Assignment 3

Q1

```
using namespace std;

#include<iostream>

struct Complex{

        int real;

        int img;

        void display(){

                cout<<"\nreal+img = "<<this->real<<"+"<<this->img<<"i";

        }

        void setreal(int a){

                this->real=a;

                        }

        void setimg(int a){

                this->img=a;

                        }

                int getreal(){

                                return this->real;

                        }

                int getimg(){

                return this->img;

                        }

                        Complex(){
```

```
                this->real=0;
                this->img=0;
        }
        Complex(int a,int b){
                this->real=a;
                this->img=b;
        }
Complex operator+(Complex a){
        Complex temp;
        temp.setreal(this->real+a.getreal());
        temp.setimg(this->img+a.getimg());
        return temp;
}
Complex operator+(int a){
                Complex temp;
        temp.setreal(this->real+a);
        temp.setimg(this->img+a);
        return temp;
}
Complex operator-(Complex a){
        Complex temp;
        temp.setreal(this->real-a.getreal());
        temp.setimg(this->img-a.getimg());
        return temp;
```

```
}
Complex operator-(int a){
                Complex temp;
        temp.setreal(this->real-a);
        temp.setimg(this->img-a);
        return temp;
}
Complex operator*(Complex a){
        Complex temp;
        temp.setreal(this->real*a.getreal());
        temp.setimg(this->img*a.getimg());
        return temp;
}
Complex operator*(int a){
                Complex temp;
        temp.setreal(this->real*a);
        temp.setimg(this->img*a);
        return temp;
}
Complex operator/(Complex a){
        Complex temp;
        temp.setreal(this->real/a.getreal());
        temp.setimg(this->img/a.getimg());
        return temp;
```

```
                }
                Complex operator/(int a){
                        Complex temp;
                    temp.setreal(this->real/a);
                    temp.setimg(this->img/a);
                    return temp;
                }
};
  Complex operator+(int b,Complex a ){
            Complex temp;
            temp.setreal(b+a.getreal());
            temp.setimg(b+a.getimg());
            return temp;
   }
   Complex operator-(int b,Complex a ){
            Complex temp;
            temp.setreal(b-a.getreal());
            temp.setimg(b-a.getimg());
            return temp;
   }
   Complex operator*(int b,Complex a ){
            Complex temp;
            temp.setreal(b*a.getreal());
            temp.setimg(b*a.getimg());
```

```
                return temp;
    }
    Complex operator/(int b,Complex a ){
                Complex temp;
                temp.setreal(b/a.getreal());
                temp.setimg(b/a.getimg());
                return temp;
    }
int main(){
        Complex c(45,55);
        Complex c2(5,3);
        Complex c3=35/c2;
        c.display();
        c2.display();
        c3.display();

}
```

Q2

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

        struct Distance {
   int feet;
   int inch;
```

```
void display(){


cout<<"\nFeet:"<<this->feet;
cout<<"\nInches:"<<this->inch;
    }
    void setfeet(int a){
            this->feet=a;
    }
    void setinch(int a){
            this->inch=a;
    }
    int getfeet(){
            return this->feet;
    }
    int getinch(){
            return this->inch;
    }
    Distance(){
            this->feet=0;
            this->inch=0;
    }
            Distance(int a,int b){
            this->feet=a;
            this->inch=b;
```

```
}
Distance operator+(Distance arr){
        Distance temp;
  temp.setfeet(this->feet+arr.getfeet());
  temp.setinch(this->inch+arr.getfeet());
        return temp;
}
        Distance operator+(int r){
                Distance temp;
  temp.setfeet(this->feet+r);
  temp.setinch(this->inch+r);
        return temp;
}
        Distance operator-(Distance arr){
        Distance temp;
  temp.setfeet(this->feet-arr.getfeet());
  temp.setinch(this->inch-arr.getfeet());
        return temp;
}
        Distance operator-(int r){
                Distance temp;
  temp.setfeet(this->feet-r);
  temp.setinch(this->inch-r);
        return temp;
```

```
}
        Distance operator*(Distance arr){
        Distance temp;
 temp.setfeet(this->feet*arr.getfeet());
 temp.setinch(this->inch*arr.getfeet());
        return temp;
}
        Distance operator*(int r){
                Distance temp;
 temp.setfeet(this->feet*r);
 temp.setinch(this->inch*r);
        return temp;
}
        Distance operator/(Distance arr){
        Distance temp;
 temp.setfeet(this->feet/arr.getfeet());
 temp.setinch(this->inch/arr.getfeet());
        return temp;
}
        Distance operator/(int r){
                Distance temp;
 temp.setfeet(this->feet/r);
 temp.setinch(this->inch/r);
        return temp;
        }
};

