**C++ ASSIGNMENT 1 2023**

**NAME-- HATIM SIDHPUR WALA**

**CLASS--BTech CSE**

**SECTION--B(G2)**

**ENROLL NO--22CS002476**

**SUBJECT--PROGRAMMING AND DATA STRUCTURE USING C++**

**Problem Statement 1:**

**A company wants to create a system to manage its fleet of vehicles. The company wants to store the following information about each vehicle:**

**Make (string)**

**Model (string)**

**Year (integer)**

**The company wants to be able to perform the following actions on each vehicle:**

**Start the engine**

**Stop the engine**

**In addition, the company wants to store the following information about each car:**

**Number of doors (integer)**

**Engine capacity (float)**

**The company wants to be able to perform the following actions on each car:**

**honk the horn**

**To achieve this, create a base class named "Vehicle" in C++ to represent a vehicle. The class should have the following member functions:**

**A constructor which takes the make, model, and year as input and sets the corresponding member variables.**

**A function named "startEngine" which displays a message indicating that the engine has started.**

**A function named "stopEngine" which displays a message indicating that the engine has stopped.**

**We can then derive a class named "Car" from the "Vehicle" class. The "Car" class should have the following member variables:**

**Number of doors (integer)**

**Engine capacity (float)**

**The "Car" class should have the following member functions:**

**A constructor which takes the make, model, year, number of doors, and engine capacity as input and sets the corresponding member variables.**

**An overridden "startEngine" function which calls the base class's "startEngine" function and displays a message indicating that the car's engine has started.**

**An overridden "stopEngine" function which calls the base class's "stopEngine" function and displays a message indicating that the car's engine has stopped.**

**A function named "honkHorn" which displays a message indicating that the horn has been honked.**

**In the main function, we can create objects of the "Vehicle" and "Car" classes to represent each vehicle in the company's fleet and perform the desired actions on each vehicle.**

**Example output:**

**Vehicle make: Honda**

**Vehicle model: Civic**

**Vehicle year: 2020**

**Vehicle engine has started.**

**Vehicle engine has stopped.**

**Car make: Toyota**

**Car model: Camry**

**Car year: 2022**

**Number of doors: 4**

**Engine capacity: 2.5**

**Car engine has started.**

**Horn has been honked.**

**Car engine has stopped.**

**SOURCE CODE:**

**#include<iostream>**

**using namespace std;**

**class vehical //PARENT CLASS**

**{**

**public:**

**string make,model;**

**int year;**

**vehical()//DEFAULT CONSTRUCTOR FOR VALUE INPUT**

**{ year=0;//VALUE INITIALISATION**

**cout<<"VEHICALE MAKE--"<<endl;**

**getline(cin,make);**

**cout<<"VEHICALE MODEL--"<<endl;**

**getline(cin,model);**

**cout<<"MANUFACTURING YEAR --"<<endl;**

**cin>>year;**

**}**

**void showvehdata()**

**{**

**cout<<"VEHICALE MAKE--"<<make<<endl;**

**cout<<"VEHICALE MODEL--"<<model<<endl;**

**cout<<"MANUFACTURING YEAR --"<<year<<endl;**

**enginestart();//NESTED CALL**

**enginestop();//NESTED CALL**

**cout<<"->->->->->->->->->->->->->->->->->->->->->->->->->->->->->->->->->->->->->->->>>"<<endl;**

**}**

**void enginestart(){cout<<"ENGINE START"<<endl;}**

**void enginestop(){cout<<"ENGINE STOP"<<endl;}**

**};**

**class car:vehical //DERIVED CLASS**

**{**

**int nogate;float ecapacity;**

**public:**

**car() //DEFAULT CONSTRUCTOR FOR VALUE INITIALISATIOIN**

**{nogate=0;ecapacity=0; }**

**car(string mk,string mdl,int y,int nog,int ecap)**

**{ }//PARAMATERISED CONSTRUCTOR**

**showvehdata();//DESCRIBING INHERITANCE WHILE CALLING THE PARENT CLASS DATA FUNCTION**

**make=mk;**

**model=mdl;**

**year=y;**

**nogate=nog;**

**ecapacity=ecap;**

**}**

**void showcardata()**

**{**

**cout<<"CAR MAKE--"<<make<<endl<<"CAR MODEL--"<<model<<endl<<"MANUFACTORING YEAR--"<<year<<endl<<"NO OF GATE--"<<nogate<<endl<<"ENGINE EFFICIENCY--"<<ecapacity<<endl;**

