在面向对象方法中，所谓多态性就是不同对象收到相同消息，产生不同的行为。在C++程序设计中，多态性是指用一个名字定义不同的函数，这些函数执行不同但又类似的操作，这样就可以用同一个函数名调用不同内容的函数。换言之，可以用同样的接口访问功能不同的函数，从而实现“一个接口，多种方法”。

在C++中，多态性的实现和联编（也称绑定）这一概念有关。一个源程序经过编译、链接，成为可执行文件的过程是把可执行代码联编（或称装配）在一起的过程。其中在运行之前就完成的联编成为静态联编（前期联编）；而在程序运行之时才完成的联编叫动态联编（后期联编）。  
  
静态联编支持的多态性称为编译时多态性（静态多态性）。在C++中，编译时多态性是通过函数重载和模板实现的。利用函数重载机制，在调用同名函数时，编译系统会根据实参的具体情况确定索要调用的是哪个函数。  
  
动态联编所支持的多态性称为运行时多态（动态多态）。在C++中，运行时多态性是通过虚函数来实现的。