```
// Rat class i.e. a class for rational numbers

#include <iostream>
using namespace std;

class Rat{
private:
   int n;
   int d;
public:
   // constructors

   // default constructor
   Rat(){
      n=0;
      d=1;
   }


   // 2 parameter constructor
   Rat(int i, int j){
      n=i;
      d=j;
   }

   // conversion constructor

   Rat(int i){
      n=i;
      d=1;
   }

   //accessor functions (usually called get() and set(...)  )
   int getN(){ return n;}
   int getD(){ return d;}

   void setN(int i){ n=i;}
   void setD(int i){ d=i;}


   //arithmetic operators

   Rat operator+(Rat r){
      Rat t;
      t.n=n*r.d+d*r.n;
      t.d=d*r.d;
      return t;
   }
```

```
  // 2 overloaded i/o operators
    friend ostream& operator<<(ostream& os, Rat r);

    friend istream& operator>>(istream& is, Rat& r);


}; //end Rat

// operator<<() is NOT a member function but since it was declared a friend of Rat
// it has access to its private parts.

ostream& operator<<(ostream& os, Rat r){
        os<<r.n<<" / "<<r.d<<endl;
        return os;}

// operator>>() is NOT a member function but since it was declared a friend of Rat
// it has access to its provate parts.

istream& operator>>(istream& is, Rat& r){
    is>>r.n>>r.d;
    return is;
}



int  main(){
    Rat x(1,2), y(2,3), z;
    z=x+y;
    cout<<z;

    x.setN(3);
    y.setD(2);
    z=x+y;
    cout<<z;

    cin>>x;
    cout<<x;

    z= x+5;
    cout<<z;
    system("pause");
    return 0;

}
```