**carenginestart();**

**presshorn();//NESTED CALL**

**carenginestop();//NESTED CALL**

**}**

**void presshorn(){cout<<"horn presssed-->>pipi\n";}**

**void carenginestart(){cout<<"CAR ENGINE STARTED";}**

**void carenginestop(){cout<<"CAR ENGINE STOPPED";}**

**};**

**int main()**

**{**

**car c2("HONDA","AMAZE",2012,5,150.3);//PARAMETRIZED CONSTRUCTOR**

**c2.showcardata();**

**return 0;**

**}**

**OUTPUT:**

**VEHICALE MAKE--**

**bajaj**

**VEHICALE MODEL--**

**pulsar**

**MANUFACTURING YEAR --**

**2022**

**VEHICALE MAKE--bajaj**

**VEHICALE MODEL--pulsar**

**MANUFACTURING YEAR --2022**

**ENGINE START**

**ENGINE STOP**

**->->->->->->->->->->->->->->->->->->->->->->->->->->->->->->->->->->->->->->->**

**ENTER DATA FOR CAR**

**CAR MAKE--HONDA**

**CAR MODEL--AMAZE**

**MANUFACTORING YEAR--2012**

**NO OF GATE--5**

**ENGINE EFFICIENCY--150**

**ENGINE START**

**horn presssed-->>pipi**

**ENGINE STOP**

**-----------------------------------------------------------------------------------------------------------------------------------------**

**Problem Statement 2:**

**Imagine you're building a program to manage a library. One aspect of the program is to keep track of the books in the library. Each book has the following attributes:**

**Title (string)**

**Author (string)**

**ISBN (string)**

**Quantity(integer)**

**Availability (bool)**

**Create a class named "Book" to store this information.**

**This class has a constructor to initialize the data members when a book object is created, as well as functions to get the information about the book and set the availability of the book, issue the book and return the book. The private data members hold the information about the book.**

**Create an array of book objects, gets information about them, sets the availability of the books, issue and return happens and then displays the information again.**

