You have learned how to declare a function become a friend function with a class, Just add the prototype of that function and put a “friend” in front of the prototype.

Example:

class A

{

private:

int age;

public:

void setAge(int x)

{

age = x;

}

friend int sum(A a1, A a2);

};

int sum(A a1, A a2)

{

return a1.age + a2.age;

}

By the declaration in class A, you give the right to function “sum” to access all the private, protected members. So you can use the function “sum” in main like this:

int main()

{

A a1, a2;

a1.setAge(10);

a2.setAge(25);

cout << "sum of a1 and a2 is: " << sum(a1, a2) << endl;

system("pause");

return 0;

}

Is it possible to have a single function become a friend with more than one classes at the same time?

Answer is yes. You just have to declare all the classes at the beginning. (declare the class name only)

Then you implement each class below with the friend function prototype.

Check the example on next page. If there is any questions, we still have SI sessions until the end of this semester !

#include <iostream>

using namespace std;

class A;

class B;

class A

{

private:

int age;

public:

void setAge(int x)

{

age = x;

}

friend int sum(A a, B b);

};

class B

{

private:

int age;

public:

void setAge(int x)

{

age = x;

}

friend int sum(A a, B b);

};

int sum(A a, B b)

{

return a.age + b.age;

}

int main()

{

A a;

B b;

a.setAge(10);

b.setAge(25);

cout << "sum of a.age and b.age is: " << sum(a, b) << endl;

system("pause");

return 0;

}