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.
- A class's friend function is defined outside that class's scope, but it has the right to access all private and protected members of the class.
- Even though the prototypes for friend functions appear in the class definition, friends are not member functions.
- A friend function in C++ is a function that is preceded by the keyword "friend."

Syntax:

```
class class_name {
    friend data_type function_name(argument); // syntax of friend
function.
};
```

- The function can be defined anywhere in the program like a normal C++ function.
- The function definition does not use either the keyword friend or scope resolution operator.

Example:

```
#include <iostream>
using namespace std;
class Rectangle {
    private:
        int length;
    public:
        Rectangle() {
            length = 10;
        }
    friend int printLength(Rectangle); //friend function
};
int printLength(Rectangle b) {
    b.length += 10;
```

```
    return b.length;
}
int main() {
    Rectangle b;
    cout << "Length of Rectangle: " << printLength(b) << endl;
    return 0;
}
Output:
Length of Rectangle: 20
```

Characteristics of friend function:

- A friend function can be declared in the private or public section of the class.
- It can be called a normal function without using the object.
- A friend function is not in the scope of the class, of which it is a friend.
- A friend function is not invoked using the class object as it is not in the class's scope.
- A friend function cannot access the private and protected data members of the class directly.
- It needs to make use of a class object and then access the members using the dot operator.
- A friend function can be a global function or a member of another class.