# abstract

Java allows methods and classes to be declared abstract. An abstract method is not actually implemented in the class. It is merely declared there. The body of the method is then implemented in subclasses of that class. An abstract method must be part of an abstract class. You create abstract classes by adding the keyword abstract after the access specifier, e.g.

public abstract class MotorVehicle

Abstract classes cannot be instantiated. It is a compile-time error to try something like

MotorVehicle m = new MotorVehicle();

when MotorVehicle has been declared to be abstract. MotorVehicle is actually a pretty good example of the sort of class that might be abstract. You're unlikely to be interested in a generic motor vehicle. Rather you'll have trucks, motorcycles, cars, go-carts and other subclasses of MotorVehicle, but nothing that is only a MotorVehicle.

An abstract method provides a declaration but no implementation. In other words, it has no method body. Abstract methods can only exist inside abstract classes and interfaces. For example, the MotorVehicle class might have an abstract fuel() method:

public abstract void fuel();

Car would override/implement this method with a fuel() method that filled the gas tank with gasoline. EighteenWheelerTruck might override this method with a fuel() method that filled its gas tank with diesel. ElectricCar would override/implement this method with a fuel() method that plugged into the wall socket.

[Previous](http://docs.google.com/42.html) | [Next](http://docs.google.com/44.html) | [Top](http://docs.google.com/index.html) | [Cafe au Lait](http://www.cafeaulait.org/)

Copyright 1997 Elliotte Rusty Harold

[elharo@metalab.unc.edu](mailto:elharo@metalab.unc.edu)

Last Modified October 20, 1998