**SOURCE CODE:**

**#include<iostream>**

**using namespace std;**

**class book**

**{**

**string title,author,issue,retname,label;**

**int isbn,quntaty ,eno, a;**

**static int noofbooks; //STATIC DATA MEMBER FOR BOOK COUNT**

**public:**

**book()//DEFAULT CONSTRUCTOR**

**{**

**isbn=0;**

**quntaty=0;**

**eno=0;//ENROLLMENT NO**

**cout<<"<-<-<-<-<-----------------WELCOME TO SPSU CENTRAL LIBERARY---------------------->->->->->"<<endl;**

**cout<<"BOOKS AVAILABLE JAVA(BY GURPREET S),DMS(BY VARSHA ),C++(BY ASHOK N),PROBABLITY(BY DR ARVINF )"<<endl;**

**cout<<endl;**

**}**

**void welcome()**

**{**

**cout<<"ENTER THE BOOK NAME,AUTHOR,ISBN NO AND NO OF THE BOOKS YOU WANT "<<endl;**

**cout<<"KINDLY ENTER THE NAME AND AUTHOR IN SMALL LETTERS "<<endl;**

**}**

**void getdata()**

**{ getline(cin,title);**

**// cin.ignore();//BUFFER**

**getline(cin,author);**

**cin>>isbn>>quntaty;**

**cout<<"SORRY DUE TO RULES AND REGULATION ONLY ONE BOOK GET ISSSUED AT A TIME"<<endl;**

**}**

**void issuing(void)**

**{**

**cout<<"TYPE ISSUE FOR THE ISSSUING OR READ FOR READING "<<endl;**

**cout<<"---------------------------------------------------------------------------"<<endl;**

**cin>>issue;**

**if(issue=="issue")**

**{**

**cout<<"ENTER THE ENROLL NO TO ISSUE"<<endl;**

**cin>>eno;**

**cout<<"THE BOOK HAS BEEN ISSUED KINDLY RETURN AFTER 5 DAYS "<<endl;**

**returnbook();**

**}**

**else**

**{**

**cout<<"THANK YOU FOR READING "<<endl;**

**}**

**}**

**void showdata()**

**{ welcome();**

**getdata();//NO OF BOOKS AVAILABLE**

**if(title=="Cplus plus" && author=="ashok n" && isbn==7800701)**

**{**

**if(noofbooks==0)**

**{**

**cout<<"SOORY THE BOOK IS NOT AVAILABLE";**

**again();**

**}**

**cout<<"BOOK NAME--ANSI C++ "<<endl;**

**cout<<"AUTHOR NAME--ASHOK N"<<endl;**

**cout<<"ISBN NO--7800701"<<endl;**

**cout<<"QUiENTITY--3"<<endl;**

**cout<<"NO OF BOOK REMAINING--"<<noofbooks<<endl;**

**noofbooks--;**

**}**

**else if(title=="java" && author=="gurpreet s" && isbn==587669)**

**{**

**if(noofbooks==0)**

**{**

**cout<<"SOORY THE BOOK IS NOT AVAILABLE";**

**again();**

**}**

**cout<<"BOOK NAME -- PROGRAMMING WITH JAVA"<<endl;**

**cout<<"AUTHOR NAME-- MR GURPREET SINGH"<<endl;**

**cout<<"ISBN NO--587669"<<endl;**

**cout<<"QUIENTITY--4"<<endl;**

**cout<<"NO OF BOOK REMAINING--"<<noofbooks<<endl;**

**noofbooks--;**

**}**

**else if(title=="dms" && author=="varsha" && isbn==5465987)**

**{**

**if(noofbooks==0)**

**{**

**cout<<"SOORY THE BOOK IS NOT AVAILABLE";**

**again();**

**}**

**cout<<"BOOK NAME -- DISCREATE MATHAMATICS"<<endl;**

**cout<<"AUTHOR NAME--VARSHA PATIL"<<endl;**

**cout<<"ISBN NO--5465987"<<endl;**

**cout<<"QUIENTITY--2"<<endl;**

**cout<<"NO OF BOOK REMAINING--"<<noofbooks<<endl;**

**noofbooks--;**

**}**

**else if(title=="probablity" && author=="arvind" && isbn==8897845)**

**{**

**if(noofbooks==0)**

**{**

**cout<<"SOORY THE BOOK IS NOT AVAILABLE";**

**again();**

**} cout<<"BOOK NAME -- PROBABLITY MATHAMATICS"<<endl;**

**cout<<"AUTHOR NAME--DR ARVIND SHARMA"<<endl;**

**cout<<"ISBN NO--8897845"<<endl;**

**cout<<"QUIENTITY--3"<<endl;**

**cout<<"NO OF