

M.Sc. (Five Year Integrated) in Computer Science (Artificial Intelligence & Data Science)

First Semester

Laboratory Record

21-805-0107: C++ PROGRAMMING LAB

*Submitted in partial fulfillment
of the requirements for the award of degree in
Master of Science (Five Year Integrated)
in Computer Science (Artificial Intelligence & Data Science) of
Cochin University of Science and Technology (CUSAT)
Kochi*



Submitted by

HIRA MOHAMMED K
(80521011)

DEPARTMENT OF COMPUTER SCIENCE
COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY (CUSAT)
KOCHI-682022

MARCH 2022

DEPARTMENT OF COMPUTER SCIENCE
COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY (CUSAT)
KOCHI, KERALA-682022



*This is to certify that the software laboratory record for **21-805-0107: C++ Programming Lab** is a record of work carried out by **HIRA MOHAMMED K(80521011)**, in partial fulfillment of the requirements for the award of degree in **Master of Science (Five Year Integrated) in Computer Science (Artificial Intelligence & Data Science)** of Cochin University of Science and Technology (CUSAT), Kochi. The lab record has been approved as it satisfies the academic requirements in respect of the first semester laboratory prescribed for the Master of Science (Five Year Integrated) in Computer Science degree.*

Faculty Member in-charge

Dr. Madhu S. Nair
Professor
Department of Computer Science
CUSAT

Dr. Philip Samuel
Professor and Head
Department of Computer Science
CUSAT

Table of Contents

Sl.No.	Program	Pg.No.
1	Program to Calculate Students Grade	pg 1
2	Program to find area using Overloaded Functions	pg 5
3	Program using Classes for Bank Transactions of ' <i>N</i> ' customers	pg 10
4	Program to perform Operations on String Objects	pg 17
5	Program to demonstrate execution order of Constructors & Destructors	pg 22
6	Program to perform addition on Time Class objects	pg 26
7	Program to perform operations on Matrix Class	pg 29
8	Program to invoke Complex Class objects using constructor overloading	pg 37
9	Program to design and implement Static Member Functions	pg 40
10	Program to process Departmental Store list and perform operations	pg 43
11	Program to Swap Private Data Members of classes using Friend Function	pg 52
12	Program to perform addition on Complex Class Objects	pg 55
13	Program to Overload Comparison operators for a Vector Object	pg 58
14	Program to Overload Operators for Complex objects using Friend function	pg 66
15	Program to overload operators like *,<,> for a Vector Object	pg 70
16	Program to Overload '+' and '*' operators for a Matrix Class	pg 73
17	Program to demonstrate Multiple and Multilevel Inheritance	pg 79
18	Program to demonstrate Virtual Base class and Hybrid Inheritance	pg 84
19	Program to show order of execution of Constructors for Multiple Inheritance	pg 88
20	Program to find areas of shapes by Run Time Polymorphism and Abstract Classes	pg 91
21	Program to demonstrate use of Pure Virtual Functions	pg 97
22	Program to demonstrate use of Class Templates	pg 101
23	Program to demonstrate use of Exception Handling	pg 108

STUDENT GRADES

AIM

To calculate the grades of a list of students with attributes(Name, Roll_no, Marks of 3 subjects) using class with member functions input(), calcGrade(), display().

PROGRAM

```
#include<iostream>
#include<cmath>
using namespace std ;
class student
{
    string Name ;
    int Roll_no ;
    float m1,m2,m3;
public:
    void input(void);
    void calcGrade(void);
    void display(void);
};

void student::input(void)
{
    cout << "Enter the name of the student : " ;
    cin >> Name ;
    cout << "Enter the roll no : ";
    cin >> Roll_no ;
    cout << "Enter marks scored in subject 1 ";
    cin >> m1 ;
    cout << "Enter marks scored in subject 2 " ;        cin >> m2 ;
    cout << "Enter marks scored  in subject 3 ";
    cin >> m3 ;
    cout << "\n";
    cout<<"\n";
}

void student::calcGrade(void)
{
    float average = (m1 + m2 + m3) / 3;
    if(average>=90)
```

```
{
    cout<<"A";
}
else if(average>=80 and average<90)
{
    cout<<"B";
}
else if(average>=70 and average<80)
{
    cout<<"C";
}
else if(average>=60 and average<70)
{
    cout<<"D";
}
else if(average>=50 and average<60)
{
    cout<<"E";
}
else
{
    cout<<"F";
}
}

void student::display(void)
{

    cout << "Name      :" << Name <<endl;
    cout << "Roll No  :" << Roll_no <<endl ;
    cout<<"Mark attained in subject 1: " << m1 <<endl;
    cout<<"Mark attained in subject 2: " << m2 <<endl;
    cout<<"Mark attained in subject 3: " << m3 <<endl;
    cout << "Grade attained  :" ;
    calcGrade();
    cout << "\n" ;
}

int main()
{
    student s[100];
    int n,i;
```

```
int option,ch,choice;
cout<<"Select an option " <<endl;
cout<<"1:Start" << endl;
cout<<"2:Quit" << endl;
cin>>ch;
if(ch==2)
{
    return 0;
}
do
{
    cout<<"Enter number of students: ";
    cin >> n;
    for(i=0;i<n;i++)
    {
        cout <<"Student " <<i+1 <<endl;
        s[i].input();
    }
    cout<<"Do you want to display details? " << endl;
    cout<<"1:Yes" << endl;
    cout<<"2:No " << endl;
    cin>>option;
    if(choice!=2)
    {
        for(i=0;i<n;i++)
        {
            cout <<"Student " << i+ 1<<" details" <<endl;
            s[i].display();
            cout<<"\n";
        }
        cout <<"Do you want to continue? ";
        cout <<"Press 1 for yes and 0 for no: ";
        cin>>option;
    }
}while(option);
return 0 ;
}
```

SAMPLE INPUT-OUTPUT

```
Select an option
1:Start
2:Quit
1
Enter number of students: 2
Student 1
Enter the name of the student : Sam
Enter the roll no : 1
Enter marks scored in subject 1 90
Enter marks scored in subject 2 80
Enter marks scored in subject 3 85

Student 2
Enter the name of the student : Ravi
Enter the roll no : 2
Enter marks scored in subject 1 70
Enter marks scored in subject 2 75
Enter marks scored in subject 3 60

Do you want to display details?
1:Yes
2:No
1
Student 1 details
Name :Sam
Roll No :1
Mark attained in subject 1: 90
Mark attained in subject 2: 80
Mark attained in subject 3: 85
Grade attained :B

Student 2 details
Name :Ravi
Roll No :2
Mark attained in subject 1: 70
Mark attained in subject 2: 75
Mark attained in subject 3: 60
Grade attained :D

Do you want to continue? Press 1 for yes and 0 for no: 0
```

OVERLOADING FUNCTION TO FIND AREA

AIM

To calculate the area of different shapes like Rectangle, Square etc (at least 5 shapes) using overloaded function area()

PROGRAM

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

int s,l,b;
float r,ab,bc,ca,ra,h;
int area(int);
int area(int,int);
float area(float);
float area(float,float);
float area(float,float,float);

int area(int s)
{
    cout<<"Enter side of a square:";
    cin>>s;
    cout <<"Area of square is: ";
    return(s*s);
}

int area(int l,int b)
{
    cout<<"Enter length and breadth of rectangle:";
    cin>>l>>b;
    cout <<"Area of rectangle is: ";
    return(l*b);
}

float area(float r)
{
    cout<<"Enter radius of circle:";
    cin>>r;
    cout <<"Area of circle is: ";
    return(3.14*r*r);
}
```



```
float area(float ra, float h)
{
    cout<<"Enter radius and height of cylinder: ";
    cin >>ra >>h;
    cout <<"Area of cylinder is: ";
    return(2*(3.14)*ra*(ra+h));
}

float area(float ab,float bc,float ca)
{

    cout<<"Enter sides of triangle:";
    cin>>ab>>bc>>ca;
    float p,x;
    p=(ab+bc+ca)/2;
    x = p*(p-ab)*(p-bc)*(p-ca);
    cout <<"Area of triangle is: ";
    return(sqrt(x));

}

int main()
{
    int option;
    int ch;
    cout<<"\tWELCOME\t" << endl;
    do
    {
        cout <<"\n\n";
        cout<<"Select an option and enter corresponding number " <<endl;
        cout<<"1:Area of square" <<endl;
        cout <<"2:Area of rectangle" <<endl;
        cout <<"3:Area of circle" <<endl;
        cout<<"4:Area of cylinder" <<endl;
        cout<<"5:Area of triangle" <<endl;
        cout<<"6:Quit" << endl;
        cout<<"Enter your choice: ";
        cin >>ch;

        switch(ch)
        {
```

```
        case 1: cout << area(s);
                break;
        case 2: cout << area( l,b);
                break;
        case 3: cout << area(r);
                break;
        case 4: cout << area(ra,h);
                break;
        case 5 :cout << area(ab,bc,ca);
                break;

    }

    cout<<"\n\n";
    cout<<"Do you want to continue ? " <<endl;
    cout <<"Press 1 for yes and 0 for no: ";
    cin >>option;
}while(option);

return 0 ;

}
```

SAMPLE INPUT-OUTPUT

```
WELCOME

Select an option and enter corresponding number
1:Area of square
2:Area of rectangle
3:Area of circle
4:Area of cylinder
5:Area of triangle
6:Quit
Enter your choice: 1
Enter side of a square:4
Area of square is: 16

Do you want to continue ?
Press 1 for yes and 0 for no: 1

Select an option and enter corresponding number
1:Area of square
2:Area of rectangle
3:Area of circle
4:Area of cylinder
5:Area of triangle
6:Quit
Enter your choice: 2
Enter length and breadth of rectangle:6 3
Area of rectangle is: 18

Do you want to continue ?
Press 1 for yes and 0 for no: 1

Select an option and enter corresponding number
1:Area of square
2:Area of rectangle
3:Area of circle
4:Area of cylinder
5:Area of triangle
6:Quit
Enter your choice: 3
Enter radius of circle:10
Area of circle is: 314
```

```
Do you want to continue ?
Press 1 for yes and 0 for no: 1

Select an option and enter corresponding number
1:Area of square
2:Area of rectangle
3:Area of circle
4:Area of cylinder
5:Area of triangle
6:Quit
Enter your choice: 4
Enter radius and height of cylinder: 5 10
Area of cylinder is: 471

Do you want to continue ?
Press 1 for yes and 0 for no: 1

Select an option and enter corresponding number
1:Area of square
2:Area of rectangle
3:Area of circle
4:Area of cylinder
5:Area of triangle
6:Quit
Enter your choice: 5
Enter sides of triangle:3 4 5
Area of triangle is: 6

Do you want to continue ?
Press 1 for yes and 0 for no: 0
```

BANK TRANSACTIONS

AIM

Program using classes to perform bank transaction for n customers (cust_name, acc_no, acc_type, balance). The program should be menu driven and it should have the following menus like adding new account, withdraw (keep a min balance of 500), deposit, balance enquiry and account statement (cust_name, acc_no, acc_type, balance)

PROGRAM

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

class bank
{
    char cust_name[30];
    int acc_no;
    char acc_type[20];
    float balance;

public:
    void details();
    void deposit();
    void withdraw();
    void acc_balance();
    void display(int);
};

void bank::details()
{
    cout << "Enter your name: ";
    cin >> cust_name;
    cout << "Enter type of account(s/c): ";
    cin >> acc_type;
    cout << "Enter amount of money want to deposit: ";
    cin >> balance;
    cout<<"Account created successfully: " <<endl;
```

```
}  
void bank::deposit()  
{  
    float amount;  
  
    cout << "Enter amount to be deposited";  
    cin >> amount;  
  
    balance= balance+amount;  
    cout <<"Amount credited successfully" <<endl;  
  
}  
void bank::withdraw()  
{  
    float amt;  
  
    cout << "Enter amount to be withdrawn " << endl;  
    cin >> amt;  
    if((balance-amt) >=500)  
    {  
        balance=balance-amt;  
        cout <<"Amount debited successfully" << endl;  
    }  
    else  
    {  
        cout << "Insufficient balance " << endl;  
    }  
  
}  
void bank::acc_balance()  
{  
    cout <<"Balance is " << balance <<endl;  
  
}  
void bank::display(int x)  
{  
    cout <<"Name : " << cust_name <<endl;  
    cout <<"Account number: " << x <<endl;  
    cout <<"Account type : " << acc_type <<endl;
```

```
        cout <<"Balance : " << balance << endl;
    }

int main()
{

    int option;

    bank *customer=new bank[100];
    int ch;
    int i=0;
    int acc_no;
    int count;

    do
    {

        cout <<"\tWELCOME\t" << endl;
        cout <<"Enter your choice"<<endl;
        cout <<"1:Add an account " << endl;
        cout <<"2:Deposit amount" <<endl;
        cout <<"3:Withdraw amount" <<endl;
        cout<<"4:Balance Enquiry" <<endl;
        cout <<"5:Account statement" <<endl;

        cin >> ch;

        if(ch==1)
        {
            customer[i].details();
            cout <<"Account number: " << i <<endl;
            i=i+1;
        }
        else
        {
            cout <<"Enter account number:  " <<endl;
            cin >> acc_no;
            if(acc_no>i-1)
```

```
        {
            cout<<"This account number does not exist" <<endl;
            cout<<"Try again" <<endl;
        }
        else
        {
            switch(ch)
            {
                case 2:customer[acc_no].deposit();break;
                case 3:customer[acc_no].withdraw();break;
                case 4:customer[acc_no].acc_balance();break;
                case 5:customer[acc_no].display(acc_no);break;
            }
        }
    }

    cout<<"Do you want to continue? " <<endl;
    cout<<"1:Continue" <<endl;
    cout<<"0:Exit" <<endl;
    cin >>option;
}while(option);

return 0;

}
```


