**friend function**

friends is keyword of c++ programming language .A friend function of a class is defined outside that class scope but it has the right to access all private and protected members of the class, provide the prototypes for friend functions appear in the class definition, friends are not member functions.

A friend can be a function, function template, or member function, or a class or class template, in which case the entire class and all of its members are friends.

To declare a function as a friend of a class, precede the function prototype in the class definition with keyword **friend** as follows

**Syntax**

class class-name

{

Member variables;

Member Function;

friend return\_type function\_name(arg1, aru2 .....aru^n);

};

return\_type function\_name(arg1, aru2 .....aru^n)

{

}

**Advantages of friend function in C++**

* **Friend function in c++** provide a degree of freedom in the interface design option
* You can use a friend function to bridge two classes by operating objects of two different classes.
* It increases the versatility of overloading operators.
* It enhances encapsulation. Only the programmer who has access to the class’s source code can make a function friend to that class.
* You may declare a member function of a class as a friend of another class.
* It works symmetrically with all its friends.

**Exam no 1**

#include<iostream.h>

#include<conio.h>

/\* show the values using friend function\*/

class Data

{

private : int x,y;

public: Data( )

{

x=10;

y=20;

}

friend void show( Data );

};

void show (Data d1)

{

cout<<”\n X=>”<<d1.x;

cout<<”\n Y=>”<<d1.y;

}

void main()

{

clrscr( );

Data d;

show( d);

getch();

}

Output

X=>10

Y=>20

**Exam no 2**

#include<iostream.h>

#include<conio.h>

/\* Interchange two values using friend function\*/

class Data

{

private: int x,y;

public: void setvalue()

{

cout<<"\nEnter the value of X=>";

cin>>x;

cout<<"\nEnter the value of Y=>";

cin>>y;

}

void show()

{

cout<<"\n The value of X=>"<<x;

cout<<"\n The value of Y=>"<<y;

}

friend void swap( Data );

};

void swap(Data d1)

{

int t;

t=d1.x;

d1.x=d1.y;

d1.y=t;

cout<<"\n After swap the value of X and Y is";

cout<<"\n The value of X=>"<<d1.x;

cout<<"\n The value of Y=>"<<d1.y;

}

void main()

{

clrscr();

Data d;

d.setvalue();

d.show();

swap(d);

getch();

}

**Friend Class:**  
A **friend class** is a class that can access the private and protected members of a class in which it is declared as **friend**. This is needed when we want to allow a particular class to access the private and protected members of a class.

#include <iostream>

class XYZ

{

private: char ch='A';

int num = 10;

public:

/\* This statement would make class ABC a friend class of XYZ, this means that ABC can access the private and protected members of XYZ class.

\*/

friend class ABC;

};

class ABC

{

public: void disp( XYZ obj )

{

cout<<obj.ch<<endl;

cout<<obj.num<<endl;

}

};

void main()

{

ABC obj1;

XYZ obj2;

obj1.disp(obj2);

getch();

}