BOOK REMAINING--"<<noofbooks<<endl;**

**noofbooks--;**

**}**

**else**

**cout<<"THE BOOK IS NOT AVAILABLE KINDLY CONTACT DEEN OFFICE"<<endl;**

**issuing();**

**}**

**void returnbook()//FUNCTION TO RETURN BOOK**

**{**

**cout<<"KINDLY ENTER THE ENROLLMENT NO AND BOOK NAME TO RETURN "<<endl;**

**cin>>eno;**

**cin.ignore();**

**getline(cin,retname);**

**cout<<"THANK YOU THE BOOK HAS BEEN RETURNED"<<endl;**

**again();**

**}**

**void again()//FUNCTON IF USER WANT ANATHOR BOOK**

**{ cout<<"WANT ANATHOR BOOK "<<endl;**

**cout<<"ENTER 1 FOR CONTINUE AND 0 FOR EXIT"<<endl;**

**cin>>a;**

**if(a==1)**

**{showdata();}**

**else{**

**cout<<"thank you for visiting come again"<<endl;}**

**}**

**};**

**int book::noofbooks=3;//STATIC DATA MEMBER INITIALISATION**

**int main()**

**{**

**book b1;//OBJECT DECLARATION**

**b1.showdata();**

**return 0;**

**}**

**OUTPUT:**

**<-<-<-<-<-----------------WELCOME TO SPSU CENTRAL LIBERARY---------------------->->->->->**

**BOOKS AVAILABLE JAVA(BY GURPREET S),DMS(BY VARSHA ),C++(BY ASHOK N),PROBABLITY(BY DR ARVIND )**

**ENTER THE BOOK NAME,AUTHOR,ISBN NO AND NO OF THE BOOKS YOU WANT**

**KINDLY ENTER THE NAME AND AUTHOR IN SMALL LETTERS**

**java**

**gurpreet s**

**587669**

**2**

**SORRY DUE TO RULES AND REGULATION ONLY ONE BOOK GET ISSSUED AT A TIME**

**BOOK NAME -- PROGRAMMING WITH JAVA**

**AUTHOR NAME-- MR GURPREET SINGH**

**ISBN NO--587669**

**QUIENTITY--4**

**NO OF BOOK REMAINING--3**

**TYPE ISSUE FOR THE ISSSUING OR READ FOR READING**

**---------------------------------------------------------------------------**

**issue**

**ENTER THE ENROLL NO TO ISSUE**

**123445**

**THE BOOK HAS BEEN ISSUED KINDLY RETURN AFTER 5 DAYS**

**KINDLY ENTER THE ENROLLMENT NO AND BOOK NAME TO RETURN**

**1234**

**java**

**THANK YOU THE BOOK HAS BEEN RETURNED**

**WANT ANATHOR BOOK**

**ENTER 1 FOR CONTINUE AND 0 FOR EXIT**

**1**

**ENTER THE BOOK NAME,AUTHOR,ISBN NO AND NO OF THE BOOKS YOU WANT**

**KINDLY ENTER THE NAME AND AUTHOR IN SMALL LETTERS**

**-----------------------------------------------------------------------------------------------------------------------------------**

**Problem Statement 3:**

**Create a class named "Date" which will store a date in the format of day, month, and year. The class should have three data members named "day", "month", and "year".**

**The class should also have the following member functions:**

**A constructor which takes the day, month, and year as input and sets the data members to the given values.**

**A function named "displayDate" which will display the date in the format of "dd/mm/yyyy".**

**A function named "incrementDay" which will increase the day by 1 and adjust the month and year as needed (e.g. if the day was 31 and the month was December, the day should become 1 and the month should become January and the year should be increased by 1). [Check if the year is a leap year or not, to handle use case for the month of February]**

**Additionally, create a main function which will create an object of the "Date" class, set the date to 31st December 2022, increment the day by 1, and display the date.**

