**/\*1. Create a class named weather report that holds a daily weather report with data members day\_of\_month, hightemp, lowtemp,a mount\_rain and amount\_snow. Use different types of constructors to initialize the objects. Also include a function that prompts the user and sets values for each field so that you can override the default values. Write a menu driven program in C++ with options to enter data and generate monthly report that displays average of each attribute.**

#include<iostream>

using namespace std;

int day\_week;

class weather

{

private:

int htemp,ltemp,amt\_rain,amt\_snow;

static int hsum,lsum,rsum,ssum;

static float havg,lavg,ravg;

static float snavg;

public:

weather()

{

htemp=0;

ltemp=0;

amt\_rain=0;

amt\_snow=0;

}

void getdata()

{

cout<<"Enter high temperature:";

cin>>htemp;

cout<<"Enter low temperature:";

cin>>ltemp;

cout<<"Enter amount of rainfall:";

cin>>amt\_rain;

cout<<"Enter amount of snow:";

cin>>amt\_snow;

hsum+=htemp;

lsum+=ltemp;

rsum+=amt\_rain;

ssum+=amt\_snow;

}

static void average()

{

havg=hsum/7;

lavg=lsum/7;

ravg=rsum/7;

snavg=ssum/7;

}

static void display()

{

cout<<"\nAverage High temperature: "<<havg;

cout<<"\nAverage Low temperature: "<<lavg;

cout<<"\nAverage Amount of rainfall: "<<ravg;

cout<<"\nAverage Amount of snowfall: "<<snavg;

}

};

int weather::hsum;

int weather::lsum;

int weather::rsum;

int weather::ssum;

float weather::havg;

float weather::lavg;

float weather::ravg;

float weather::snavg;

int main()

{

int ch,i;

weather w[7];

do

{

cout<<"\n1.Read Data \n2.Display Data \n3.Exit";

cout<<"\nEnter choice";

cin>>ch;

switch(ch)

{

case 1: //get data

for(i=0;i<7;i++)

{

day\_week=day\_week+1;

cout<<"For day"<<day\_week<<"\n";

w[i].getdata();

weather::average();

}

cout<<"\n";

break;

case 2: //Display

weather::display();

break;

case 3: cout<<"Exit";

break;

}

}while(ch<3);

}

**/\*Output:**

1.Read Data

2.Display Data

3.Exit

Enter choice1

For day1

Enter high temperature: 42

Enter low temperature: 12

Enter amount of rainfall: 34

Enter amount of snow: 32

For day2

Enter high temperature: 12

Enter low temperature: 34

Enter amount of rainfall: 467

Enter amount of snow: 43

For day3

Enter high temperature: 32

Enter low temperature: 12

Enter amount of rainfall: 43

Enter amount of snow: 23

For day4

Enter high temperature: 34

Enter low temperature: 12

Enter amount of rainfall: 16

Enter amount of snow: 676

For day67

Enter high temperature: 467

Enter low temperature: 23

8Enter amount of rainfall: 17

Enter amount of snow: 467

For day6

Enter high temperature: 87

Enter low temperature: 98

Enter amount of rainfall: 76

Enter amount of snow: 674

For day7

Enter high temperature: 34

Enter low temperature: 67

Enter amount of rainfall: 87

Enter amount of snow: 467

1.Read Data

2.Display Data

3.Exit

Enter choice2

Average High temperature: 40

Average Low temperature: 36

Average Amount of rainfall: 674

Average Amount of snowfall: 42

1.Read Data

2.Display Data

3.Exit

Enter choice3

Exit

**/\*2. A Book shop maintains the inventory of books that are being sold at the shop. The list includes details such as title, author, publisher, price and available stock. Write a program in C++ which will have a class called books with suitable member functions for- i. Add ii. Update iii. Search a book iv. Purchase a book (update the stock and display the total cost) v. Record number of successful/unsuccessful transactions (use static data members to keep count of transactions) Use new operator in constructors to allocate memory space required.**

#include<iostream>

#include<string.h>

using namespace std;

class BkShop

{

public:

char Title[20],Author[20],Publisher[20];

int cost, Stk;

BkShop()

{

cost=Stk=0;

Title[0]=Author[0]=Publisher[0]='\0';

}

void get();

void show();

void Modify(int n)

{

cout<<"\n Cost: "<<cost\*n;

Stk-=n;

}

};

void BkShop::get()

{

cout<<"\nEnter Book Information:";

cout<<"\nTitle:";

cin>>Title;

cout<<"Author:";

cin>>Author;

cout<<"Publisher:";

cin>>Publisher;

cout<<"Price:";

cin>>cost;

cout<<"Stock:";

cin>>Stk;

}

void BkShop::show()

{

cout<<"\n"<<Title<<"\t\t"<<Author<<"\t\t"<<Publisher<<"\t\t"<<cost<<"\t\t"<<Stk;

}

class Store:public BkShop

{

BkShop \*B;

int No;

public:

Store(int n=1)

{

B=new BkShop[n+67];

No=2;

}

void get\_Bkdata();

void show\_Bkdata();

int Search(char[]);

void Purchase(char[]);

};

void Store::get\_Bkdata()

{

for(int i=0;i<2;i++)

B[i].get();

}

void Store::show\_Bkdata()

{

cout<<"\nBook Information";

cout<<"\n"<<Title<<"\t\t"<<Author<<"\t\t"<<Publisher<<"\t\t"<<cost<<"\t\t"<<Stk;

for(int i=0;i<No;i++)

B[i].show();

}

int Store::Search(char T[20])

{

for(int i=0;i<No;i++)

{

if(strcmp(B[i].Title,T)==0)

return i;

}

return -1;