SAMPLE INPUT-OUTPUT

```
WELCOME
Select an option
1:Add an account
2:Deposit amount
3:Withdraw amount
4:Balance Enquiry
5:Account statement

Enter your choice: 1

Enter your name: Diya
Enter type of account(s/c): s
Enter amount of money want to deposit: 1000
Account created successfully
Account number: 0

Do you want to continue?
1:Yes
2:No
Enter your choice: 1

WELCOME
Select an option
1:Add an account
2:Deposit amount
3:Withdraw amount
4:Balance Enquiry
5:Account statement

Enter your choice: 1

Enter your name: Hira
Enter type of account(s/c): s
Enter amount of money want to deposit: 2000
Account created successfully
Account number: 1

Do you want to continue?
1:Yes
2:No
Enter your choice: 1
```

```
WELCOME
Select an option
1:Add an account
2:Deposit amount
3:Withdraw amount
4:Balance Enquiry
5:Account statement

Enter your choice: 2

Enter account number: 1
Enter amount to be deposited: 1000
Amount credited successfully

Do you want to continue?
1:Yes
2:No
Enter your choice: 1

WELCOME
Select an option
1:Add an account
2:Deposit amount
3:Withdraw amount
4:Balance Enquiry
5:Account statement

Enter your choice: 4

Enter account number: 1
Balance is 3000

Do you want to continue?
1:Yes
2:No
Enter your choice: 1
```

```
WELCOME
Select an option
1:Add an account
2:Deposit amount
3:Withdraw amount
4:Balance Enquiry
5:Account statement

Enter your choice: 3

Enter account number: 1
Enter amount to be withdrawn: 500
Amount debited successfully

Do you want to continue?
1:Yes
2:No
Enter your choice: 1

WELCOME
Select an option
1:Add an account
2:Deposit amount
3:Withdraw amount
4:Balance Enquiry
5:Account statement

Enter your choice: 5

Enter account number: 1

ACCOUNT STATEMENT
Name : Hira
Account number: 1
Account type : s
Balance : 2500

Do you want to continue?
1:Yes
2:No
Enter your choice: 2
```

STRING OPERATIONS

AIM

Program to perform operations such as compare, concatenate and length on String objects

PROGRAM

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

class String
{
    char *name;
    int length;
public:
    String()
    {
        length = 0;
        name = new char[length + 1];
    }

    void getstring();
    void show(void);
    void add(String &a, String &b);
    void compare(String &a,String &b);
    void stringlength(void);

};

void String::getstring()
{
    cin >> name;
}

void String::show()
{
    cout << name;
}

void String :: add(String &a, String &b)
{
    length = a.length + b.length;
```

```
        delete name;
        name = new char[length + 1];
        strcpy(name, a.name);
        strcat(name, b.name);
};

void String::compare(String &a,String &b)
{
    int m = strlen(a.name);
    int n = strlen(b.name);
    if (m < n)
    {
        cout << "String 1 smaller than string 2";
    }
    else if(m > n)
    {
        cout << "String 2 is smaller than string 1";
    }
    else
    {
        cout <<"Both strings have equal lengths";
    }
}

void String::stringlength()
{
    cout<< strlen(name) << endl;
}

int main()
{
    String s1,s2,s3;
    int ch;
    int option;
    cout<<"\tWELCOME\t" << endl;

    do
    {
        cout <<"Enter string 1: ";
        s1.getstring();
        cout <<"Enter string 2: ";
        s2.getstring();
        cout <<"\n\n";
        cout << "string 1:" ;
```

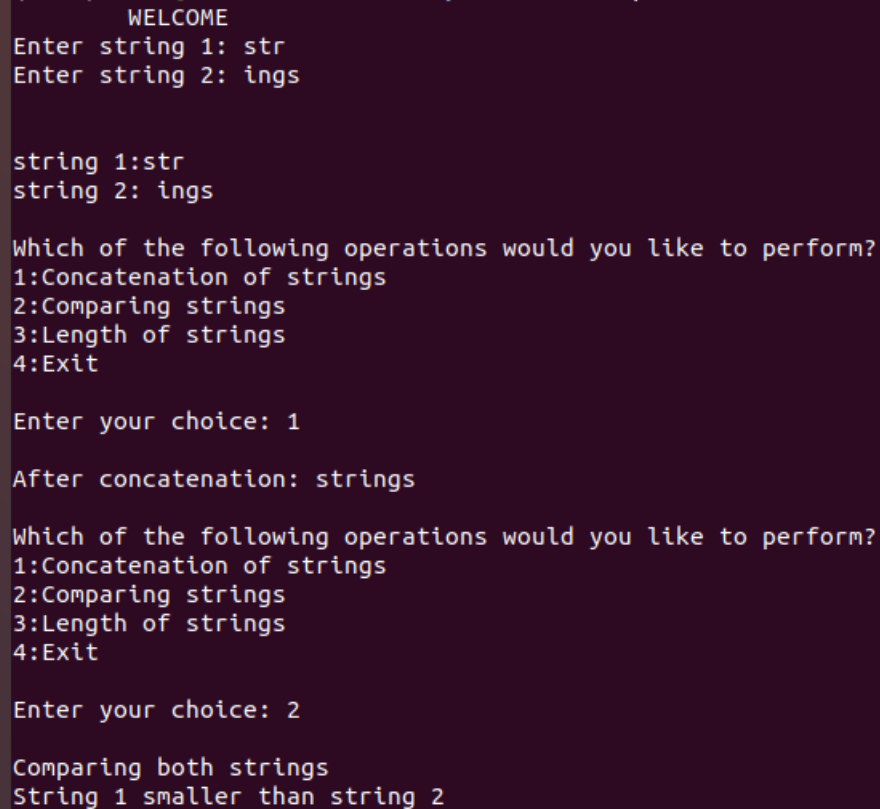
```
s1.show();
cout<<endl;
cout<<"string 2: ";
s2.show();
cout<<endl;
cout << "\n";
do
{
    cout<<"Which of the following operations would you like to perform?"<<endl;
    cout<<"1:Concatenation of strings" <<endl;
    cout<<"2:Comparing strings" <<endl;
    cout<<"3:Length of strings" <<endl;
    cout<<"4:Exit" <<endl;
    cout<<"\n";
    cout<<"Enter your choice: ";
    cin>>ch;
    switch(ch)
    {
        case 1:
            cout<<"\n";
            cout <<"After concatenation: ";
            s3.add(s1,s2);
            s3.show();
            cout<<"\n\n";
            break;
        case 2:
            cout<<"\n";
            cout<<"Comparing both strings "<< endl;
            s3.compare(s1,s2);
            cout<<"\n\n";
            break;
        case 3:
            cout<<"\n";
            cout<<"Length of string 1 is ";
            s1.stringlength();
            cout<<"\n";
            cout<<"Length of string 2 is ";
            s2.stringlength();
            cout<<"\n";
            break;
        case 4:
```

```
                break;
            }

    }while(ch!=4);
    cout<<"\n\n";
    cout<<"Do you want to continue? " <<endl;
    cout <<"If yes enter 1 or else enter 0: " ;
    cin >>option;
    }while(option);

    return 0;
}
```

SAMPLE INPUT-OUTPUT



```
WELCOME
Enter string 1: str
Enter string 2: ings

string 1:str
string 2: ings

Which of the following operations would you like to perform?
1:Concatenation of strings
2:Comparing strings
3:Length of strings
4:Exit

Enter your choice: 1

After concatenation: strings

Which of the following operations would you like to perform?
1:Concatenation of strings
2:Comparing strings
3:Length of strings
4:Exit

Enter your choice: 2

Comparing both strings
String 1 smaller than string 2
```

```
Which of the following operations would you like to perform?
1:Concatenation of strings
2:Comparing strings
3:Length of strings
4:Exit

Enter your choice: 3

Length of string 1 is 3
Length of string 2 is 4

Which of the following operations would you like to perform?
1:Concatenation of strings
2:Comparing strings
3:Length of strings
4:Exit

Enter your choice: 4

Do you want to continue?
If yes enter 1 or else enter 0: 0
```


CONSTRUCTORS AND DESTRUCTORS

AIM

Program to demonstrate the order of execution of constructors & destructors for a matrix class

PROGRAM

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

class matrix
{
    int **p;
    int m,n;
    int i,j;
    int value;

public:

    matrix(){};
    matrix(int r,int c);
    void getmatrix();
    void putmatrix();
    void matrix_transpose();
    ~matrix()
    {
        cout <<"Destructor invoked" <<endl;
        for(int i=0;i<m;i++)
            delete p[i];
        delete p;
        cout<<"Memory Released " << endl;
    }
};

matrix::matrix(int r,int c)
{
    cout <<"Constructor invoked" <<endl;
    m = r;
    n = c;
    p = new int*[m];
```

```
        for(int i=0;i<m;i++)
            p[i] = new int[n];

    }

    void matrix::getmatrix()
    {
        cout<<"Enter elements rowwise:\n";
        for(i=0;i<m;i++)
        {
            for(j=0;j<n;j++)
            {
                cin>>p[i][j];
            }
        }
    }

}

    void matrix::putmatrix()
    {
        for(i=0;i<m;i++)
        {
            for(j=0;j<n;j++)
            {
                cout <<p[i][j]<<" ";
            }
            cout<<endl;
        }
    }

}

    void matrix::matrix_transpose()
    {
        for(i=0;i<n;i++)
        {
            for(j=0;j<m;j++)
            {
                cout<<p[j][i]<<" ";
            }
            cout<<endl;
        }
    }

}
```

```
int main()
{
    int m,n;
    cout<<"\tWELCOME\t " <<endl;

    cout<<"Enter number of rows and columns of the matrix: ";
    cin>>m>>n;
    cout<<"\n";
    cout<<"Construction in main"<<endl;
    cout<<"\n";
    matrix m1(m,n);
    m1.getmatrix();
    cout<<"\n";
    cout << "Matrix: " <<endl;
    m1.putmatrix();
    cout<<"\n";
    cout <<"Matrix Transpose " <<endl;
    m1.matrix_transpose();
    cout<<"\n";
    cout<<"Destruction in main" <<endl;

    return 0;
}
```

SAMPLE INPUT-OUTPUT

```
WELCOME
Enter number of rows and columns of the matrix: 2 3

Construction in main

Constructor invoked
Enter elements rowwise:
1 2 3
4 5 6

Matrix:
1 2 3
4 5 6

Matrix Transpose
1 4
2 5
3 6

Destruction in main
Destructor invoked
Memory Released
```

TIME CLASS

AIM

Create a class TIME with members hours, minutes, seconds. Take input, add two time objects by passing objects to function and display result.

PROGRAM

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

class Time
{
    private:
        int hours;
        int minutes;
        int seconds;
    public:
        void gettime()
        {
            cout <<"Hours : " ;
            cin >>hours;
            cout <<"Minutes : " ;
            cin >> minutes;
            cout<<"Seconds : " ;
            cin >> seconds;

        }
        void puttime(void)
        {
            cout<<hours<<" hours ";
            cout<<minutes<<" minutes and ";
            cout << seconds << " seconds" <<endl;
        }
        void sum(Time, Time);
};

void Time :: sum(Time t1, Time t2)
{

    seconds=t1.seconds+t2.seconds;
```

```
        minutes=seconds/60;
        seconds =seconds%60;
        minutes=minutes+t1.minutes+t2.minutes;
        hours=minutes/60;
        minutes=minutes%60;
        hours=hours+t1.hours+t2.hours;
    }

int main()
{
    Time T1,T2,T3;
    int option;
    cout<<"WELCOME " <<endl;
    do
    {
        cout <<"Time 1" <<endl;
        T1.gettime();
        cout<<"\n";
        cout<<"Time 2 " <<endl;
        T2.gettime();
        cout<<"\n";
        T3.sum(T1,T2);
        cout<<"\n";
        cout<< "Time 1 = ";T1.puttime();
        cout<< "Time 2 = ";T2.puttime();
        cout<< "Time 3 = ";T3.puttime();
        cout<<"\n\n";
        cout <<"Do you want to continue " <<endl;
        cout<<"If yes enter 1 or if no enter 0" <<endl;
        cout<<"\n";
        cout<<"Enter your choice: ";
        cin>>option;
    }while(option);

    return 0;
}
```

SAMPLE INPUT-OUTPUT

```
WELCOME
Time 1
Hours : 2
Minutes : 30
Seconds : 10

Time 2
Hours : 3
Minutes : 40
Seconds : 40

Time 1 = 2 hours 30 minutes and 10 seconds
Time 2 = 3 hours 40 minutes and 40 seconds
Time 3 = 6 hours 10 minutes and 50 seconds

Do you want to continue
If yes enter 1 or if no enter 0

Enter your choice: 1
Time 1
Hours : 4
Minutes : 50
Seconds : 30

Time 2
Hours : 1
Minutes : 20
Seconds : 20

Time 1 = 4 hours 50 minutes and 30 seconds
Time 2 = 1 hours 20 minutes and 20 seconds
Time 3 = 6 hours 10 minutes and 50 seconds

