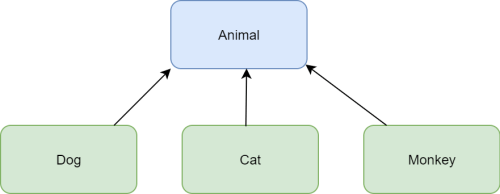
**Inheritance:**

Inheritance allows you to define a class in terms of another class, which makes it easier to create and maintain an application. To use inheritance, you start with a base (parent) class and then derive a child class from the base class. Inheritance works under the “is a” relationship model.

What is unique about an “is a” relation is that it works only in one direction, which is a child to parent. The following diagram demonstrates the “is a” relationship for some animals.

[](https://i2.wp.com/www.brightdevelopers.com/wp-content/uploads/2017/07/cpp_inheritance_basics.png?ssl=1)

The base class is Animal and the derived classes (children) are Dog, Cat, and Monkey. Translating one of the relationships into a statement, you would get along the lines of “A monkey is an animal”. Notice that the other way around doesn’t work. The phrase, “An animal is a monkey” is not a true statement.

Inheritance is one of the most important aspects of Object Oriented Programming (OOP). The key to understanding Inheritance is that it provides code re-usability. In place of writing the same code, again and again, we can simply inherit the properties of one class into the other.

OOP is all about real-world objects and inheritance is a way of representing real-world relationships. Here’s an example – **car, bus, bike** – all of these come under a broader category called **Vehicle**. That means they’ve inherited the properties of class vehicles i.e all are used for transportation.

We can represent this relationship in code with the help of inheritance.