**PROGRAM** –
CREATE TWO CLASSES WITH DATA MEMBERS IN BOTH WHICH ARE TO BE ADDED USING THE FUNCTION WHICH IS FRIEND FOR BOTH THE CLASSES.

```
//BHAVNA VERMA-171210019-22/01/2019
//FRIEND FUNCTION
#include <iostream>
using namespace std;
class B;
class A
{
   int a;
   public:
     A(): a(12) { }
     // friend function
     friend int add(A, B);
};

class B
{
            int b;
   public:
      B(): b(30) { }
      // friend function
      friend int add(A , B);
};

int add(A obA, B obB)
{
  return (obA.a + obB.b);
}

int main()
{
   A obA;
   B obB;
   cout<<"Sum: "<< add(obA, obB);
   return 0;
}
```

```
/* Explanation :

Data members of class A  & class B
are to be added with the help of a
function which is made friend to both
the class A & class B  */
```

```
Sum: 42
--------------------------------
Process exited after 0.02517 seconds with return value 0
Press any key to continue . . .
```

**Friend function** –

A friend function can be given access to private and protected members. A friend function can be:
a) A method of another class
b) A global function

Some important points about friend functions:

**1)** Friends should be used only for limited purpose. too many functions declared as friends of a class with protected or private data, it lessens the value of encapsulation of separate classes in object-oriented programming.

**2)** Friendship is not mutual. If a class A is friend of B, then B doesn't become friend of A automatically.

**3)** Friendship is not inherited

**4)** The concept of friends is not there in Java.