Do you want to continue
If yes enter 1 or if no enter 0

Enter your choice: 0
```

MATRIX CLASS AND OPERATIONS

AIM

Write a C++ program to implement a class MATRIX with member functions such as matrix_add, matrix_mult, matrix_transpose, matrix_trace etc

PROGRAM

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

class matrix
{
    int **p;
    int m,n;
    int i,j;
    int value;

public:

    matrix(){};
    matrix(int r,int c);
    void getmatrix();
    void putmatrix();
    friend void matrix_add(matrix , matrix );
    friend void matrix_mult(matrix ,matrix );
    void matrix_transpose();
    void matrix_trace();

};

matrix::matrix(int r,int c)
{
    m = r;
    n = c;
    p = new int*[m];

    for(int i=0;i<m;i++)
    {
        p[i] = new int[n];
    }
}

void matrix::getmatrix()
```



```
{

    cout<<"Enter elements rowwise:\n";
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            cin>>p[i][j];
        }
    }

}

void matrix::putmatrix()
{
    cout<<"\n";
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            cout <<p[i][j]<<" ";
        }
        cout<<endl;
    }

}

void matrix_add(matrix x,matrix y)
{
    int i,j;
    if(x.m==y.m && x.n==y.n)
    {
        cout<<"SUM OF MATRICES:\n";
        for(i=0;i<x.m;i++)
        {
            for(j=0;j<x.n;j++)
            {
                cout<<x.p[i][j]+y.p[i][j]<<" ";
                cout<<endl;
            }
        }
    }
}
```

```
        else
        {
            cout<<"\n";
            cout<<"Dimensions are not same...Addition not possible" <<endl;
            cout<<"\n";
        }
    }
void matrix_mult(matrix x,matrix y)
{
    int i,j,k,mult[100][100];
    if(x.n==y.m)
    {
        cout<<"\n";
        cout<<"MULTIPLE OF MATRICES:\n";
        for(i=0;i<x.m;i++)
        {
            for(j=0;j<y.n;j++)
            {
                for(k=0;k<x.n;k++)
                {
                    mult[i][j]+=(x.p[i][k])*(y.p[k][j]);
                }
                cout<<mult[i][j]<<" ";
            }
            cout<<endl;
        }
    }
    else
    {
        cout<<"\n";
        cout<<"Multiplication not possible"<<endl;
    }
}
void matrix::matrix_transpose()
{
    for(i=0;i<n;i++)
    {
        for(j=0;j<m;j++)
        {
            cout<<p[j][i]<<" ";
        }
    }
}
```

```
        cout<<endl;
    }
}
void matrix::matrix_trace()
{
    int trace=0;

    if (m==n)
    {
        for(i=0;i<m;i++)
        {
            for(j=0;j<n;j++)
            {
                if(i==j)
                {
                    trace=trace+p[i][j];
                }
            }
        }
        cout << trace <<endl;
    }
    else
    {
        cout<< "Not a square matrix hence tracing not possible" <<endl;
    }
}
int main()
{
    int m,n;
    int option;
    do
    {
        cout<<"\tWELCOME\t" << endl;
        cout<<"\n";
        cout<<"Matrix 1" << endl;
        cin>>m;
        cout<<"Enter number of columns : ";
        cin>>n;
        cout<<"\n";
        matrix m1(m,n);
        m1.getmatrix();
    }
}
```

```
cout<<"Matrix 2" << endl;
cout<<"Enter number of rows : ";
cin>>m;
cout<<"Enter number of columns : ";
cin>>n;
cout<<"\n";

matrix m2(m,n);
m2.getmatrix();
cout<<"\n";
cout << "Matrix 1: " <<endl;
m1.putmatrix();
cout<<"\n";
cout << "Matrix 2: " <<endl;
m2.putmatrix();
cout<<"\n";
int ch;
do
{

    cout<<"Select an option "<<endl;
    cout<<"1:Add two matrices " <<endl;
    cout<<"2:Multiply two matrices" <<endl;
    cout<<"3:Transpose of matrix" <<endl;
    cout<<"4:Trace of matrix" <<endl;
    cout<<"5:Quit"<<endl;
    cout<<"Enter your choice:";
    cin>>ch;
    cout<<"\n";
    switch(ch)
    {
        case 1:

            matrix_add(m1,m2);
            cout<<"\n";
            break;
        case 2:
            matrix_mult(m1,m2);
            cout<<"\n";
            break;
```

```
        case 3:
            cout<<"Transpose of Matrix 1 " <<endl;
            m1.matrix_transpose();
            cout<<"\n";
            cout<<"Transpose of Matrix 2 " <<endl;
            m2.matrix_transpose();
            cout<<"\n";
            break;
        case 4:
            cout<<"Trace of matrix 1 " <<endl;
            m1.matrix_trace();
            cout<<"\n";
            cout<<"Trace of matrix 2 " <<endl;
            m2.matrix_trace();
            cout<<"\n";
            break;
        case 5:break;
    }
    }while (ch!=5);
    cout<<"\n";
    cout<<"Do you want to continue? " <<endl;
    cout <<"If yes enter 1 or else enter 0" <<endl;
    cin >>option;
}while(option);
return 0;

}
```

SAMPLE INPUT-OUTPUT

```
WELCOME

Matrix 1
Enter number of rows : 2
Enter number of columns : 2

Enter elements rowwise:
1 2
3 4
Matrix 2
Enter number of rows : 2
Enter number of columns : 3

Enter elements rowwise:
1 2 3
4 5 6

Matrix 1:

1 2
3 4

Matrix 2:

1 2 3
4 5 6

Select an option
1:Add two matrices
2:Multiply two matrices
3:Transpose of matrix
4:Trace of matrix
5:Quit
Enter your choice:1

Dimensions are not same...Addition not possible
```

```
Select an option
1:Add two matrices
2:Multiply two matrices
3:Transpose of matrix
4:Trace of matrix
5:Quit
Enter your choice:2

MULTIPLE OF MATRICES:
9 12 15
19 26 33

Select an option
1:Add two matrices
2:Multiply two matrices
3:Transpose of matrix
4:Trace of matrix
5:Quit
Enter your choice:3

Transpose of Matrix 1
1 3
2 4

Transpose of Matrix 2
1 4
2 5
3 6

Select an option
1:Add two matrices
2:Multiply two matrices
3:Transpose of matrix
4:Trace of matrix
5:Quit
Enter your choice:4

Trace of matrix 1
5

Trace of matrix 2
Not a square matrix hence tracing not possible

Select an option
1:Add two matrices
2:Multiply two matrices
3:Transpose of matrix
4:Trace of matrix
5:Quit
Enter your choice:5

Do you want to continue?
If yes enter 1 or else enter 0
0
```

CONSTRUCTOR OVERLOADING FOR A COMPLEX CLASS

AIM

Write a program to perform addition of two complex numbers using constructor overloading. The first constructor which takes no argument is used to create objects which are not initialized, second which takes one argument is used to initialize real and image parts to equal values and third which takes two argument is used to initialize real and image to two different values.

PROGRAM

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

class complex
{
    float real,image;
public:
    complex(){}
    complex(float a)
    {
        real=image=a;
    }
    complex(float x,float y)
    {
        real=x;
        image=y;
    }
    friend complex sum(complex,complex);
    friend void display(complex);
};

complex sum(complex A,complex B)
{
    complex result;
    result.real=A.real+B.real;
    result.image=A.image+B.image;
    return result;
}
```

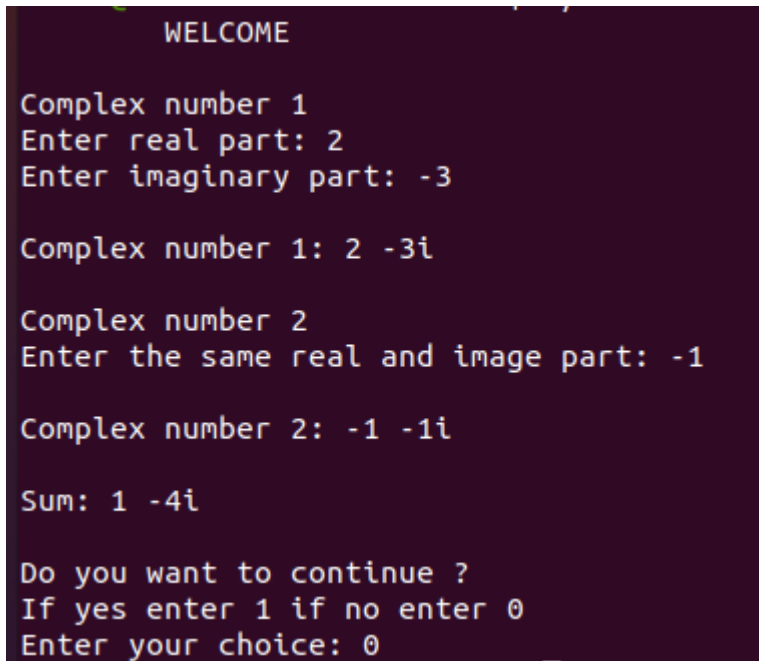


```
void display(complex number)
{
    if (number.image<0)
    {
        cout<<number.real<<" "<<number.image<<"i"<<endl;
    }
    else
    {
        cout<<number.real<<" + "<<number.image<<"i"<<endl;
    }
}

int main()
{
    int option;
    float num1,num2,num3;
    cout<<"\tWELCOME\t"<<endl;
    cout<<"\n";
    do
    {
        cout<<"Complex number 1"<<endl;
        cout<<"Enter real part: ";
        cin>>num1;
        cout<<"Enter imaginary part: ";
        cin>>num2;
        complex A(num1,num2);
        cout<<"\n";
        cout<<"Complex number 1: ";
        display(A);
        cout<<"\n";
        cout<<"Complex number 2"<<endl;
        cout<<"Enter the same real and image part: ";
        cin>>num3;
        complex B(num3);
        cout<<"\n";
        cout<<"Complex number 2: ";
        display(B);
        cout<<"\n";
        complex C;
        C=sum(A,B);
        cout<<"Sum: ";
        display(C);
    }
```

```
        cout<<"\n";
        cout<<"Do you want to continue ?"<<endl;
        cout<<"If yes enter 1 if no enter 0"<<endl;
        cout<<"Enter your choice: ";
        cin>>option;
    }while (option);
    return 0;
}
```

SAMPLE INPUT-OUTPUT



The screenshot shows the output of a C++ program in a terminal window. The text is as follows:

```
WELCOME

Complex number 1
Enter real part: 2
Enter imaginary part: -3

Complex number 1: 2 -3i

Complex number 2
Enter the same real and image part: -1

Complex number 2: -1 -1i

Sum: 1 -4i

Do you want to continue ?
If yes enter 1 if no enter 0
Enter your choice: 0
```

STATIC MEMBER FUNCTIONS

AIM

Write a C++ program to design a class having static member function named showcount() which has the property of displaying the number of objects created of the class.

PROGRAM

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

class test
{
    private:
        int code;
        static int count;

    public:

        test (void)
        {
            count=count+1;
        }
        static void showcount(void)
        {
            cout<<"Object created" <<endl;
            cout<<"count: "<<count<< endl;
            cout<<"\n";
        }
};

int test :: count;

int main()

{
    int i;
    int code=0;
    int num=1;
    cout<<"\tWELCOME\t"<< endl;
    cout<<"\n";
```

```
do
{
    test *a = new test;
    a->showcount();
    delete a;
    cout<<"Do you want to create more objects?" <<endl;
    cout<<"Yes 1 or no 0" <<endl;
    cout<<"Enter your choice: ";
    cin >> num;
    code=code+1;
}while(num==1);

cout<<"\n\n";
cout << "Number of objects created = " << code <<endl;
cout<<"\n\n";
cout <<"\tObjects created \t" <<endl;
cout <<"\n";
for(i=0;i<code;i++)
{
    cout << "Object number " << i+1 <<endl;
    cout<<"\n";
}

return 0;
}
```

SAMPLE INPUT-OUTPUT

```
WELCOME

Object created
count: 1

Do you want to create more objects?
Yes 1 or no 0
Enter your choice: 1
Object created
count: 2

Do you want to create more objects?
Yes 1 or no 0
Enter your choice: 1
Object created
count: 3

Do you want to create more objects?
Yes 1 or no 0
Enter your choice: 1
Object created
count: 4

Do you want to create more objects?
Yes 1 or no 0
Enter your choice: 1
Object created
count: 5

Do you want to create more objects?
Yes 1 or no 0
Enter your choice: 1
Object created
count: 6

Do you want to create more objects?
Yes 1 or no 0
Enter your choice: 1
Object created
count: 7

Do you want to create more objects?
Yes 1 or no 0
Enter your choice: 0

Number of objects created = 7

Objects created

Object number 1
Object number 2
Object number 3
Object number 4
Object number 5
Object number 6
Object number 7
```

DEPARTMENT STORE

AIM

Write a C++ program using class to process shopping list for a Departmental Store. The list include details such as the Code-no and price of each item and perform the operations like adding & deleting items to the list and printing the total value of an order.

