

Overloading methods

Just as C++ allows functions to be overloaded, class methods can be overloaded too – including constructor methods. An overloaded constructor method is useful to assign default values to member variables when an object is created without passing values to the constructor.



overloaded.cpp

1 Rename a copy of the previous example “constructor.cpp” as a new program “overloaded.cpp”

2 In the public section of the Dog class declaration, add inline an overloaded bark method – to output a passed string argument when called
void bark (string noise) { cout << noise << endl ; }

3 Now declare a constructor method prototype that takes no arguments (a default constructor method) and an overloaded constructor method prototype that takes two arguments
Dog() ;
Dog (int, int) ;

4 After the Dog class declaration, add a definition for the default constructor method – assigning default values to class variables when an object is created without passing any arguments
Dog::Dog()
{
 age = 1 ;
 weight = 2 ;
 color = "black" ;
}

5 Now add a definition for the overloaded constructor method – assigning default values to class variables when an object is created passing two arguments
Dog::Dog (int age, int weight)
{
 this -> age = age ;
 this -> weight = weight ;
 color = "white" ;
}

Don't forget



The **this ->** pointer is used to explicitly identify class members when arguments have the same name as members.