Distance operator+(int a,Distance D){
                Distance temp;
                temp.setfeet(a+D.getfeet());
                temp.setinch(a+D.getinch());
                return temp;
        }
        Distance operator-(int a,Distance D){
                Distance temp;
                temp.setfeet(a-D.getfeet());
                temp.setinch(a-D.getinch());
                return temp;
        }
        Distance operator*(int a,Distance D){
                Distance temp;
                temp.setfeet(a*D.getfeet());
                temp.setinch(a*D.getinch());
                return temp;
        }

        Distance operator/(int a,Distance D){
                Distance temp;
                temp.setfeet(a/D.getfeet());
                temp.setinch(a/D.getinch());
                return temp;
        }
int main(){
        Distance d1(75,85);
        Distance d2(5,15);
        Distance d3=100/d2;
        d3.display();
}
```

Q3

Complex class

```
using namespace std;
#include<iostream>
struct Complex{
        int real;
        int img;
        void display(){
                cout<<"\nreal+img = "<<this->real<<"+"<<this->img<<"i";
        }
        void setreal(int a){
                this->real=a;
                        }
        void setimg(int a){
                this->img=a;
                        }
                int getreal(){
                                return this->real;
                        }
                int getimg(){
                return this->img;
                        }
                        Complex(){
                                this->real=0;
                                this->img=0;
                        }
                        Complex(int a,int b){
                                this->real=a;
                                this->img=b;
                        }
                         Complex operator&&(Complex a) {
        Complex temp;
        if ((this->real!=0||this->img != 0) &&
(a.getreal()!=0||a.getimg()!= 0)) {
            temp.setreal(1);
            temp.setimg(1);
        } else {
            temp.setreal(0);
            temp.setimg(0);
        }
        return temp;
    }
    Complex operator||(Complex c){
        Complex temp;
        if (this->real!=0||this->img!=0||c.getreal()!=0||c.getimg()!= 0) {
            temp.setreal(1);
            temp.setimg(1);
        } else {

            temp.setreal(0);
            temp.setimg(0);
        }
        return temp;
    }
                         Complex operator!() {
        Complex temp;
        if (this->real==0&&img==0) {
            temp.setreal(1);
            temp.setimg(1);
        } else {
            temp.setreal(0);
            temp.setimg(0);
        }
        return temp;
    }

};
int main(){

    Complex c1(0, 0);
    Complex c2(3, 4);
    Complex c3(0, 0);
    Complex c4 = !c1;
    c4.display();



    Complex c5 = c1 && c2;
    c5.display();

    Complex c6 = c1 || c3;
     c6.display();
}
```

Distance

```
using namespace std;
#include<iostream>
struct Distance {
```

```
int feet;
int inch;

void display() {
   cout<<"\nfeet = "<<this->feet;
            cout<<"\n inch = " <<this->inch;
}

void setfeet(int f) {
   this->feet=f;
}

void setinch(int i) {
   this->inch=i;
}

int getfeet() {
   return this->feet;
}

int getinch() {
   return this->inch;
}
```

```
    Distance() {
        this->feet=0;
        this->inch=0;
    }

    Distance(int f, int i) {
        this->feet=f;
        this->inch=i;
    }

    Distance operator&&(Distance d) {
        Distance temp;
        if ((this->feet!=0||this-
>inch!=0)&&(d.getfeet()!=0||d.getinch()!=0)) {
            temp.setfeet(1);
            temp.setinch(1);
        } else {
            temp.setfeet(0);
            temp.setinch(0);
        }
        return temp;
    }

    Distance operator||(Distance d) {
        Distance temp;
        if (this->feet!=0||this->inch!=0||d.getfeet()!=0||d.getinch()!=0)
{
           temp.setfeet(1);
            temp.setinch(1);
        } else {
            temp.setfeet(0);
            temp.setinch(0);
        }
        return temp;
    }

    Distance operator!() {
        Distance temp;
        if (this->feet==0&&this->inch==0) {
            temp.setfeet(1);
            temp.setinch(1);
        } else {
            temp.setfeet(0);
            temp.setinch(0);
        }
        return temp;
    }
};
```

```
int main() {
    Distance a(5, 8);
    Distance b(6, 3);
    Distance c = a && b;
    c.display();
}
```

Q4

Complex