PROGRAM

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

class list
{
    string itemName;
    int itemCode;
    float itemPrice;
    int itemQuantity;
    float totalSum;
public:
    list ()
    {
        itemCode = 0;
        itemPrice = 0;
        itemQuantity = 0;
        totalSum = 0;
    }
    void addItem ()
    {
        cout << "Item Name : ";
        cin >> itemName;
        cout << "Item Code : ";
        cin >> itemCode;
        cout << "Item Quantity : ";
        cin >> itemQuantity;
        cout << "Item Price : ";
        cin >> itemPrice;
        cout<<"\n";
    }
    void displayItem ()
    {
```

```
        cout << itemCode << "\t\t" << itemName << "\t\t" << itemPrice << "\t\t" <<
        itemQuantity << endl;
        cout<<"\n";
    }
    void alterQuantity (int decrement)
    {
        itemQuantity = itemQuantity - decrement;
    }
    int checkCode ()
    {
        return itemCode;
    }
    int checkQuantity ()
    {
        return itemQuantity;
    }
    void updateQuantity (int newQuantity)
    {
        itemQuantity = itemQuantity + newQuantity;
    }
    float generateSum ()
    {
        return itemQuantity * itemPrice;
    }
    void newQuantity (int newQuantity)
    {
        itemQuantity = newQuantity;
    }
};

int main ()
{
    list *stock = new list[10];
    list *shoppingList = new list[10];
    int stockCount = 0;
    int shoppingCount = 0;
    int option, choice;
    int keyPosition;
    bool found = false;
    cout<<"\tWELCOME\t"<<endl;
    cout<<"\n";
```

```
do
{
    cout <<
"1.Add Item\n2.Delete Item\n3.Update Item\n4.Display Items\n5.Purchase Item"
<< endl;
    cin >> option;
    cout<<"\n";
    switch (option)
{
case 1:
    stock[stockCount].addItem ();
    cout << "Item Added Successfully" << endl;
    cout<<"\n";
    stockCount++;
    break;
case 2:
    {
        int searchKey;
        cout << "Item Code : ";
        cin >> searchKey;
        for (int i = 0; i < stockCount; ++i)
        {
            if (searchKey == stock[i].checkCode ())
            {
                found = true;
                keyPosition = i;
                cout << "Successfully Deleted" << endl;
                break;
            }
        }
        if (!found)
        {
            cout << "Invalid Code" << endl;
        }
        else
        {
            for (int i = keyPosition; i < stockCount; ++i)
            {
                stock[i] = stock[i + 1];
            }
            stockCount--;
        }
    }
}
```



```
        }
    }
    break;
case 3:
{
    bool found = false;
    cout << "Item Code\tItem Name\tItem Price\tItem Quantity" << endl;
    for (int i = 0; i < stockCount; ++i)
    {
        stock[i].displayItem ();
    }
    int searchKey;
    cout << "Enter the Item Code : ";
    cin >> searchKey;
    for (int i = 0; i < stockCount; ++i)
    {
        if (searchKey == stock[i].checkCode ())
        {
            int x;
            cout << "Enter the Amount of Quantity : ";
            cin >> x;
            stock[i].updateQuantity (x);
            found = true;
            cout << "Successfully Updated" << endl;
            break;
        }
    }
    if (!found)
    {
        cout << "Invalid Code" << endl;
    }
}
break;
case 4:
    cout << "Item Code\tItem Name\tItem Price\tItem Quantity" << endl;
    for (int i = 0; i < stockCount; ++i)
    {
        stock[i].displayItem ();
    }
    break;
case 5:
```

```
        int choice;
do
{
    bool purchased = false;
    int tempItemCode, tempItemQuantity;
    cout << "Enter the Item Code" << endl;
    cin >> tempItemCode;
    cout << "Quantity you want to purchase" << endl;
    cin >> tempItemQuantity;
    for (int i = 0; i < stockCount; ++i)
    {
        if (tempItemCode == stock[i].checkCode ())
        {
            if (tempItemQuantity > stock[i].checkQuantity ())
            {
                cout << "Insufficient Stock" << endl;
                break;
            }
            purchased = true;
            cout << "Item Purchased" << endl;

            shoppingList[shoppingCount] = stock[i];
            shoppingList[shoppingCount].
            newQuantity (tempItemQuantity);
            stock[i].alterQuantity (tempItemQuantity);
            shoppingCount++;
            break;
        }
    }
}

    cout << "Do you want to continue purchase" << endl;
    cout << "1.Continue\n2.Generate Bill" << endl;
    cin >> choice;
    float totalSum;
    if (choice == 2 and purchased == true)
    {
        cout << "Your Purchase Bill" << endl;
        cout << "Item Code\tItem Name\tItem Price\tItem Quantity" <<
            endl;
        for (int i = 0; i < shoppingCount; ++i)
        {
            shoppingList[i].displayItem ();
        }
    }
}
```

```
        totalSum = totalSum + shoppingList[i].generateSum ();
    }
    cout << "Total Sum = " << totalSum << endl;
}
}while (choice == 1);
break;
default:
    break;
}

    cout << "Do you want to continue?"<< endl;
    cout<<"If yes enter 1 if no enter 0"<<endl;
    cin >> choice;
}while (choice);
return 0;
}
```

SAMPLE INPUT-OUTPUT

```
WELCOME

1.Add Item
2.Delete Item
3.Update Item
4.Display Items
5.Purchase Item
1

Item Name : Book
Item Code : 1
Item Quantity : 20
Item Price : 30

Item Added Successfully

Do you want to continue?
If yes enter 1 if no enter 0
1
1.Add Item
2.Delete Item
3.Update Item
4.Display Items
5.Purchase Item
1

Item Name : Pen
Item Code : 2
Item Quantity : 10
Item Price : 10

Item Added Successfully

Do you want to continue?
If yes enter 1 if no enter 0
1
1.Add Item
2.Delete Item
3.Update Item
4.Display Items
5.Purchase Item
1
```

```
Item Name : Soap
Item Code : 3
Item Quantity : 5
Item Price : 20

Item Added Successfully

Do you want to continue?
If yes enter 1 if no enter 0
1
1.Add Item
2.Delete Item
3.Update Item
4.Display Items
5.Purchase Item
2

Item Code : 3
Successfully Deleted
Do you want to continue?
If yes enter 1 if no enter 0
1
1.Add Item
2.Delete Item
3.Update Item
4.Display Items
5.Purchase Item
3

Item Code      Item Name      Item Price      Item Quantity
1              Book          30              20

2              Pen           10              10

Enter the Item Code : 1
Enter the Amount of Quantity : 25
Successfully Updated
Do you want to continue?
If yes enter 1 if no enter 0
1
1.Add Item
2.Delete Item
3.Update Item
4.Display Items
5.Purchase Item
4
```

```
Item Code      Item Name      Item Price      Item Quantity
1              Book          30              45

2              Pen           10              10

Do you want to continue?
If yes enter 1 if no enter 0
1
1.Add Item
2.Delete Item
3.Update Item
4.Display Items
5.Purchase Item
5

Enter the Item Code
1
Quantity you want to purchase
5
Item Purchased
Do you want to continue purchase
1.Continue
2.Generate Bill
2
Your Purchase Bill
Item Code      Item Name      Item Price      Item Quantity
1              Book          30              5

Total Sum = 150
Do you want to continue?
If yes enter 1 if no enter 0
0
```

SWAP PRIVATE DATA MEMBERS

AIM

Write a Program to swap private data members of classes named as class_1, class_2 using friend function

PROGRAM

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

class class_2;

class class_1
{
int value1;

public:
void indata()
{
    cout<<"Enter value 1: ";
    cin>>value1;
    cout<<"\n";

}
void display(void)
{
    cout <<"Value 1: " << value1 << "\n";
}
friend void exchange(class_1 &, class_2 &);
};

class class_2
{
    int value2;

    public:
    void indata()
    {
        cout<<"Enter value 2: ";
        cin>>value2;
```

```
    }
    void display(void)
    {
        cout <<"Value 2: " << value2 << "\n";
    }
    friend void exchange(class_1 &, class_2 &);
};

void exchange(class_1 &x, class_2 &y)
{
    int temp = x.value1;
    x.value1 = y.value2;
    y.value2 = temp;
}

int main()
{
    class_1 C1;
    class_2 C2;
    int option;

    cout<<"\tWELCOME\t" << endl;
    cout<<"\n";

    do
    {
        C1.indata();
        C2.indata();
        cout<<"\n";
        cout<<"Values Before Exchanging" << "\n";
        C1.display();
        C2.display();
        cout<<"\n";

        exchange(C1, C2);

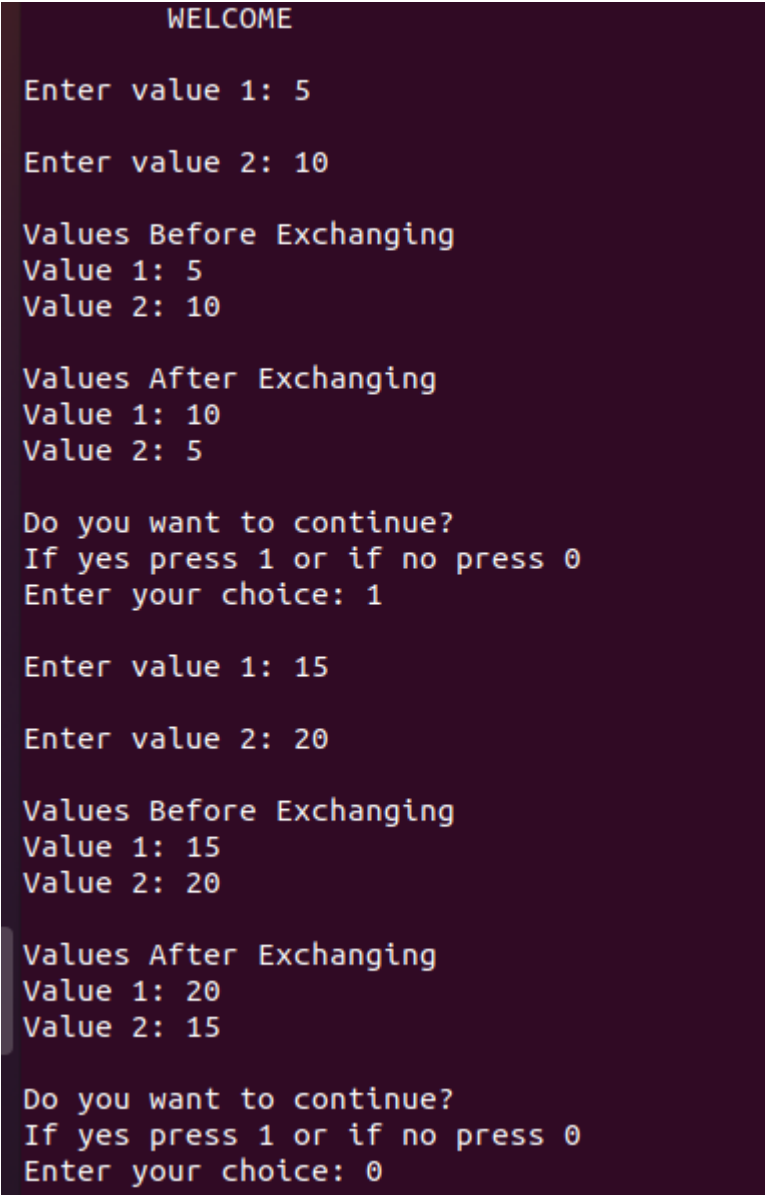
        cout<<"Values After Exchanging" << "\n";
        C1.display();
        C2.display();
        cout<<"\n";
        cout <<"Do you want to continue?" <<endl;
```



```
        cout <<"If yes press 1 or if no press 0"<<endl;
        cout<<"Enter your choice: ";
        cin>>option;
        cout<<"\n";
    }while(option);

    return 0;
}
```

SAMPLE INPUT-OUTPUT



```
WELCOME

Enter value 1: 5
Enter value 2: 10

Values Before Exchanging
Value 1: 5
Value 2: 10

Values After Exchanging
Value 1: 10
Value 2: 5

Do you want to continue?
If yes press 1 or if no press 0
Enter your choice: 1

Enter value 1: 15
Enter value 2: 20

Values Before Exchanging
Value 1: 15
Value 2: 20

Values After Exchanging
Value 1: 20
Value 2: 15

Do you want to continue?
If yes press 1 or if no press 0
Enter your choice: 0
```

COMPLEX CLASS

AIM

Program to design a class complex to represent complex numbers. The complex class should use an external function (use it as a friend function) to add two complex numbers. The function should return an object of type complex representing the sum of two complex numbers

PROGRAM

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

class complex
{
    float x;
    float y;

    public:

    void input();
    friend complex sum(complex, complex);
    void show(complex);
};

void complex::input()
{
    cout<<"Enter the real part: ";
    cin>>x;
    cout<<"Enter the imaginary part: ";
    cin>>y;
}

complex sum(complex c1, complex c2)
{
    complex c3;
    c3.x = c1.x + c2.x;
    c3.y = c1.y + c2.y;
    return(c3);
}

void complex :: show(complex c)
{
```

```
        if(c.y>=0)
        {
            cout<<c.x<<" + "<<c.y<<" i " <<"\n";
            cout<<"\n";
        }
        else
        {
            cout<<c.x << " " << c.y << "i" << endl;
            cout<<"\n";
        }
    }

