**C++ friend function**

If a function is defined as a friend function in C++ then the protected and private data of a class can be accessed using the function.

By using the keyword **friend** compiler knows the given function is a friend function.

For accessing the data, the declaration of a friend function should be done inside the body of a class starting with the keyword **friend.**

Declaration of friend function in C++

**class** class\_name

{

**friend** data\_type function\_name(argument/s);

};

C++ friend function Example

|  |  |
| --- | --- |
|  |  |
| #include <iostream>  using namespace std;  class myclass {  int a, b;  public:  friend int sum(myclass x);  void set\_ab(int i, int j);  };  void myclass :: set\_ab(int i, int j) {  a = i;  b = j;  }  // Note: sum() is not a member function of any class.  int sum(myclass x) {  /\* Because sum() is a friend of myclass, it can directly access a and b. \*/  return x.a + x.b;  }  int main() {  myclass n;  n.set\_ab(3, 4);  cout << sum(n);  return 0;  }  //7 | 1. #include <iostream> 2. **using** **namespace** std; 3. **class** Box 4. { 5. **private**: 6. **int** length; 7. **public**: 8. Box(): length(0) { } 9. **friend** **int** printLength(Box); //friend function 10. }; 11. **int** printLength(Box b) 12. { 13. b.length += 10; 14. **return** b.length; 15. } 16. **int** main() 17. { 18. Box b; 19. cout<<"Length of box: "<< printLength(b)<<endl; 20. **return** 0; 21. } |