. In other languages one class can inherit from another class, taking over all its characteristics before the derived class gets extended or altered. Inheritance goes further than that in the new computer language. Next to *class inheritance* (the traditional form of inheritance), there is also *object inheritance*. This means, that one object takes over the characteristics of a specific other object. In fact, the other object melts together with the new object. When you change base members of the new object, the base object’s data also changes. This is callled *object inheritance*. There are even more forms of inheritance. A collection can be inherited from, so that a whole existing table of items can be extended with new columns. This is called *list inheritance*. A whole *system* can also inherit from another system, taking over all of its characteristics, and the derived system can be altered, augmented, specialized and extended. This is called *system inheritance*. If the original system gets changed, this may also change the derived systems. So basically: everything can be inherited from and this can render specialized versions of any class, object, collection or module.

Such specialization can be also be applied to *commands*. The inheritancedocumentation also covers otherconcepts in the area of *specialization* and *extension*. Inheritance is very closely related to the concept of *interfaces*.