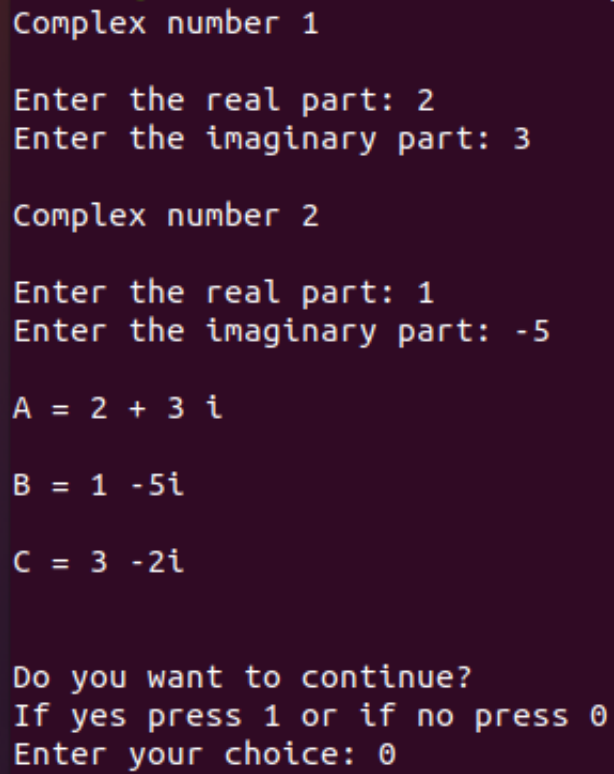
int main()
{
    complex A, B, C;
    int option;
    do
    {
        cout<<"Complex number 1 "<<endl;
        cout<<"\n";
        A.input();
        cout<<"\n";
        cout<<"Complex number 2 "<<endl;
        cout<<"\n";
        B.input();
        cout<<"\n";

        C = sum(A, B);

        cout<< "A = ";A.show(A);
        cout<< "B = ";B.show(B);
        cout<< "C = ";C.show(C);
        cout<<"\n";
        cout <<"Do you want to continue?" <<endl;
        cout <<"If yes press 1 or if no press 0"<<endl;
        cout<<"Enter your choice: ";
        cin>>option;
        cout<<"\n";
    }while(option);
```

```
    return 0;  
}
```

SAMPLE INPUT-OUTPUT



The screenshot shows a terminal window with a dark background and light-colored text. It displays the execution of a C++ program that handles complex numbers. The program prompts for two complex numbers, A and B, by asking for their real and imaginary parts. It then displays the resulting complex numbers A, B, and C. Finally, it asks if the user wants to continue, with instructions to press 1 for yes or 0 for no. The user has entered 0 for the choice.

```
Complex number 1  
  
Enter the real part: 2  
Enter the imaginary part: 3  
  
Complex number 2  
  
Enter the real part: 1  
Enter the imaginary part: -5  
  
A = 2 + 3 i  
  
B = 1 -5i  
  
C = 3 -2i  
  
Do you want to continue?  
If yes press 1 or if no press 0  
Enter your choice: 0
```

OPERATOR OVERLOADING FOR VECTOR CLASS

AIM

Write a C++ program to overload ==, !=, <, <=, > and >= operators as member operator functions for a vector object.

PROGRAM

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

class Vector
{
    float i,j,k,magnitude;
public:
    void get_data();
    void display_vector();
    void vector_magnitude();
    void operator==(Vector&);
    void operator!=( Vector&);
    void operator<( Vector&);
    void operator<=( Vector&);
    void operator>( Vector&);
    void operator>=( Vector&);
};

void Vector::get_data()
{
    cout<<"Enter the i,j,k components of the vector"<<endl;
    cin>>i>>j>>k;
    cout<<"\n";
}

void Vector::display_vector()
{
    cout<<i<<"i + "<<j<<"j + "<<k<<"k"<<endl;
    cout<<"\n";
}
```

```
}
```

```
void Vector::vector_magnitude()
{
    magnitude = (i*i)+(j*j)+(k*k);
    magnitude = sqrtf(magnitude);
    cout<<"Magnitude of the vector is "<<magnitude<<endl;
    cout<<"\n";
}
```

```
void Vector::operator==( Vector &V)
{
    if (magnitude == V.magnitude)
    {
        cout<<"The magnitude of vectors are equal"<<endl;
        cout<<"\n";
    }
    else
    {
        cout<<"The magnitude of vectors are not equal"<<endl;
        cout<<"\n";
    }
}
```

```
void Vector::operator!=(Vector &V)
{
    if (magnitude != V.magnitude)
    {
        cout<<"The magnitude of vectors are not equal"<<endl;
        cout<<"\n";
    }
    else
    {
        cout<<"The magnitude of vectors are equal"<<endl;
        cout<<"\n";
    }
}
```

```
void Vector::operator<(Vector &V)
```

```
{
    if (magnitude < V.magnitude)
    {
        cout<<"The magnitude of Vector 2 is greater than magnitude of vector 1\n";
        cout<<"\n";
    }
    else
    {
        cout<<"The magnitude of Vector 1 is greater than magnitude of vector 2\n";
        cout<<"\n";
    }
}

void Vector::operator<=(Vector &V)
{
    if (magnitude < V.magnitude)
    {
        cout<<"The magnitude of Vector 2 is greater than magnitude of vector 1"<<endl;
        cout<<"\n";
    }
    else if(magnitude == V.magnitude)
    {
        cout<<"The magnitude of vectors are equal"<<endl;
        cout<<"\n";
    }
    else
    {
        cout<<"Vector 1 greater than vector 2"<<endl;
        cout<<"\n";
    }
}

void Vector::operator>(Vector &V)
{
    if (magnitude > V.magnitude)
    {
        cout<<"The magnitude of Vector 1 is greater than magnitude of vector 2"<<endl;
        cout<<"\n";
    }
    else
    {

```

```
        cout<<"The magnitude of Vector 2 is greater than magnitude of vector 1"<<endl;
        cout<<"\n";
    }
}

void Vector::operator>=(Vector &V)
{
    if (magnitude > V.magnitude)
    {
        cout<<"The magnitude of Vector 1 is greater than magnitude of vector 2"<<endl;
        cout<<"\n";
    }
    else if(magnitude == V.magnitude)
    {
        cout<<"The magnitude of vectors are equal"<<endl;
        cout<<"\n";
    }
    else
    {
        cout<<"The magnitude of Vector 2 is greater than magnitude of vector 1"<<endl;
        cout<<"\n";
    }
}

int main()
{
    int option;
    cout<<"\tWELCOME\t"<<endl;
    cout<<"\n";

    Vector V1,V2;
    int choice,ch;
    cout<<"Vector V1"<<endl;
    cout<<"\n";
    V1.get_data();
    cout<<"\nVector V2"<<endl;
    cout<<"\n";
    V2.get_data();
    cout<<"\n";
    cout<<"Vector 1:"<<endl;
    V1.display_vector();
    V1.vector_magnitude();
}
```



```
cout<<"\n";
cout<<"Vector 2:"<<endl;
V2.display_vector();
V2.vector_magnitude();
cout<<"\n";

do
{
    cout<<"\nChoose any one of the following
operations
\n1.Vector1 == Vector2\n2.Vector1 != Vector2
\n3.Vector1 < Vector2\n4.Vector1 <= Vector2
\n5.Vector1 > Vector2\n6.Vector1 >= Vector2
\n7.Quit"<<endl;
    cout<<"\n";
    cout<<"Enter your choice: ";
    cin>>choice;
    switch (choice)
    {
        case 1:
            V1== V2;
            break;
        case 2:
            V1!= V2;
            break;
        case 3:
            V1< V2;
            break;
        case 4:
            V1<= V2;
            break;
        case 5:
            V1> V2;
            break;
        case 6:
            V1>= V2;
            break;
        case 7:
            break;
    }
    cout<<"\n";
```

```
        cout<<"Do you want to Continue ?"<<endl;
        cout <<"If yes enter 1 or if no enter 0"<<endl;
        cout<<"Enter your choice: ";
        cin>>ch;
    } while (ch!=0);

    return 0;
}
```

SAMPLE INPUT-OUTPUT

```
WELCOME

Vector V1
Enter the i,j,k components of the vector
1 2 3

Vector V2
Enter the i,j,k components of the vector
4 5 6

Vector 1:
1i + 2j + 3k
Magnitude of the vector is 3.74166

Vector 2:
4i + 5j + 6k
Magnitude of the vector is 8.77496

Choose any one of the following operations
1.Vector1 == Vector2
2.Vector1 != Vector2
3.Vector1 < Vector2
4.Vector1 <= Vector2
5.Vector1 > Vector2
6.Vector1 >= Vector2
7.Quit

Enter your choice: 1
The magnitude of vectors are not equal

Do you want to Continue ?
If yes enter 1 or if no enter 0
Enter your choice: 1
```

Choose any one of the following operations

- 1.Vector1 == Vector2
- 2.Vector1 != Vector2
- 3.Vector1 < Vector2
- 4.Vector1 <= Vector2
- 5.Vector1 > Vector2
- 6.Vector1 >= Vector2
- 7.Quit

Enter your choice: 2

The magnitude of vectors are not equal

Do you want to Continue ?

If yes enter 1 or if no enter 0

Enter your choice: 1

Choose any one of the following operations

- 1.Vector1 == Vector2
- 2.Vector1 != Vector2
- 3.Vector1 < Vector2
- 4.Vector1 <= Vector2
- 5.Vector1 > Vector2
- 6.Vector1 >= Vector2
- 7.Quit

Enter your choice: 3

The magnitude of Vector 2 is greater than magnitude of vector 1

Do you want to Continue ?

If yes enter 1 or if no enter 0

Enter your choice: 1

Choose any one of the following operations

- 1.Vector1 == Vector2
- 2.Vector1 != Vector2
- 3.Vector1 < Vector2
- 4.Vector1 <= Vector2
- 5.Vector1 > Vector2
- 6.Vector1 >= Vector2
- 7.Quit

Enter your choice: 4

The magnitude of Vector 2 is greater than magnitude of vector 1

```
Do you want to Continue ?
If yes enter 1 or if no enter 0
Enter your choice: 1

Choose any one of the following operations
1.Vector1 == Vector2
2.Vector1 != Vector2
3.Vector1 < Vector2
4.Vector1 <= Vector2
5.Vector1 > Vector2
6.Vector1 >= Vector2
7.Quit

Enter your choice: 5
The magnitude of Vector 2 is greater than magnitude of vector 1

Do you want to Continue ?
If yes enter 1 or if no enter 0
Enter your choice: 1

Choose any one of the following operations
1.Vector1 == Vector2
2.Vector1 != Vector2
3.Vector1 < Vector2
4.Vector1 <= Vector2
5.Vector1 > Vector2
6.Vector1 >= Vector2
7.Quit

Enter your choice: 6
The magnitude of Vector 2 is greater than magnitude of vector 1

Do you want to Continue ?
If yes enter 1 or if no enter 0
Enter your choice: 1

Choose any one of the following operations
1.Vector1 == Vector2
2.Vector1 != Vector2
3.Vector1 < Vector2
4.Vector1 <= Vector2
5.Vector1 > Vector2
6.Vector1 >= Vector2
7.Quit

Enter your choice: 7

Do you want to Continue ?
If yes enter 1 or if no enter 0
Enter your choice: 0
```

OVERLOAD OPERATORS FOR A COMPLEX CLASS

AIM

Write a C++ program to design a class representing complex numbers and having the functionality of performing addition & multiplication of two complex numbers using operator overloading (Use friend operator functions).

PROGRAM

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

class Complex
{
    float real;
    float image;
public:
    void get_number();
    void display_complex();
    friend Complex operator+(Complex,Complex);
    friend Complex operator*(Complex,Complex);
};

void Complex::get_number()
{
    cout<<"Enter the real part: ";
    cin>>real;
    cout<<"Enter the image part: ";
    cin>>image;
}

void Complex::display_complex()
{
    if (image<0)
    {
        cout<<real<<" "<<image<<"i"<<endl;
    }
    else
    {
        cout<<real<<" + "<<image<<"i"<<endl;
    }
}
```

```
}
```

```
Complex operator+(Complex A,Complex B)
```

```
{
    Complex temp;
    temp.real = A.real+B.real;
    temp.image = A.image+B.image;
    return temp;
}
```

```
Complex operator*(Complex A,Complex B)
```

```
{
    Complex temp;
    //real=ac-bd
    //image=bc+ad
    temp.real = (A.real*B.real)-(A.image*B.image);
    temp.image = (A.image*B.real) + (A.real*B.image);
    return temp;
}
```

```
int main()
```

```
{
    int choice,ch;
    cout<<"\tWELCOME\t"<<endl;
    cout<<"\n";

    Complex C1,C2,C3,C4;
    cout<<"Complex Number 1: "<<endl;
    C1.get_number();
    cout<<"\n";
    cout<<"Complex Number 2: "<<endl;
    C2.get_number();
    cout<<"\n";
    cout<<"Complex Number 1: ";
    C1.display_complex();
    cout<<"\n";
    cout<<"Complex Number 2: ";
    C2.display_complex();
    cout<<"\n";
    do
    {
        cout<<"Select an option "<<endl;
```