**Example output:**

**The date is: 01/01/2023**

**SOURCE CODE:**

**#include <iostream>**

**using namespace std;**

**class date2**

**{**

**int date,month,year;**

**public:**

**date2()**

**{ }**

**date2(int d,int m,int y)//PARAMETRSED CONSTRUCTOR FOR VALUE INPUT**

**{**

**date =d;**

**month=m;**

**year=y;**

**}**

**void displaydate()//FUNCTION TO DISPLAY THE CURRENT DATE**

**{ cout<<date<<"/"<<month<<"/"<<year<<endl;}**

**void increamentdate()**

**{**

**if(month>=1 && month<=8)//ODD NO OF MONTHS ARE HAVING 31 DAYS EX1,3,5,7 AND EVEN NO OF MONTHS HAVIN 30 DAYS**

**{**

**if(month==2)//CHECK MONTH IS FEBURARY**

**{**

**if(year%400==0 ||year%4==0)//CHECK FOR THE LEEP YEAR HENCE FEBURARY IS HAVING 29 DAYS**

**{**

**if (date >=1 && date<29)**

**{**

**date=date+1;**

**cout<<date<<"/"<<month<<"/"<<year<<endl;**

**}**

**else if(date==29)**

**{**

**date=1;**

**month=month+1;**

**cout<<date<<"/"<<month<<"/"<<year<<endl;**

**}**

**else{cout<<"invalid date ";}**

**}**

**else if(date>=1 && date<=28)//NORMAL FEBURARY HAVING NO LEEP YEAR**

**{**

**if(date==28)**

**{**

**date=1;**

**month=month+1;**

**cout<<date<<"/"<<month<<"/"<<year<<endl;**

**}**

**else if(date>=1 && date<28)**

**{**

**date=date+1;**

**cout<<date<<"/"<<month<<"/"<<year<<endl;**

**}**

**}**

**else{cout<<"invalid date kindly enter the correct date ";}**

**}**

**else if(month%2!=0)//CHECK THE MONTH IS ODD**

**{ if(date>1 && date<=31)**

**{**

**if(date==31)**

**{**

**date=1;**

**month=month+1;**

**cout<<date<< "/"<<month<<"/"<<year<<endl;**

**}**

**else**

**{ date=date+1;**

**cout<<date<<"/"<<month<<"/"<<year<<endl;**

**}**

**}**

**else{cout<<"invalid date ";}**

**}**

**else if(month%2==0)//CHECK THE MONTH IS EVEN**

**{**

**if(date>=1 && date<=30)**

**{**

**if(date==30)**

**{**

**date=1;**

**month=month+1;**

**cout<<date<< "/"<<month<<"/"<<year;**

**}**

**else**

**{ date=date+1;**

**cout<<date<<"/"<<month<<"/"<<year;**

**}**

**}**

**else{cout<<"invalid date ";}**

**}**

**}**

**else if (month>8 && month <=12)//HENCE JULY AND AUGUST HAVING 31 DAYS TO DIFFERENTIATE THE USED TO AREAS 1-8 AND 9-12**

**{**

**if(month%2!=0)**

**{ if(date>1 && date<=31)**

**{**

**if(date==31)**

**{**

**date=1;**

**month=month+1;**

**cout<<date<< "/"<<month<<"/"<<year<<endl;**

**}**

**else**

**{ date=date+1;**

**cout<<date<<"/"<<month<<"/"<<year<<endl;**

**}**

**}**

**else{cout<<"invalid date ";}**

**}**

**else**

**{**

**if(month==12)//CHECK THE LAST MONTH**

**{ if(date==31)**

**{**

**date=1;**

**month=1;**

**year=year+1;**

**cout<<date<<"/"<<month<<"/"<<year<<endl;**

**}**

**else**

**{**

**date=date+1;**

**cout<<date<<"/"<<month<<"/"<<year<<endl;**

**}**

**}**

**else if(date>1 && date<=30)**

**{**

**if(date==30)**

**{**

**date=1;**

**month=month+1;**

**cout<<date<< "/"<<month<<"/"<<year<<endl;**

**}**

**else**

**{ date=date+1;**

**cout<<date<<"/"<<month<<"/"<<year<<endl;**

**}**

**}**

**else{cout<<"invalid date ";}**

**}**

**}**

**else{cout<<"invalid month";}**

**}**

**};**

**int main()**

**{**

**int d,m,y;**

**date2 d1;**

**cout<<"enter the date"<<endl;**

**cin>>d>>m>>y;**

**date2 d2(d,m,y);**

**cout<<"the date entered is"<<endl;**

**d2.displaydate();**

**cout<<"the next date is"<<endl;**

**d2.increamentdate();**

**}**

**OUTPUT:**

**enter the date**

**31**

**12**

**2022**

**the date entered is**

**31/12/2020**

**the next date is**

**01/01/2023**

**-----------------------------------------------------------------------------------------------------------------------------------**

**Problem Statement 4:**

**Create two classes named "Rectangle" and "Square". The "Rectangle" class will have two data members named "length" and "width" and a member function named "area" which will return the area of the rectangle.**

**The "Square" class will have a single data member named "side" and a member function named "area" which will return the area of the square.**

**Make the "Square" class a friend of the "Rectangle" class. This will allow the "Square" class to access the private data members of the "Rectangle" class.**

**Additionally, create a main function which will create an object of the "Square" class, set the side to 5, and display the area of a rectangle with the same dimensions as the square.**

