```cpp
#include "Complex.h"

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

Complex::Complex(double r , double i) { //giving values to data members

 /* if this was written in files only in.h file
 we put the default values not in func call*/

    real = r;
    imag = i;

}
/*without operator overloading*/ /* +  */
Complex Complex::add(Complex x) { //add func takes the Complex x and
                                  //add it to calling obj complex nb
                                 //and return tot
    Complex tot;
    tot.real = this->real + x.real;
    tot.imag = this->imag + x.imag;

    return tot;

}


/*with operator overloading*/  /*  +  */
Complex Complex ::operator+(Complex x) {  /* operator overload is only
  available
                                       to its complex class*/

    Complex tot;  //same code //so now we can use c3=c1+c2
    tot.real = this->real + x.real;
    tot.imag = this->imag + x.imag;

    return tot;
}

/*without operator overloading*/ /* = */
bool Complex:: isEqual(Complex x) {

    if (real == x.real && imag == x.imag) return true;
    else                                  return false;

}

/*with operator overloading*/ /* = */
bool Complex::operator == (Complex x) {


    if (real == x.real && imag == x.imag) return true;
```

```
53      else                                    return false;
54
55  }
56  void Complex::prnt(void) {
57
58      cout << fixed << setprecision(2); //it will print two digits after each
          number
59
60      cout << real << "  + i" << imag << endl;
61  }
62
```