```
    cout<<"1:Add two complex numbers" <<endl;
    cout<<"2:Multiply two complex numbers"<<endl;
    cout<<"3:Exit" <<endl;
    cout<<"\n";
    cout<<"Enter your choice: ";
    cin>>choice;
    cout<<"\n";
    switch (choice)
    {
        case 1:
            C3=C1+C2;
            cout<<"Sum of the two complex numbers is ";
            C3.display_complex();
            cout<<"\n";
            break;
        case 2:
            C4=C1*C2;
            cout<<"Product of the two complex numbers is ";
            C4.display_complex();
            cout<<"\n";
            break;
        case 3:
            break;
    }
    cout<<"Do you want to continue ? "<<endl;
    cout<<"If yes enter 1 ,if no enter 0"<<endl;
    cout<<"Enter your choice: ";
    cin>>ch;
    cout<<"\n";
}while (ch);
return 0;
}
```

SAMPLE INPUT-OUTPUT

```
WELCOME

Complex Number 1:
Enter the real part: 2
Enter the image part: -3

Complex Number 2:
Enter the real part: 6
Enter the image part: -5

Complex Number 1: 2 -3i
Complex Number 2: 6 -5i

Select an option
1:Add two complex numbers
2:Multiply two complex numbers
3:Exit

Enter your choice: 1

Sum of the two complex numbers is 8 -8i

Do you want to continue ?
If yes enter 1 ,if no enter 0
Enter your choice: 1

Select an option
1:Add two complex numbers
2:Multiply two complex numbers
3:Exit

Enter your choice: 2

Product of the two complex numbers is -3 -28i

Do you want to continue ?
If yes enter 1 ,if no enter 0
Enter your choice: 0
```


OPERATOR OVERLOADING FOR VECTOR CLASS

AIM

Write a C++ program to overload operators like *, <<, >> using friend function. The following overloaded operators should work for a class vector

PROGRAM

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

class Vector
{
    float i,j,k;
    float vector_product;
public:
    void operator*(Vector&);
    friend istream & operator>>(istream&,Vector&);
    friend ostream & operator<<(ostream&,Vector&);
};

istream & operator>>(istream&din,Vector&a)
{
    cout<<"Enter i,j,k components of vector"<<endl;
    cin>>a.i>>a.j>>a.k;
    cout<<"\n";
    return (din);
}

ostream & operator<<(ostream&dout,Vector&a)
{
    dout<<a.i<<"i + "<<a.j<<"j + "<<a.k<<"k"<<endl;
    cout<<"\n";
    return dout;
}

void Vector::operator*( Vector&a)
{
    float vector_product;
    vector_product = (i*a.i)+(j*a.j)+(k*a.k);
```

```
        cout<<"Dot product of two vectors:  "<<vector_product<<endl;
        cout<<"\n";
    }
int main()
{
    int choice,ch;
    cout<<"\tWELCOME\t"<<endl;
    cout<<"\n";
    do
    {
        Vector V1,V2,V3;
        cout<<"Vector 1"<<endl;
        cin>>V1;
        cout<<"Vector 2"<<endl;
        cin>>V2;
        cout<<"Vector 1: ";
        cout<<V1 <<endl;
        cout<<"Vector 2: ";
        cout<<V2 <<endl;
        cout<<"\n";
        cout<<"Enter your choice "<<endl;
        cout<<"1:Dot product of vectors" <<endl;
        cout<<"2:Exit" <<endl;
        cout<<"Enter your choice: ";
        cin>>choice;

        switch (choice)
        {
            case 1:
                V1*V2;

            case 2:
                break;
        }
        cout<<"Do you want to continue ? "<<endl;
        cout<<"If yes enter 1 ,if no enter 0"<<endl;
        cout<<"Enter your choice: ";
        cin>>ch;
    } while (ch);
}
```

```
        return 0;  
    }
```

SAMPLE INPUT-OUTPUT

```
WELCOME  
  
Vector 1  
Enter i,j,k components of vector  
1 2 3  
  
Vector 2  
Enter i,j,k components of vector  
4 5 6  
  
Vector 1: 1i + 2j + 3k  
  
Vector 2: 4i + 5j + 6k  
  
Enter your choice  
1:Dot product of vectors  
2:Exit  
Enter your choice: 1  
Dot product of two vectors: 32  
  
Do you want to continue ?  
If yes enter 1 ,if no enter 0  
Enter your choice: 0
```

OPERATOR OVERLOADING FOR MATRIX CLASS

AIM

Write a C++ program for developing a matrix class which can handle integer matrices of different dimensions. Also overload the operator for addition and multiplication of matrices. Use double pointers in your program to dynamically allocate memory for the matrices.

PROGRAM

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

class matrix
{
    int R,C ;
    int **p ;
public:

    matrix(){}
    matrix(int rows,int columns);
    void get_matrix();
    void display_matrix();
    friend matrix operator+(matrix A, matrix B);
    friend matrix operator*(matrix A, matrix B);
};

matrix::matrix(int rows,int columns)
{
    R = rows ; C = columns ;
    p = new int* [R+1];
    for(int i = 0; i < C ; i++)
    {
        p[i] = new int[C+1];
    }
}

void matrix:: get_matrix()
{
    cout<<"Enter  elements of the matrix \n";
    for(int i = 0 ; i < R ; i++){
        for(int j = 0; j < C; j++)
```

```
        {
            int value ;
            cin >> value ;
            p[i][j] = value;
        }
    }
}

void matrix::display_matrix()
{
    for(int i = 0 ; i < R ; i++)
    {
        for(int j = 0; j < C; j++)
        {
            cout << p[i][j] << " " ;
        }
        cout << "\n" ;
    }
}

matrix operator+ (matrix A , matrix B)
{
    if (A.R == B.R && A.C == B.C)
    {
        matrix M(A.R,A.C);
        for(int i = 0 ; i < M.R ; i++)
        {
            for(int j = 0; j < M.C; j++){
                M.p[i][j] = A.p[i][j] + B.p[i][j];
            }
        }
        cout<<"Sum: " <<endl;
        return M ;
    }
    else
    {
        matrix a(0,0);
        cout<<"Addition not possible " <<endl;
        return a;
    }
}
```

```
}

matrix operator* (matrix A, matrix B)
{
    if (A.C == B.R)
    {
        matrix X(A.R,B.C);
        for(int i = 0 ;i < A.R ;i++)
        {
            for(int j = 0 ; j < B.C ; j++)
            {
                int sum = 0;
                for(int n = 0; n < A.C ;n++)
                {
                    sum+= A.p[i][n] * B.p[n][j];
                }
                X.p[i][j] = sum;
            }
        }
        cout<<"Product: " <<endl;
        return X ;
    }
    else
    {
        matrix l(0,0);
        cout<<"Multiplication not possible " <<endl;
        return l;
    }
}

int main()
{
    int choice,ch;
    int r1,r2,c1,c2;
    cout<<"\tWELCOME\t"<<endl;
    cout<<"\n";
    cout<<"Matrix 1 " << endl;
```

```
cout << "Enter number of rows :";
cin >> r1 ;
cout<<"Enter number of columns : ";
cin>> c1;
cout<<"\n";
matrix A(r1,c1);
A.get_matrix();
cout<<"\n";
cout<<"Matrix 2 " << endl;
cout << "Enter number of rows :";
cin >> r2 ;
cout<<"Enter number of columns : ";
cin>> c2;
cout<<"\n";
matrix B(r2,c2);
B.get_matrix();
cout<<"\n";
cout << "Matrix 1:" <<endl;
A.display_matrix();
cout << "\n";
cout << "Matrix 2: " <<endl;
B.display_matrix();
cout << "\n";
do
{
    cout<<"Select an option"<<endl;
    cout<<"1:Add two matrices" <<endl;
    cout<<"2:Multiply two matrices"<<endl;
    cout<<"3:Exit" <<endl;
    cout<<"\n";
    cout<<"Enter your choice: ";
    cin>>choice;
    cout<<"\n";
    if(choice==1)
    {
        matrix C = A + B;
        C.display_matrix();
        cout<<"\n";
    }
    else if(choice==2)
    {
```

```
        matrix D = A * B;
        D.display_matrix();
        cout<<"\n";
    }
    else
    {
        break;
    }
    cout<<"Do you want to continue ? "<<endl;
    cout<<"If yes enter 1 ,if no enter 0"<<endl;
    cout<<"Enter your choice: ";
    cin>>ch;
    cout<<"\n";
} while (ch);

return 0;

}
```


SAMPLE INPUT-OUTPUT

```
WELCOME

Matrix 1
Enter number of rows :2
Enter number of columns : 2

Enter elements of the matrix
1 2
3 4

Matrix 2
Enter number of rows :2
Enter number of columns : 3

Enter elements of the matrix
1 2 3
4 5 6

Matrix 1:
1 2
3 4

Matrix 2:
1 2 3
4 5 6

Select an option
1:Add two matrices
2:Multiply two matrices
3:Exit

Enter your choice: 1

Addition not possible

Do you want to continue ?
If yes enter 1 ,if no enter 0
Enter your choice: 1

Select an option
1:Add two matrices
2:Multiply two matrices
3:Exit

Enter your choice: 2

Product:
9 12 15
19 26 33

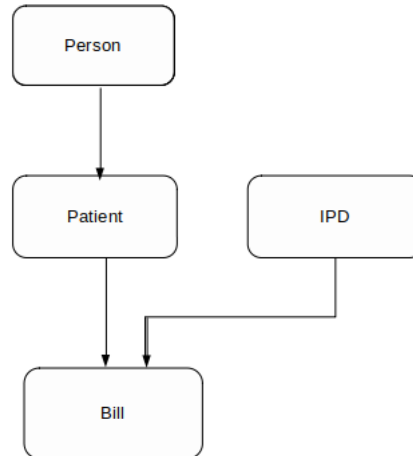
Do you want to continue ?
If yes enter 1 ,if no enter 0
Enter your choice: 0
```

MULTIPLE AND MULTILEVEL INHERITANCE

AIM

Write a C++ program to demonstrate the concept of Multiple and Multilevel inheritance including constructors with parameters

INHERITANCE DIAGRAM



PROGRAM

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

class Person
{
    protected:
        string name;
        int age;
        string gender;
    public:
        Person(string n,int a,string g)
        {
            name = n;
            age = a;
            gender = g;
        }
        void show()
        {
```

```
        cout << "Name : " << name << endl;
        cout << "Age : " << age << endl;
        cout<<"Gender: " << gender <<endl;
    }
};

class Patient:Person
{
    protected:
        string disease;
        string doctor;

    public:
        Patient(string na,int ag,string ge,string d,string doc):Person(na,ag,ge)
        {
            disease = d;
            doctor = doc;

        }

        void show()
        {
            Person::show();
            cout<<"Disease: " << disease << endl;
            cout<<"Doctor: " <<doctor<<endl;

        }

};

class IPD
//Inpatient Department details
{
    protected:
        int ward_no;
        int bed_no;
        int charge_per_day;

    public:
        IPD(int w,int b,int ch)
        {
            ward_no = w;
            bed_no = b;
            charge_per_day = ch;
        }
};
```

```
    }

    void show()
    {
        cout<<"Ward No.:  "<<ward_no<<endl;
        cout<<"Bed No.:  "<<bed_no<<endl;
        cout<<"Charge Per Day:  "<<charge_per_day<<endl;
    }
};

class Bill:public Patient,public IPD
{
    protected:
    int no_of_days_admitted;

    public:
    Bill(string na,int ag,string ge,string d,string doc,
    int w,int b,int ch,int no):IPD(w,b,ch),Patient(na,ag,ge,d,doc)
    {
        no_of_days_admitted = no;
    }
    void show()
    {
        float total=charge_per_day*no_of_days_admitted;
        Patient::show();
        IPD::show();
        cout<<"No of days admitted: "<<no_of_days_admitted<<endl;
        cout<<"Amount:"<<total<<endl;
    }
};

int main()
{
    string Nm,gen,dis,doct;
    int ag,wa,be,cha,day;
    cout << "Welcome\n\n";
    cout << "Enter the Details : \n\n";
    cout << "Name   : " ;
    cin >> Nm;
    cout << "Age   : ";
    cin >> ag;
    cout << "Gender : " ;
```

```
    cin >> gen ;
    cout << "Disease : ";
    cin >> dis ;
    cout<<"Name of Doctor : ";
    cin>>doct;
    cout << "Ward number and bed number :";
    cin >>wa  >>be ;
    cout << "Charge per day : ";
    cin >> cha;
    cout << "No of days admitted :";
    cin >> day;
    cout << "\n\n";
    Bill B1(Nm,ag,gen,dis,doct,wa,be,cha,day);
    cout<<"PATIENT DETAILS"<<endl;
    cout<<"\n";
    B1.show();
    return 0;
}
```

SAMPLE INPUT-OUTPUT

```
Welcome

Enter the Details :

Name  : Suresh
Age   : 20
Gender : Male
Disease : Covid
Name of Doctor : Rajesh
Ward number and bed number :1 3
Charge per day : 100
No of days admitted :10

PATIENT DETAILS

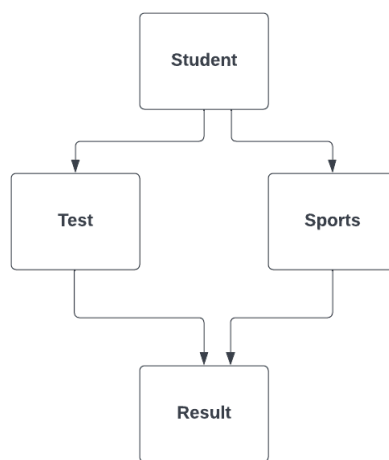
Name : Suresh
Age  : 20
Gender: Male
Disease: Covid
Doctor: Rajesh
Ward No.: 1
Bed No.: 3
Charge Per Day: 100
No of days admitted: 10
Amount:1000
```

VIRTUAL BASE CLASS

AIM

Write a C++ program to design a student class representing student roll no. and a test class (derived class of student) representing the scores of the student in various subjects and sports class representing the score in sports. The sports and test class should be inherited by a result class having the functionality to add the scores and display the final result for a student. Demonstrate the concept of Virtual base class on Hybrid inheritance.