**Example output:**

**The area of a rectangle with the same dimensions as the square is: 25**

**SOURCE CODE**

**#include<iostream>**

**using namespace std;**

**class rectangle**

**{**

**int length;**

**int bredth;**

**public:**

**rectangle(){length=0;bredth=0;}//CONSTRUCTOR TO INITIALISE THE VALUE**

**void rectarea();**

**friend class square;//DECLARATION OF FRIEND CLASS**

**};**

**void rectangle::rectarea()//TO CALCULATE THE AREA OF RECTANGLE**

**{**

**cout<<endl<<"enter the length and breadth of the reactange "<<endl;**

**cin>>length>>bredth;**

**cout<<endl<<"the area of rectangle "<<endl<<length\*bredth;**

**}**

**class square//FRIEND CLASS WHICH ACCESS THE DATA MEMBER**

**{ int side;**

**public://RETURN THE AREA OF SQUARE**

**void squarearea(){**

**cout<<endl<<"enter the side of the squre to calculate the area "<<endl;**

**cin>>side;**

**cout<<endl<<"the area of square"<<endl<<(side\*side)<<endl;**

**samearea();**

**}**

**void samearea()//RETURN THE AREA OF RECTANGLE WITH SAME DIMENSION AS SQUARE**

**{ rectangle r;**

**r.rectarea();**

**cout<<endl<<"enter the dimension to get the same area of reactangle and square"<<endl;**

**cin>>r.length;**

**r.length=r.bredth;**

**side=r.length;**

**cout<<endl<<"the area of reactangle with the same dimension as the square is"<<endl<<side\*side<<endl;**

**}**

**};**

**int main()**

**{ square s1;**

**s1.squarearea();**

**//s1.samearea(r1);**

**return 0;**

**}**

**OUTPUT:**

**enter the side of the squre to calculate the area**

**10**

**the area of square**

**100**

**enter the length and breadth of the reactange**

**10**

**2**

**the area of rectangle**

**20**

**enter the dimension to get the same area of reactangle and square**

**5**

**the area of reactangle with the same dimension as the square is**

**25**

**---------------------------------------------------------------------------------------------------------------------------**

**Problem Statement 5:**

**Create a class named "Complex" which will store a complex number in the form of real and imaginary parts. The class should have two data members named "real" and "imag" and two friend functions named "add" and "multiply".**

**The "add" function should take two Complex objects as input and return a Complex object representing the sum of the two Complex numbers.**

**The "multiply" function should take two Complex objects as input and return a Complex object representing the product of the two Complex numbers.**

**Additionally, create a main function which will create two Complex objects, perform addition and multiplication of the two Complex numbers and display the results.**

**Example output:**

**The sum of the two complex numbers is: (5 + 7i)**

**The product of the two complex numbers is: (-1 + 35i)**

**SOURCE CODE:**

**#include<iostream>**

**using namespace std;**

**class complex**

**{**

**int real1,imag1;//IMAGINARY AND REAL PART OF FIRST COMPLEX NUMBER**

**int real2,imag2;//IMAGINARY AND REAL PART OF SECOND COMPLEX NUMBER**

**public:**

**friend void sumcomplex(complex s);//FRIEND FUNCTION FOR COMPLEX ADDITION**

**friend void multcomplex(complex m);//FRIEND FUNCTION FOR COMPLEX MULTIPLICATIOIN**

**};**

**void sumcomplex(complex s)**

**{ cout<<"do not disturb complex addition is going on"<<endl;**

**cout<<"enter the real and imaginary part of 1st c number"<<endl;**

**cin>>s.real1>>s.imag1;**

**cout<<"enter the real and imaginary part of the 2nd complex number"<<endl;**

**cin>>s.real2>>s.imag2;**

**cout<<"the sum of the two complex number is"<<endl;**

**cout<<"["<<(s.real1+s.real2)<<"+"<<(s.imag1+s.imag2)<<"i"<<"]"<<endl;**

**}**

**void multcomplex(complex m)**

**{ cout<<"do not disturb complex multiplication is going on"<<endl;**

**cout<<"enter the real and imaginary part of 1st c number"<<endl;**

**cin>>m.real1>>m.imag1;**

**cout<<"enter the real and imaginary part of the 2nd complex number"<<endl;**

**cin>>m.real2>>m.imag2;**

**cout<<"the sum of the two complex number is"<<endl;**

**cout<<"the multiplication of two complex number is";**

**cout<<"["<<((m.real1\*m.real2)-(m.imag1\*m.imag2))<<"+"<<"i"<<((m.real1\*m.imag2)+(m.real1\*m.real2))<<"]"<<endl;**

**}**

**int main()**

**{**

**complex c1;**

**sumcomplex(c1);//CALLING THE FRIEND FUNCTION WITHOUT OBJECT BECAUSE THEY ARE NOT THE PART OF THE CLASS**

**multcomplex(c1);**

**return 0;**

**}**

**OUTPUT:**

**do not disturb complex addition is going on**

**enter the real and imaginary part of 1st c number**

**10**

**20**

**enter the real and imaginary part of the 2nd complex number**

**5**

**4**

**the sum of the two complex number is**

**[15+24i]**

**do not disturb complex multiplication is going on**

**enter the real and imaginary part of 1st c number**

**4**

**5**

**enter the real and imaginary part of the 2nd complex number**

**6**

**8**

**the sum of the two complex number is**

**the multiplication of two complex number is[-16+i56]**

**-------------------------------------------------------------------------------------------------------------------------------**