A **forward declaration** tells the compiler about the existence of an entity before actually defining the entity. Forward declarations can also be used with other entity in C++, such as functions, variables and user-defined types, identifier, variable, function, class, etc.

|  |
| --- |
| #  **class B** {    public:      int x;        void getdata(int n)      {          x = n;      }      friend int sum(A, B);  };    **class A** {  public:      int y;        void getdata(int m)      {          y = m;      }      friend int sum(A, B);  };    int sum(A m, B n)  {      int result;      result = m.y + n.x;      return result;  }    int main()  {      B b;      A a;      a.getdata(5);      b.getdata(4);      cout << "The sum is : " << sum(a, b);      return 0;  } |

**Output:**

Compile Errors :

prog.cpp:14:18: error: 'A' has not been declared

friend int sum(A, B);

**Explanation:** Here the compiler throws this error because, in class B, the object of class A is being used, which has no declaration till that line. Hence compiler couldn’t find class A. So what if class A is written before class B?

|  |
| --- |
| #include <iostream>      using namespace std;    class A {  public:      int y;        void getdata(int m)      {          y = m;      }      friend int sum(A, B);  };    class B {    public:      int x;        void getdata(int n)      {          x = n;      }      friend int sum(A, B);  };    int sum(A m, B n)  {      int result;      result = m.y + n.x;      return result;  }    int main()  {      B b;      A a;      a.getdata(5);      b.getdata(4);      cout << "The sum is : " << sum(a, b);      return 0;  } |

**Output:**

Compile Errors :

prog.cpp:16:23: error: 'B' has not been declared

friend int sum(A, B);

**Now it is clear that any of the above codes wouldn’t work, no matter in which order the classes are written. Hence this problem needs a new solution- Forward Declaration.**

|  |
| --- |
| // Forward declaration  class A;  class B;    class B {      int x;    public:      void getdata(int n)      {          x = n;      }      friend int sum(A, B);  };    class A {      int y;    public:      void getdata(int m)      {          y = m;      }      friend int sum(A, B);  };  int sum(A m, B n)  {      int result;      result = m.y + n.x;      return result;  }    int main()  {      B b;      A a;      a.getdata(5);      b.getdata(4);      cout << "The sum is : " << sum(a, b);      return 0;  } |

**Output:**

The sum is : 9