#include<iostream.h>

www.hndit.com

class Csquare;

class Rectangle{

int width,hight;

public:

 int area(){

 return (width\*hight);

 }

 void convert(Csquare a);

};

class Csquare{

private:

 int side;

public:

 void set\_side(int a){

 side=a;

 }

 frind class Rectangle;

 };

 void Rectangle::convert(Csquare a){

 width=a.side;

 hight=a.side;

www.hndit.com

 }

 int main(){

Csquare a;

 Rectangle b;

a.set\_side(4);

b.convert(a);

cout<<b.area();

return 0;}

1. A function can be a member of one class and a friend of another

#include <iostream>
**using**namespace std;

**class**Cylinder; // a forward declaration

**enum**colors { red, green, yellow };

**class**Cube {
 colors color;
**public**:
  Cube(colors c) { color= c; }
  bool sameColor(Cylinder y);
};
**class**Cylinder {
 colors color;
**public**:
  Cylinder(colors c) { color = c; }

friend bool Cube::sameColor(Cylinder y);
};
bool Cube::sameColor(Cylinder y) {
  **if**(color == y.color)
    **return true**;
  **else**
    **return false**;
}
**int**main()
{
  Cube cube1(red);
  Cube cube2(green);
  Cylinder cyl(green);

  **if**(cube1.sameColor(cyl))
    cout << "cube1 and cyl are the same color.\n";
  **else**
    cout << "cube1 and cyl are different colors.\n";

  **if**(cube2.sameColor(cyl))
    cout << "cube2 and cyl are the same color.\n";
  **else**
    cout << "cube2 and cyl are different colors.\n";

  **return**0;
}

www.hndit.com