

Class

In object-oriented programming, a class is an extensible program-code-template for creating objects, providing initial values for state (member variables) and implementations of behavior (member functions or methods).

Snippet from *Wikipedia*: **[Class \(computer programming\)](#)**

In object-oriented programming, a **class** is an extensible program-code-template for creating objects, providing initial values for state (member variables) and implementations of behavior (member functions or methods). In many languages, the class name is used as the name for the class (the template itself), the name for the default constructor of the class (a subroutine that creates objects), and as the type of objects generated by instantiating the class; these distinct concepts are easily conflated. Although, to the point of conflation, one could argue that is a feature inherent in a language because of its polymorphic nature and why these languages are so powerful, dynamic and adaptable for use compared to languages without polymorphism present. Thus they can model dynamic systems (i.e. the real world, machine learning, AI) more easily.

When an object is created by a constructor of the class, the resulting object is called an instance of the class, and the member variables specific to the object are called instance variables, to contrast with the class variables shared across the class.

In some languages, classes are only a compile-time feature (new classes cannot be declared at run-time), while in other languages classes are first-class citizens, and are generally themselves objects (typically of type Class or similar). In these languages, a class that creates classes is called a metaclass.

[Creative Commons Attribution-Share Alike 3.0](#)

[kb](#), [programming](#), [devopscreate](#)

From:

<https://almbok.com/> - **ALMBoK.com**

Permanent link:

<https://almbok.com/kb/class>

Last update: **2019/08/17 16:35**

