**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Name of Developer: ASHLESH UMESH KHAJBAGE, roll no -2508,**

**Title: OPERATIONS ON COMPLEX NUMBER**

**Lab: OOPS**

**Date: 04/01/2018**

**Objective of Project: The objective of program is to learn concept of classes and object**

**Verified By:**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**#include<iostream> //header file declaration**

**using namespace std;**

**class complex**

**{**

**float real,img;**

**public:**

**void read();**

**void display();**

**complex add(complex); // class functions**

**complex sub(complex);**

**complex multiply(complex);**

**complex div(complex);**

**complex conjugate();**

**};**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*defination-of-class-methods\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**void complex::read()**

**{**

**cout<<"\nEnter the complex number";**

**cout<<"\nreal part:";**

**cin>>real;**

**cout<<"imaginary part:";**

**cin>>img;**

**}**

**void complex::display()**

**{**

**if(img>=0)**

**cout<<real<<"+i"<<img<<endl;**

**else**

**{**

**float t=-1\*img;**

**cout<<real<<"-i"<<t<<endl;**

**}**

**}**

**complex complex::add(complex o) //add complex**

**{**

**complex t;**

**t.real= real+ o.real;**

**t.img=img+o.img;**

**return t;**

**}**

**complex complex::sub(complex o) //subtract complex**

**{**

**complex t;**

**t.real= real- o.real;**

**t.img=img-o.img;**

**return t;**

**}**

**complex complex::multiply(complex o) //multiply complex**

**{**

**complex t;**

**t.real=real\*o.real - img \*o.img;**

**t.img=img\*o.real+real\*o.img;**

**return t;**

**}**

**complex complex :: conjugate() //conjugate complex**

**{**

**complex t;**

**t.real=real;**

**t.img=-1\*img;**

**return t;**

**}**

**complex complex::div(complex o) //divide complex**

**{**

**complex t,co;**

**float d;**

**co=o.conjugate();**

**t=multiply(co);**

**d=o.real\*o.real+o.img\*o.img;**

**t.real=t.real/d;**

**t.img=t.img/d;**

**return t;**

**}**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*main function\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**int main()**