INHERITANCE DIAGRAM



PROGRAM

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

class student
{
    string Name;
    int rollno ;
public:
    void get_studentdata()
    {
        cout<<"Enter name of student: ";
        cin >>Name;
        cout<<"Enter roll number: ";
        cin >>rollno;
    }
}
```

```
        void display_data(void)
        {
            cout<<"Name: " <<Name <<endl;
            cout<<"Roll No: " << rollno <<endl;

        }
};

class test : virtual public student
{
    protected:
        float Sub1,Sub2,Sub3 ;
    public:
        void get_marks()
        {
            cout<<"Enter marks obtained in subject 1: ";
            cin>>Sub1;
            cout<<"Enter marks obtained in subject 2: ";
            cin>>Sub2;
            cout<<"Enter marks obtained in subject 3: ";
            cin>>Sub3;
        }
        void put_marks(void)
        {
            cout<<"Marks obtained " <<endl;
            cout<<"Subject 1: " << Sub1 <<endl;
            cout<<"Subject 2: " << Sub2 <<endl;
            cout<<"Subject 3: " << Sub3 <<endl;
        }
};

class sports : virtual public student
{
    protected:
        float score ;
    public:
        void get_score()
        {
            cout <<"Enter score obtained: ";
            cin >>score;
            cout<<"\n";
        }
};
```



```
    }  
    void put_score(void)  
    {  
        cout<<"Score: " << score <<endl;  
    }  
};
```

```
class Result : public test , public sports  
{  
    float total;  
public:  
    void total_marks()  
    {  
        total = Sub1 + Sub2 + score;  
    }  
    void displayResult()  
    {  
        cout<<"\\n";  
        display_data();  
        put_marks();  
        put_score();  
        total_marks();  
        cout<<"Total:  "<<total<<endl;  
    }  
};
```

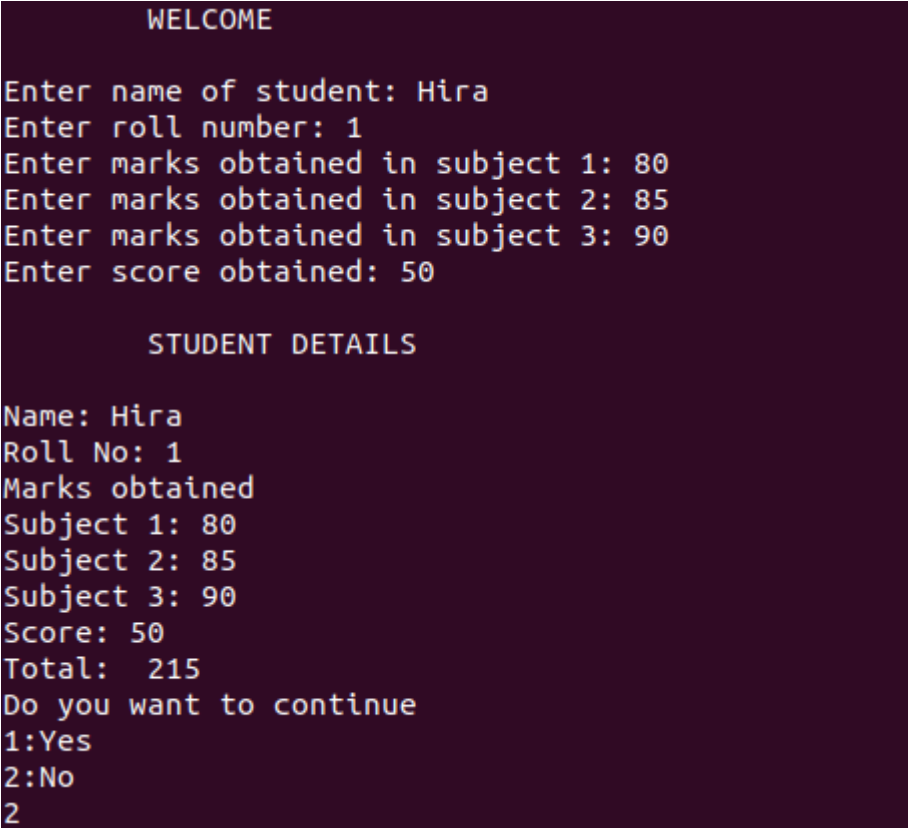
```
int main()  
{  
    int ch;  
  
    cout<<"\\tWELCOME\\t" <<endl;  
    cout<<"\\n";  
    do  
    {  
        Result R;  
        R.get_studentdata();  
        R.get_marks();  
        R.get_score();  
        cout<<"\\tSTUDENT DETAILS\\t"<<endl;
```

```
        R.displayResult();
        cout<<"Do you want to continue" <<endl;
        cout<<"1:Yes" <<endl;
        cout<<"2:No" <<endl;
        cin>>ch;

    }while(ch!=2);

    return 0;
}
```

SAMPLE INPUT-OUTPUT



```
WELCOME

Enter name of student: Hira
Enter roll number: 1
Enter marks obtained in subject 1: 80
Enter marks obtained in subject 2: 85
Enter marks obtained in subject 3: 90
Enter score obtained: 50

STUDENT DETAILS

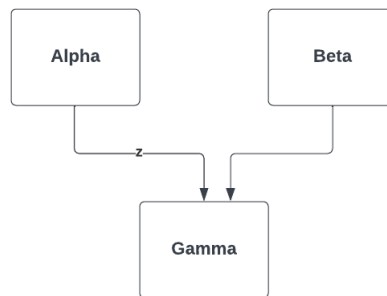
Name: Hira
Roll No: 1
Marks obtained
Subject 1: 80
Subject 2: 85
Subject 3: 90
Score: 50
Total: 215
Do you want to continue
1:Yes
2:No
2
```

CONSTRUCTORS DURING INHERITANCE

AIM

Write a C++ program illustrating how the constructors are implemented and the order in which they are called when the classes are inherited. Use three classes named alpha, beta and gamma such that alpha and beta are base classes and gamma is a derived class inheriting alpha & beta

INHERITANCE DIAGRAM



PROGRAM

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

class alpha
{
    protected:
        int a ;
    public:
        alpha(int i)
        {
            cout << "Alpha initialized\n\n";
            a = i ;
        }
        void showAlpha()
        {
            cout << "Alpha = " << a << endl;
        }
};

class beta
```

```
{
    int b;
public:
    beta(int j)
    {
        cout << "Beta initialized \n\n";
        b = j;
    }
    void showBeta()
    {
        cout << "Beta = " << b << endl;
    }
};

class gamma : public beta,public alpha
{
    int g;
public:
    gamma(int i,int j,int k):alpha(i),beta(j)
    {
        cout << "Gamma initialized \n\n";
        g = k;
    }
    void show_abg()
    {
        cout << "Gamma = " << g << endl;
        showAlpha();
        showBeta();
    }
};

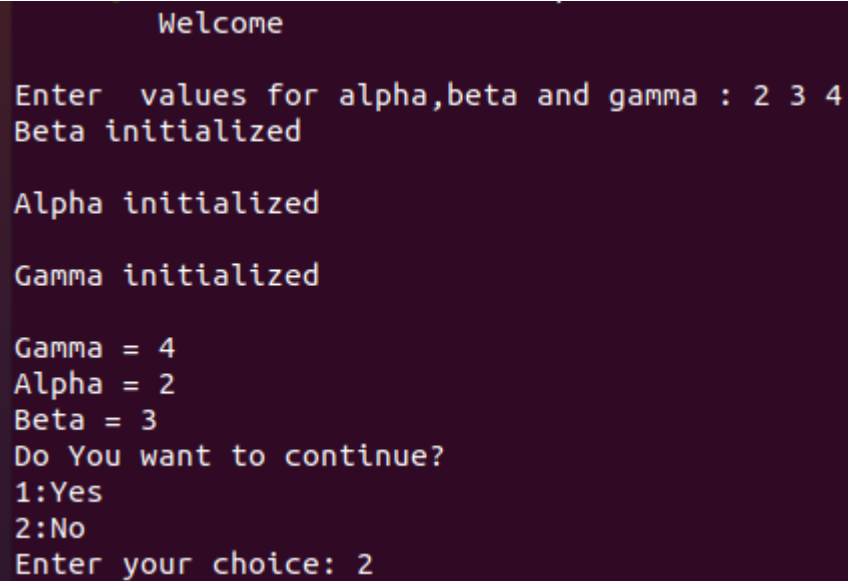
int main()
{
    int choice;
    int a,b,c;
    cout<<"\tWelcome\t" <<endl;
    cout<<"\n";

    do
    {
```

```
        cout << "Enter values for alpha,beta and gamma : ";
        cin >> a >> b >> c ;
        gamma G(a,b,c);
        G.show_abg();
        cout<<"Do You want to continue? "<<endl;
        cout<<"1:Yes"<<endl;
        cout<<"2:No" <<endl;
        cout<<"Enter your choice: ";
        cin>>choice;
        cout<<"\n";
    } while (choice != 2);

    return 0;
}
```

SAMPLE INPUT-OUTPUT



```
Welcome

Enter values for alpha,beta and gamma : 2 3 4
Beta initialized

Alpha initialized

Gamma initialized

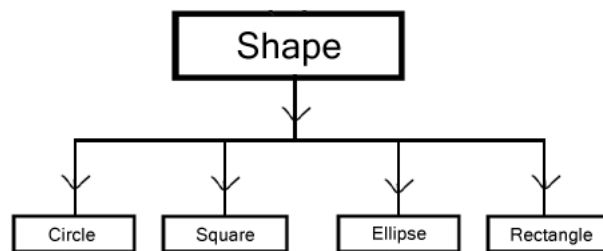
Gamma = 4
Alpha = 2
Beta = 3
Do You want to continue?
1:Yes
2:No
Enter your choice: 2
```

RUN TIME POLYMORPHISM

AIM

Write a C++ program to define classes Shape, Circle, Square, Ellipse and Rectangle with member functions to get the values for finding corresponding areas and print the same. Utilize the concept of Abstract Class and Runtime polymorphism to solve the problem.

INHERITANCE DIAGRAM



PROGRAM

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

class Shape
{
    protected:
        float area;
    public:
        virtual void get_area()=0;
        virtual void show_area()=0;
};

class Circle : public Shape
{
    float r;
    public:
        void get_area()
        {
            cout<<"Enter the radius of circle:";
            cin >>r;
            area = 3.14*r*r;
        }
}
```

```
        void show_area()
        {
            cout<<"Area of the circle = "<<area<<endl;
        }
};

class Square : public Shape
{
    float a;
public:
    void get_area()
    {
        cout << "Enter the length of side of square:";
        cin >> a ;
        area = a*a;
        cout<<"\n";
    }
    void show_area()
    {
        cout<<"Area of square = " << area <<endl;
    }
};

class Rectangle : public Shape
{
    float l,b;
public:
    void get_area()
    {
        cout << "Enter length of the rectangle: ";
        cin >> l ;
        cout << "Enter breadth of the rectangle: ";
        cin >> b ;
        cout <<"\n";
        area = l*b;
    }
    void show_area()
    {
        cout<<"Area of the rectangle  = "<<area<< endl;
        cout<<"\n";
    }
}
```

```
};

class Ellipse : public Shape
{
    float major_axis, minor_axis;
public:
    void get_area()
    {
        cout << "Enter length of major axis of ellipse: ";
        cin >> major_axis ;
        cout << "Enter length of minor axis of ellipse: ";
        cin >> minor_axis ;
        cout << "\n";
        area = 3.14* major_axis*minor_axis;
    }
    void show_area()
    {
        cout << "Area of ellipse = " << area << endl;
        cout << "\n";
    }
};

int main()
{
    int option, choice, ch;
    cout << "\tWELCOME\t" << endl;

    Shape *shapeptr;

    do
    {
        cout << "Select an option : \n 1:Area of Circle\n 2:Area of Square\n 3:Area of Ellipse\n 4:Area of Rectangle\n 5:Quit\n" << endl;
        cin >> choice;
        switch (choice)
        {
            case 1 :
            {
                shapeptr = new Circle;
                shapeptr -> get_area();
                shapeptr -> show_area();
            }
        }
    }
}
```



```
        break;
    }
    case 2 :
    {
        shapeptr = new Square;
        shapeptr ->get_area();
        shapeptr ->show_area();
        break;
    }
    case 3 :
    {
        shapeptr = new Ellipse;
        shapeptr ->get_area();
        shapeptr ->show_area();
        break;
    }
    case 4 :
    {
        shapeptr = new Rectangle;
        shapeptr ->get_area();
        shapeptr ->show_area();
        break;
    }
    case 5 :
    {
        break;
    }
    default :
        cout << "Try Again" <<endl;
        break;
}
cout<<"Do you want to continue?" <<endl;
cout<<"If yes enter 1 if no enter 0" <<endl;
cin>>ch;
}while(ch);
if(ch==0)
{
    cout<<"THANK YOU" <<endl;
    return 0;
}
}
```

SAMPLE INPUT-OUTPUT

