2021-10-05 -- Scratchpad of CSE213 (Sec-1)

How stream works:

```
cin>>x;         //123

Perception: we entered integer value one hundred and twenty three
Reality:        '1'             '2'             '3'
ASCII code:     49              50              51
Binary:         00110001        00110010        00110011
```

User input: 3 bytes --> 00110001 00110010 00110011 --> accumulated in a buffer inside stream object

conversion process is called parsing

```
Desired content of x:   00000000 00000000 00000000 01111011
```

```
cin>>x>>y>>f;
12 13 3.5

12
13
3.5
```

NOTE: "space" and "Enter" kay are natural seperator of bytes inside the stream buffer

operator overloading:

Note:

- for unary operator, the sole operand (object) is the client of the operator method
- for binary operator, the FIRST operand (object) ALWAYS has to qualify as client of the operator method

```
        ComplexNo c1, c2, c3, c4;
        c4 = c1 + c2 + c3;              //method chain: case-1
        int x,y,z;
        cin >> x >> y >> z;             //method chain: case-2
```

Q: What is the difference between case-1 & case-2 in terms of client?

```
class ComplexNo{
        int real,img;
        public:
        void setComplexNo(){
                cout<<"Enter real value: "; cin>>real;
                cout<<"Enter imaginary value: "; cin>>img;
        }
        void showComplexNo(){
                cout<<real<<(img>=0?"+":"")<<img<<"i"<<endl;
        }
        ComplexNo operator+(ComplexNo c){
                ComplexNo temp;
                temp.real = real + c.real;
                temp.img = img + c.img;
                return temp;
        }
        ComplexNo operator+(int val){
                ComplexNo temp;
                temp.real = real + val;
                temp.img = img;
                return temp;
        }
};
int main(){
        ComplexNo c1, c2, c3, c4, c5;
        c1.setComplexNo();      c2.setComplexNo();
c3.setComplexNo();
        //c4 = c1.add(c2);              //c1+c2;
        c4 = c1 + c2 + c3;
        cout<<"c1="; c1.showComplexNo();
        cout<<"c2="; c2.showComplexNo();
        cout<<"c3"; c3.showComplexNo();
        cout<<"after c4=c1+c2, c4:"; c4.showComplexNo();
        c5=c4+10;
        cout<<"after c5=c4+10, c5:"; c5.showComplexNo();
        c6=2+c5;   //increase both real & img by 2
        cout<<"after c6=c5+2, c6:"; c6.showComplexNo();
}
Note:
        - for c6=2+c5 to work we need to define global operator+
function
                with both operands as parameter

        - Now we need define some getter methods in ComplexNo class to
return the
                value of the fields to its caller

        - by convention, getter method is field specific, and the name
begins with 'get' and
                suffix-ed with the field name;
        - Ex: field: id, getter method will be
                int getId(){ return id;}

        - by convention, setter method is field specific, to set the
value of the field
                and the name begins with 'set' and suffix-ed with the
field name;
        - Ex: field: id, setter method will be
                void setId(int val){ id=val;}
```