**{**

**complex c1,c2,c3;**

**int ch;**

**char z;**

**c1.read(); //reading complex1**

**c1.display();**

**c2.read(); //reading complex2**

**c2.display();**

**do{**

**//menu for complex**

**cout<<"\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";**

**cout<<"\n\t\*\tMENU FOR COMPLEX OPERATIONS\t\*\t\**

**\n\t\*\t1.addition\t\t\t\*\**

**\n\t\*\t2.subtraction\t\t\t\*\**

**\n\t\*\t3.multiply\t\t\t\*\**

**\n\t\*\t4.divide\t\t\t\*\**

**\n\t\*\t5.conjugate\t\t\t\*";**

**cout<<"\n\t\*\tenter your choice:";**

**cin>>ch;**

**cout<<"\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";**

**cout<<"\n\n";**

**switch(ch)**

**{**

**case 1:{**

**c3=c1.add(c2);**

**c3.display();**

**break;**

**}**

**case 2:{**

**c3=c1.sub(c2);**

**c3.display();**

**break;**

**}**

**case 3:{**

**c3=c1.multiply(c2);**

**c3.display();**

**break;**

**}**

**case 4:{**

**c3=c1.div(c2);**

**c3.display();**

**break;**

**}**

**case 5:{**

**cout<<"\nWhich complex numbers conjugate do you want:";**

**cout<<"\n1.";**

**c1.display();**

**cout<<"2.";**

**c2.display();**

**cout<<"your choice:";**

**cin>>ch;**

**if(ch==1)**

**{**

**c3=c1.conjugate();**

**c3.display();**

**}**

**else**

**{**

**c3=c2.conjugate();**

**c3.display();**

**}**

**break;**

**}**

**default:cout<<"invalid input!!!";**

**}**

**cout<<"\ndo you want to continue..(y/n):";**

**cin>>z;**

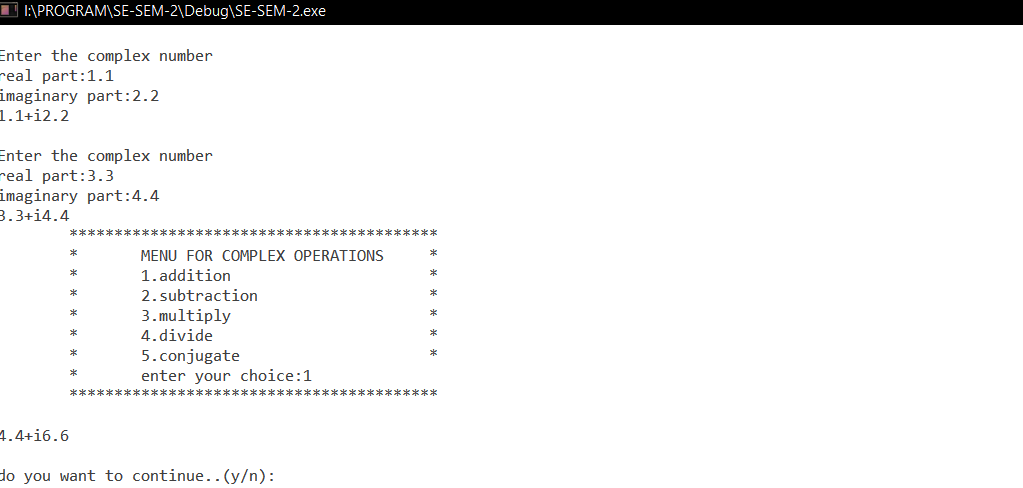
**}while(z=='y'||z=='Y');**

**return 0;**

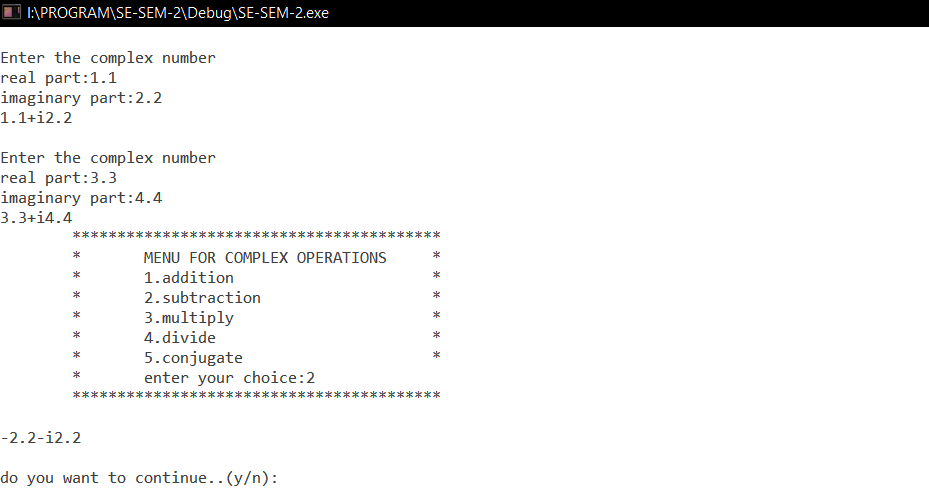
**}**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*end-of-program\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

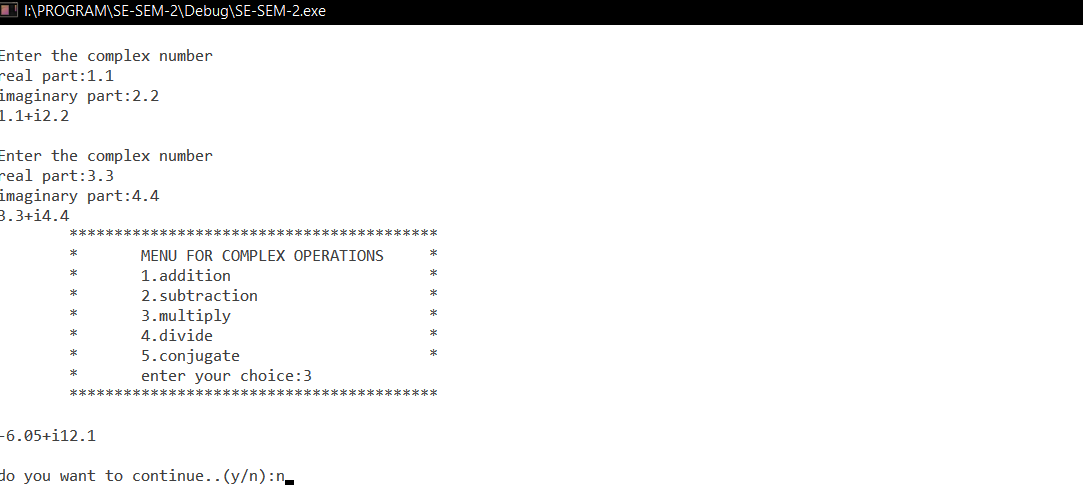
**OUTPUT: ADDITION**



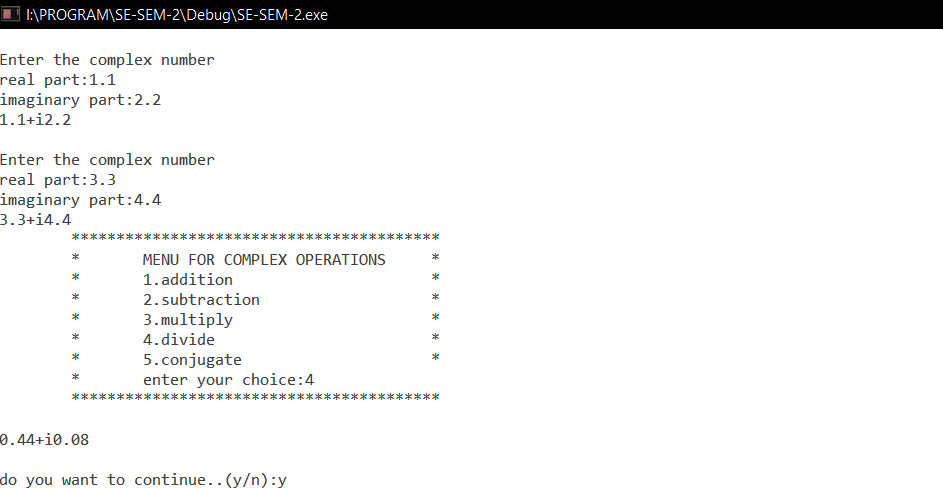
**SUBTRACTION**



**MULTIPLICATION**



**DIVISION**



**CONJUGATION**