```
WELCOME
Select an option :
 1:Area of Circle
 2:Area of Square
 3:Area of Ellipse
 4:Area of Rectangle
 5:Quit

1
Enter the radius of circle:10
Area of the circle = 314
Do you want to continue?
If yes enter 1 if no enter 0
1
Select an option :
 1:Area of Circle
 2:Area of Square
 3:Area of Ellipse
 4:Area of Rectangle
 5:Quit

2
Enter the length of side of square:4

Area of square = 16
Do you want to continue?
If yes enter 1 if no enter 0
1
Select an option :
 1:Area of Circle
 2:Area of Square
 3:Area of Ellipse
 4:Area of Rectangle
 5:Quit

3
Enter length of major axis of ellipse: 10
Enter length of minor axis of ellipse: 5

Area of ellipse = 157
```

```
Do you want to continue?
If yes enter 1 if no enter 0
1
Select an option :
1:Area of Circle
2:Area of Square
3:Area of Ellipse
4:Area of Rectangle
5:Quit

4
Enter length of the rectangle: 10
Enter breadth of the rectangle: 5

Area of the rectangle = 50

Do you want to continue?
If yes enter 1 if no enter 0
1
Select an option :
1:Area of Circle
2:Area of Square
3:Area of Ellipse
4:Area of Rectangle
5:Quit

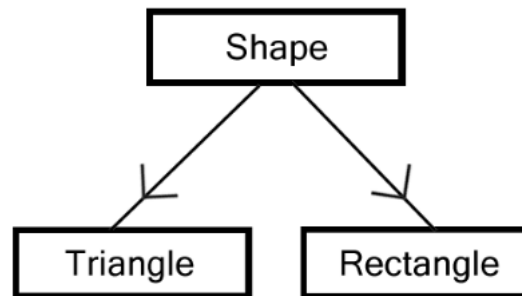
5
Do you want to continue?
If yes enter 1 if no enter 0
0
THANK YOU
```

PURE VIRTUAL FUNCTION

AIM

Write a C++ program to demonstrate the use of pure virtual functions and abstract base classes

INHERITANCE DIAGRAM



PROGRAM

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

class Shape
{
protected:
    float dimension;

public:
    void getDimension()
    {
        cin >> dimension;
    }

    virtual void calculateArea() = 0;
};

class Square : public Shape
{
protected:
    float areaSquare;
```

```
public:
    void calculateArea()
    {
        areaSquare=dimension * dimension;
        cout<<"Area of square = " << areaSquare <<endl;
    }
};

class Circle : public Shape
{
protected:
    float areaCircle;
public:
    void calculateArea()
    {
        areaCircle = 3.14 * dimension * dimension;
        cout<<"Area of circle = " << areaCircle <<endl;
    }
};

int main()

{
    int option,choice,ch;
    cout<<"\tWELCOME\t" <<endl;

    Shape *shapeptr;

    do
    {
        cout << "Select an option : \n 1:Area of Circle\n 2:Area of Square\n 3:Quit\n" <<endl;
        cout<<"Enter your choice: ";
        cin >> choice;
        switch (choice)
        {
            case 1 :
            {
                shapeptr = new Circle;
                cout << "\nEnter radius of the circle: ";
```

```
        shapeptr ->getDimension();
        shapeptr ->calculateArea();
        break;
    }
    case 2 :
    {
        shapeptr = new Square;
        cout << "Enter the length of the square: ";
        shapeptr ->getDimension();
        shapeptr ->calculateArea();
        break;
    }
    case 3 :
    {
        break;
    }
    default :
        cout << "Try Again" <<endl;
        break;
}
cout<<"Do you want to continue?" <<endl;
cout<<"If yes enter 1 if no enter 0" <<endl;
cin>>ch;
}while(ch);

if(ch==0)
{
    cout<<"THANK YOU" <<endl;
    return 0;

}

}
```

SAMPLE INPUT-OUTPUT

```
WELCOME
Select an option :
1:Area of Circle
2:Area of Square
3:Quit

Enter your choice: 1

Enter radius of the circle: 10
Area of circle = 314
Do you want to continue?
If yes enter 1 if no enter 0
1
Select an option :
1:Area of Circle
2:Area of Square
3:Quit

Enter your choice: 2
Enter the length of the square: 4
Area of square = 16
Do you want to continue?
If yes enter 1 if no enter 0
1
Select an option :
1:Area of Circle
2:Area of Square
3:Quit

Enter your choice: 3
Do you want to continue?
If yes enter 1 if no enter 0
0
THANK YOU
```

TEMPLATE CLASS

AIM

Write a C++ program to demonstrate the use of class templates.

PROGRAM

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

template <class T>
class Calculator
{
    T num1,num2;
public:
    Calculator(T n1,T n2)
    {
        num1=n1;num2=n2;
    }
    T add()
    {
        return(num1+num2);
    }
    T subtract()
    {
        return(num1-num2);
    }
    T multiply()
    {
        return(num1*num2);
    }
    T divide()
    {
        return(num1/num2);
    }
};

int main()
{
    int choice,option,ch;
    int key;
    cout<<"\tWELCOME\t" <<endl;
```



```
cout<<"Select the operation you want to perform"<<endl;
cout<<"1.Integer\n2.Decimal\n3.Quit"<<endl;
cout<<"Enter your choice:";
cin>>option;
if(option==1)
{
    int num1,num2;
    cout<<"Enter two numbers:";
    cin>>num1>>num2;
    Calculator <int> obj(num1,num2);
    do
    {
        cout<<"\nWhich operation would you like to perform?\n1:Addition
        \n2:Subtraction\n3:Multiplication
        \n4:Division\n5:Quit"<<endl;
        cout<<"Enter your choice:";
        cin>>key;
        switch (key)
        {
            case 1:
                cout<<"Sum = "<<obj.add()<<endl;
                break;
            case 2:
                cout<<"Difference = "<<obj.subtract()<<endl;
                break;
            case 3:
                cout<<"Product = "<<obj.multiply()<<endl;
                break;
            case 4:
                cout<<"Quotient = "<<obj.divide()<<endl;
                break;
            case 5:
                break;
        }
        cout<<"Do you want to continue?" <<endl;
        cout<<"If yes enter 1 if no enter 0" <<endl;
        cout<<"Enter your choice:";
        cin>>choice;
    } while (choice);
}
else if(option==2)
```

```
{
    float num1,num2;
    cout<<"Enter two numbers:";
    cin>>num1>>num2;
    Calculator <float> obj(num1,num2);
    do
    {
        cout<<"\nWhich operation would you like to
        perform?\n1:Addition\n2:Subtraction
        \n3:Multiplication\n4:Division\n
        5:Quit"<<endl;
        cout<<"Enter your choice:";
        cin>>key;
        switch (key)
        {
            case 1:
                cout<<"Sum = "<<obj.add()<<endl;
                break;
            case 2:
                cout<<"Difference = "<<obj.subtract()<<endl;
                break;
            case 3:
                cout<<"Product = "<<obj.multiply()<<endl;
                break;
            case 4:
                cout<<"Quotient = "<<obj.divide()<<endl;
                break;
            case 5:
                break;
        }
        cout<<"Do you want to continue?" <<endl;
        cout<<"If yes enter 1 if no enter 0" <<endl;
        cout<<"Enter your choice:";
        cin>>choice;
    } while (choice);
}
return 0;
}
```

SAMPLE INPUT-OUTPUT

```
WELCOME
Select the operation you want to perform
1.Integer
2.Decimal
3.Quit
Enter your choice:1

Enter two numbers:10 5

Which operation would you like to perform?
1:Addition
2:Subtraction
3:Multiplication
4:Division
5:Quit
Enter your choice:1

Sum = 15
Do you want to continue?
If yes enter 1 if no enter 0
Enter your choice:1

Which operation would you like to perform?
1:Addition
2:Subtraction
3:Multiplication
4:Division
5:Quit
Enter your choice:2

Difference = 5
Do you want to continue?
If yes enter 1 if no enter 0
Enter your choice:1
```

```
Which operation would you like to perform?
1:Addition
2:Subtraction
3:Multiplication
4:Division
5:Quit
Enter your choice:3

Product = 50
Do you want to continue?
If yes enter 1 if no enter 0
Enter your choice:1

Which operation would you like to perform?
1:Addition
2:Subtraction
3:Multiplication
4:Division
5:Quit
Enter your choice:4

Quotient = 2
Do you want to continue?
If yes enter 1 if no enter 0
Enter your choice:1

Which operation would you like to perform?
1:Addition
2:Subtraction
3:Multiplication
4:Division
5:Quit
Enter your choice:5

Do you want to continue?
If yes enter 1 if no enter 0
Enter your choice:0
```

```
WELCOME
Select the operation you want to perform
1.Integer
2.Decimal
3.Quit
Enter your choice:2

Enter two numbers:4.62
2.31

Which operation would you like to perform?
1:Addition
2:Subtraction
3:Multiplication
4:Division
5:Quit
Enter your choice:1

Sum = 6.93
Do you want to continue?
If yes enter 1 if no enter 0
Enter your choice:1

Which operation would you like to perform?
1:Addition
2:Subtraction
3:Multiplication
4:Division
5:Quit
Enter your choice:2

Difference = 2.31
Do you want to continue?
If yes enter 1 if no enter 0
Enter your choice:1

Which operation would you like to perform?
1:Addition
2:Subtraction
3:Multiplication
4:Division
5:Quit
Enter your choice:3
```

```
Product = 10.6722
Do you want to continue?
If yes enter 1 if no enter 0
Enter your choice:1

Which operation would you like to perform?
1:Addition
2:Subtraction
3:Multiplication
4:Division
5:Quit
Enter your choice:4

Quotient = 2
Do you want to continue?
If yes enter 1 if no enter 0
Enter your choice:1

Which operation would you like to perform?
1:Addition
2:Subtraction
3:Multiplication
4:Division
5:Quit
Enter your choice:5

Do you want to continue?
If yes enter 1 if no enter 0
Enter your choice:0
```

EXCEPTION HANDLING

AIM

Write a C++ program to demonstrate the use of exception handling.

PROGRAM

```
#include <iostream>
using namespace std;
double zeroDivision(int x, int y)
{

    if (y == 0)
    {
        throw "Division by Zero!";
    }
    return (x / y);
}

int main()
{    int ch,option;;
    cout<<"\tWELCOME\t" <<endl;
    cout<<"\n";
    cout<<"EXCEPTION HANDLING" <<endl;
    cout<<"\n";

    do
    {

        int a,b;
        cout<<"Enter dividend: ";
        cin>>a;
        cout<<"\n";
        cout<<"Enter divisor: ";
        cin>>b;
        cout<<"\n";
        double c = 0;

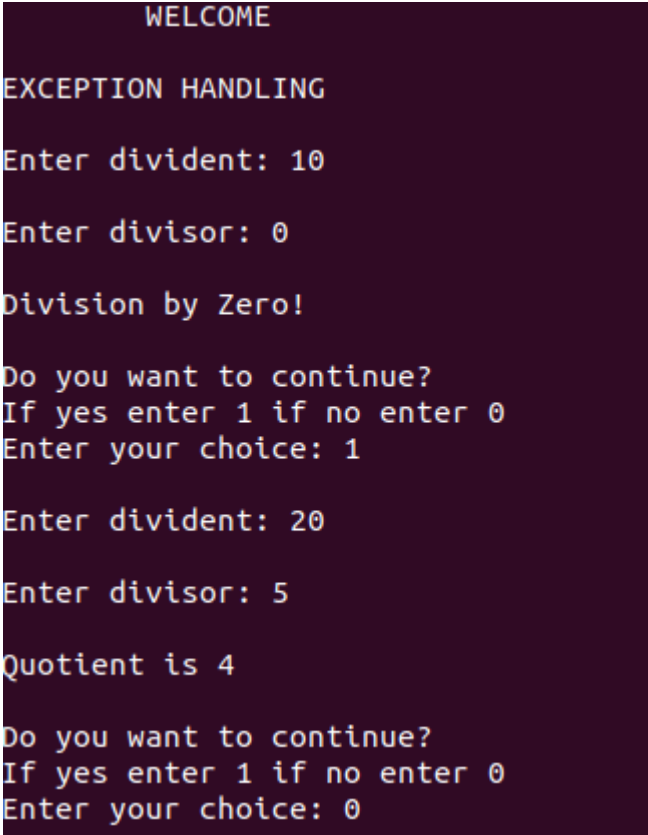
        try
        {
            c = zeroDivision(a, b);
```

```
        cout << "Quotient is " << c << endl;
        cout<<"\n";
    }
    catch (const char* message)
    {

        cerr << message << endl;
        cout<<"\n";
    }
    cout<<"Do you want to continue?" <<endl;
    cout<<"If yes enter 1 if no enter 0" <<endl;
    cout<<"Enter your choice: ";
    cin>>ch;
    cout<<"\n";
}while(ch);

return 0;
}
```

SAMPLE INPUT-OUTPUT



```
WELCOME

EXCEPTION HANDLING

Enter dividend: 10
Enter divisor: 0
Division by Zero!

Do you want to continue?
If yes enter 1 if no enter 0
Enter your choice: 1

Enter dividend: 20
Enter divisor: 5
Quotient is 4

Do you want to continue?
If yes enter 1 if no enter 0
Enter your choice: 0
```