Inheritance

The purpose of inheritance is for inheriting child classes (or sub-classes) to contain everything within their parent class. This allows a programmer to write one class, filled with methods and attributes that are then inherited by one or more child classes, which can each do unique things built upon those shared methods and attributes. If using the real world as an example, it is like a parent who pays to educate their children how to cook and speak multiple languages, but then all of those children can go off and do their own things with that knowledge – one might open a restaurant, another a food truck, while another becomes a private chef. Those children end up specializing in different things, and build upon the education given them by their parent, but they still share the same foundation. It works the same way in programming, with each child class containing what they inherited from their parent, but still being able to specialize in different tasks. This inheritance prevents code from being needlessly duplicated in multiple locations, making updates more consistent and stable while reducing the load on the program and programmer.

To share an example, let me show the following short code segments:  
  
class Organ  
{

private \_organName;

private \_bloodNeeded;

private \_thermogenicRequirements

}

Class Heart : Organ {}

Class Liver : Organ {}

Class Brain : Organ {}

In the above example, all of the subclasses are organs, and they all do different things, but they share many things in common. In this example, each subsequent organ will have an attribute for their name, how much blood they need, and what their thermogenic requirements are. There are other things that all organs have, which can be further added to the Organ class and automatically become part of all of its subclasses, without interfering with what each individual organ does, while still allowing each organ to specialize in its own thing.