

# OBJECT ORIENTED PROGRAMMING

In all the programs we wrote till now, we have designed our program around functions i.e. blocks of statements which manipulate data. This is called the *procedure-oriented* way of programming. There is another way of organizing your program which is to combine data and functionality and wrap it inside something called an object. This is called the *object oriented* programming paradigm. Most of the time you can use procedural programming, but when writing large programs or have a problem that is better suited to this method, you can use object oriented programming techniques.

Classes and objects are the two main aspects of object oriented programming.

A **class** creates a new *type* where **objects** are **instances** of the class. An analogy is that you can have variables of type `int` which translates to saying that variables that store integers are variables which are instances (objects) of the `int` class.

## *Note for Static Language Programmers*

### **NOTE**

Note that even integers are treated as objects (of the `int` class). This is unlike C++ and Java (before version 1.5) where integers are primitive

native types.

See `help(int)` for more details on the class.

C# and Java 1.5 programmers will find this similar to the *boxing and unboxing* concept.

Objects can store data using ordinary variables that *belong* to the object. Variables that belong to an object or class are referred to as **fields**. Objects can also have functionality by using functions that *belong* to a class. Such functions are called **methods** of the class. This terminology is important because it helps us to differentiate between functions and variables which are independent and those which belong to a class or object. Collectively, the fields and methods can be referred to as the **attributes** of that class.

Fields are of two types - they can belong to each instance/object of the class or they can belong to the class itself. They are called **instance variables** and **class variables** respectively.

A class is created using the `class` keyword. The fields and methods of the class are listed in an indented block.

Source: <http://www.swaroopch.com/notes/python/>