```
using namespace std;
#include<iostream>
struct Complex {
    int real;
    int img;

    void display() {
        cout<<"\nreal = "<<this->real;
        cout<<"\nimg = "<<this->img;
    }

    void setreal(int r) {
        this->real=r;
    }
}

void setimg(int i) {
    this->img=i;
}

int getreal() {
    return this->real;
}

int getimg() {
    return this->img;
}

Complex() {
    this->real=0;
    this->img=0;
}

Complex(int r, int i) {
    this->real=r;
    this->img=i;
}
```

```
}

Complex operator==(Complex c) {
    Complex temp;
    if (this->real == c.getreal() && this->img == c.getimg()) {
        temp.setreal(1);
        temp.setimg(1);
    } else {
        temp.setreal(0);
        temp.setimg(0);
    }
    return temp;
}
Complex operator!=(Complex c) {
    Complex temp;
    if (this->real != c.getreal()||this->img != c.getimg()) {
        temp.setreal(1);
        temp.setimg(1);
    } else {
        temp.setreal(0);
        temp.setimg(0);
    }
    return temp;
}
```

```
Complex operator<(Complex c) {
   Complex temp;
   double mag1 = this->real * this->real + this->img * this->img;
   double mag2 = c.real * c.real + c.img * c.img;
   if (mag1 < mag2) {
      temp.setreal(1);
      temp.setimg(1);
   } else {
      temp.setreal(0);
      temp.setimg(0);
   }
   return temp;
}

Complex operator>(Complex c) {
   Complex temp;
   double mag1 = this->real * this->real + this->img * this->img;
   double mag2 = c.real * c.real + c.img * c.img;
   if (mag1 > mag2) {
      temp.setreal(1);
      temp.setimg(1);
   } else {
      temp.setreal(0);
      temp.setimg(0)
}
        return temp;
    }
};

int main() {
    Complex a(3, 4);
    Complex b(3, 4);
    Complex c(5, 12);

    Complex result;

    result = a == b;
    cout << "\nResult of a == b: ";
    result.display();



    result = a != c;
    cout << "\nResult of a != c: ";
    result.display();



    result = a < c;
    cout << "\nResult of a < c: ";
    result.display();



    result = c > a;
    cout << "\nResult of c > a: ";
    result.display();
}
```

Q5

```
using namespace std;
#include<iostream>
struct Complex{
        int real;
        int img;
        void display(){
                cout<<"real+img = "<<this->real<<"+"<<this-
>img<<"i"<<endl;
        }
        void setreal(int a){
                this->real=a;
        }
        void setimg(int a){
                this->img=a;
        }
   int getreal(){
      return this->real;
}
int getimg(){
      return this->img;
}
Complex(int a,int b){
      this->real=a;
      this->img=b;
}
Complex(){
      this->real=0;
      this->img=0;
}
Complex operator++(){
      Complex temp;
      temp.real=++this->real;
      temp.img=++this->img;
      return temp;
}
 Complex operator++(int){
      Complex temp;
      temp.real=this->real++;
      temp.img=this->img++;
      return temp;
```

```
   }
   Complex operator--(){
          Complex temp;
          temp.real=--this->real;
          temp.img=--this->img;
          return temp;
   }
    Complex operator--(int){
          Complex temp;
          temp.real=this->real--;
          temp.img=this->img--;
          return temp;
   }
};
int main(){
          Complex X(85,45);
//        X.display();
          Complex c=X--;
          c.display();
          X.display();
          Complex v(5,10);
          // c=++v;
          //        c.display();
```

```
}
```

Distance

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

struct Distance {
    int feet;
    int inch;

    Distance(int f, int i) {
        this->feet = f;
        this->inch = i;
    }

    Distance() {
        this->feet = 0;
        this->inch = 0;
    }

    void display() {
        cout << "Distance = " << this->feet << " feet " << this->inch << " 
inches" << endl;
    }
```

```
}

void setfeet(int f) {
    this->feet = f;
}

int getfeet() {
    return this->feet;
}

void setinch(int i) {
    this->inch = i;
}

int getinch() {
    return this->inch;
}

Distance operator++() {
    Distance temp;
    temp.feet = ++this->feet;
    temp.inch = ++this->inch;
    return temp;
}

Distance operator++(int) {
    Distance temp;
    temp.feet=this->feet;
    temp.inch=this->inch;
    this->feet++;
    this->inch++;
    return temp;
}

Distance operator--() {
    Distance temp;
    temp.feet = --this->feet;
    temp.inch = --this->inch;
    return temp;
}


Distance operator--(int) {
    Distance temp;
    temp.feet=this->feet;
    temp.inch=this->inch;
    this->feet--;
    this->inch--;
    return temp;
}
```

```
};

int main() {
    Distance X(85, 45);
    X.display();

    Distance c = X--;
    c.display();
    X.display();

    Distance v(5, 10);
    v = ++v;
    v.display();

    return 0;
}
```