3 assign cpp

#include<stdio.h>

struct Complex{

int real;

int img;

Complex(){

this->real=0;

this->img=0;

printf("Defalut construtor called!\n");

}

Complex(int r,int i){

this->real=r;

this->img=i;

printf("parameterised constructor!\n");

}

void setReal(int r){

this->real=r;

}

void setImg(int i){

this->img=i;

}

int getReal(){

return this->real;

}

int getImg(){

return this->img;

}

void display(){

printf("\n%d+%di\n",this->real,this->img);

}

Complex sub(Complex c){

Complex temp;

temp.real=this->real-c.real;

temp.img=this->img-c.img;

return temp;

}

Complex sub(int a){

Complex temp;

temp.real=this->real-a;

temp.img=this->img-a;

return temp;

}

};

//declaration of the global function

Complex sub(int ,Complex);

int main(){

Complex c1(20,40);

c1.display();

Complex c2(2,4);

c2.display();

Complex c3;

c3=c1.sub(c2);

c3.display();

Complex c4;

c4=c1.sub(10);

c4.display();

Complex c5;

c5=sub(10,c1);

c5.display();

return 0;

}

Complex sub(int a,Complex c){

Complex temp;

temp.setReal(c.getReal()-a);

temp.setImg(c.getImg()-a);

return temp;

}

//complex add

#include<stdio.h>

struct Complex{

int real;

int img;

//setters

void setreal(int r){

this->real=r;

}

void setimg(int i){

this->img=i;

}

void display(){

printf("%d+%di\n",this->real,this->img);

}

//getters

int getReal(){

return this->real;

}

int getImg(){

return this->img;

}

//constructor

Complex(){

this->real=0;

this->img=0;

printf("Constructor called\n");//POC

}

//parameterised constructor

Complex(int r, int i){

this->real=r;

this->img=i;

printf("Parameterised Constructor called\n");//POC

}

//add

Complex add(Complex c){

Complex temp;

temp.real=this->real+c.real;

temp.img=this->img+c.img;

return temp;

}

Complex add(int a){

Complex temp;

temp.real=this->real+a;

temp.img=this->img+a;

return temp;

}

};

Complex add(int ,Complex);

int main(){

Complex c1(10,20);

c1.display();

Complex c2(5,3);

c2.display();

Complex c3;

c3=c1.add(c2);

c3.display();

Complex c4;

c4=c1.add(10);

c4.display();

Complex c5;

c5=add(10,c1);

c5.display();

return 0;

}

Complex add(int a,Complex c){

Complex temp;

int r=c.getReal()+a;

int i=c.getImg()+a;

temp.setreal(r);

temp.setimg(i);

return temp;

}

//div

#include<stdio.h>

struct Complex{

int real;

int img;

Complex(){

printf("Defalut constructor called!!\n");

this->real=0;

this->img=0;

}

Complex(int r,int i){

printf("parameterised constructor called!!\n");

this->real=r;

this->img=i;

}

void setReal(int r){

this->real=r;

}

void setImg(int i){

this->img=i;

}

int getReal(){

return this->real;

}

int getImg(){

return this->img;

}

void display(){

printf("%d+%di\n",this->real,this->img);

}

Complex div(Complex c){

Complex temp;

temp.real=this->real/c.real;

temp.img=this->img/c.img;

return temp;

}

Complex div(int a){

Complex temp;

temp.real=this->real/a;

temp.img=this->img/a;

return temp;

}

};

Complex div(int,Complex);

int main(){

Complex c1(20,40);

c1.display();

Complex c2(2,4);

c2.display();

Complex c3;

c3=c1.div(c2);

c3.display();

Complex c4;

c4=c1.div(2);

c4.display();

Complex c5;

c5=div(2,c1);

c5.display();

return 0;

}

Complex div(int a,Complex c){

Complex temp;

int r=c.getReal()/a;

int i=c.getImg()/a;

temp.setReal(r);

temp.setImg(i);

return temp;

}

//div

#include<stdio.h>

struct Complex{

int real;

int img;

Complex(){

printf("Deflaut constructor called!\n");

this->real=0;

this->img=0;

}

Complex(int r,int i){

printf("Parameterised constructor called!!\n");

this->real=r;

this->img=i;

}

void setReal(int r){

this->real=r;

}

void setImg(int i){

this->img=i;

}

int getReal(){

return this->real;

}

int getImg(){

return this->img;

}

void display(){

printf("%d+%di\n",this->real,this->img);

}

Complex mul(Complex c){

Complex temp;

temp.real=this->real\*c.real;

temp.img=this->img\*c.img;

return temp;

}

Complex mul(int a){

Complex temp;

temp.real=this->real\*a;

temp.img=this->img\*a;

return temp;

}

};

//declartion of global function

Complex mul(int , Complex );

int main(){

Complex c1(10,20);

c1.display();

Complex c2(2,3);

c2.display();

Complex c3;

c3=c1.mul(c2);

c3.display();

Complex c4;

c4=c1.mul(2);

c4.display();

Complex c5;

c5=mul(2,c1);

c5.display();

return 0;

}

Complex mul(int a, Complex c){

Complex temp;

int r=c.real\*a;

temp.setReal(r);

int i=c.real\*a;

temp.setImg(i);

return temp;

}

#include<stdio.h>

struct Complex{

int real;

int img;

Complex(){

this->real=0;

this->img=0;

printf("Defalut construtor called!\n");

}

Complex(int r,int i){

this->real=r;

this->img=i;

printf("parameterised constructor!\n");

}

void setReal(int r){

this->real=r;

}

void setImg(int i){

this->img=i;

}

int getReal(){

return this->real;

}

int getImg(){

return this->img;

}

void display(){

printf("\n%d+%di\n",this->real,this->img);

}

Complex sub(Complex c){

Complex temp;

temp.real=this->real-c.real;

temp.img=this->img-c.img;

return temp;

}

Complex sub(int a){

Complex temp;

temp.real=this->real-a;

temp.img=this->img-a;

return temp;

}

};

//declaration of the global function

Complex sub(int ,Complex);

int main(){

Complex c1(20,40);

c1.display();

Complex c2(2,4);

c2.display();

Complex c3;

c3=c1.sub(c2);

c3.display();

Complex c4;

c4=c1.sub(10);

c4.display();

Complex c5;

c5=sub(10,c1);

c5.display();

return 0;

}

Complex sub(int a,Complex c){

Complex temp;

temp.setReal(c.getReal()-a);

temp.setImg(c.getImg()-a);

return temp;

}