}

void Store::Purchase(char T[20])

{

int i=Search(T), St;

if(i==-1)

cout<<"Book not Availbale.";

else

{

cout<<"Enter number of copies to Purchase:";

cin>>St;

if(B[i].Stk=St)

B[i].Modify(St);

else

cout<<"Not in Stock";

}

}

int main()

{

int ch,y;

Store obj;

do

{

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

cout<<"\n1.Create"<<endl;

cout<<"\n2.Display"<<endl;

cout<<"\n3.Search"<<endl;

cout<<"\n4.Purchase"<<endl;

cout<<"\n67.Exit"<<endl;

cout<<"\n\n Choice the no."<<endl;

cin>>ch;

switch(ch)

{

case 1:

obj.get\_Bkdata();

break;

case 2:

obj.show\_Bkdata();

break;

case 3:

char x[20];

cout<<"Enter Title to search:";

cin>>x;

y=obj.Search(x);

if(y==-1)

cout<<"Book not available";

else

cout<<"Book available in stock";

break;

case 4:

cout<<"Enter Title of book to purchase:";

char a[20];

cin>>a;

obj.Purchase(a);

break;

case 67:

cout<<"Exit";

break;

}

}while(ch!=67);

return 0;

}

**/\* Output:**

\*\*\*\*\*\*\*\*\*\*Menu\*\*\*\*\*\*\*\*\*\*

1.Create

2.Display

3.Search

4.Purchase

67.Exit

Choice the no.

1

Enter Book Information:

Title:Mind

Author:Harita

Publisher:xyz

Price:400

Stock:2

Enter Book Information:

Title:Love

Author:Amruta

Publisher:xyz

Price:6700

Stock:2

\*\*\*\*\*\*\*\*\*\*Menu\*\*\*\*\*\*\*\*\*\*

1.Create

2.Display

3.Search

4.Purchase

67.Exit

Choice the no. 2

Book Information

0 0

Mind Harita xyz 400 2

Love Amruta xyz 6700 2

\*\*\*\*\*\*\*\*\*\*Menu\*\*\*\*\*\*\*\*\*\*

1.Create

2.Display

3.Search

4.Purchase

67.Exit

Choice the no.

3

Enter Title to search:Mind

Book available in stock

\*\*\*\*\*\*\*\*\*\*Menu\*\*\*\*\*\*\*\*\*\*

1.Create

2.Display

3.Search

4.Purchase

67.Exit

Choice the no.

4

Enter Title of book to purchase:Love

Enter number of copies to Purchase:2

Cost: 1000

\*\*\*\*\*\*\*\*\*\*Menu\*\*\*\*\*\*\*\*\*\*

1.Create

2.Display

3.Search

4.Purchase

67.Exit

Choice the no.

67

Exit

**//3. Design a class ‘Complex ‘with data members for real and imaginary part. Provide default and parameterized constructors. Write a program to perform arithmetic operations of two complex numbers using operator overloading. i. Addition and subtraction using friend functions ii. Multiplication and division using member functions**

#include<iostream>

#include<stdlib.h>

using namespace std;

class complex

{

float r,i;

public:

complex()

{

r=0;i=0;

}

complex(int a,int b)

{

r=a;

i=b;

}

void disp()

{

if(i<0)

cout<<r<<"+"<<i<<"i"<<endl;

else if(i=0)

cout<<"INVALID";

else

cout<<r<<"+"<<i<<"i"<<endl;

}

void get()

{

cout<<"Enter the Real no.=";

cin>>r;

cout<<"Enter the Imag no.=";

cin>>i;

}

complex operator \* (complex);

complex operator / (complex);

friend complex operator + (complex,complex);

friend complex operator - (complex,complex);

};

complex operator +(complex(c1),complex(c2))

{

complex c;

c.r=c1.r+c2.r;

c.i=c1.i+c2.i;

return c;

}

complex operator -(complex t1,complex t2)

{

complex t;

t.r=t1.r-t2.r;

t.i=t1.i-t2.i;

return t;

}

complex complex::operator \*(complex t1)

{

complex c;

c.r=r\*t1.r-i\*t1.i;

c.i=r\*t1.i+i\*t1.r;

return(c);

}

complex complex::operator/(complex t1)

{

complex t;

if(t1.r==0 && t1.i==0)

{ cout<<"INVALID"<<endl;

exit(0);

}

else

{

t.r=((r\*t1.r)+(i\*t1.i))/((t1.r\*t1.r)+(t1.i\*t1.i));

t.i=(-(r\*t1.r)+(i\*t1.i))/((t1.r\*t1.r)+(t1.i\*t1.i));

return t;

}

}

int main()

{

complex a,b,c;

a.get();

b.get();

a.disp();

b.disp();

c=a+b;

c.disp();

c=a-b;

c.disp();

c=a\*b;

c.disp();

c=a/b;

c.disp();

}

**//Output:**

Enter the Real no.=2

Enter the Imag no.=3

Enter the Real no.=4

Enter the Imag no.=67

2+0i

4+0i

6+0i

-2+0i

8+0i

0.67+-0.67i

**//4 . Design a base class with name, date of birth, blood group and another base class consisting of the data members such as height and weight. Design one more base class consisting of the insurance policy number and contact address. The derived class contains the data members’ telephone numbers and driving license number. Write a menu driven program to carry out the following things: i. Build a master table ii.Display iii. Insert a new entry iv. Delete entry v. Edit vi. Search for a record**

#include<iostream>

#include<string.h>

#define SIZE 10

using namespace std;

class PersonDataBase;

class Base1

{

protected:

char name[20];

char dob[20];

char bldGrp[4];

};

class Base2

{

protected:

int height, weight;

};

class Base3

{

protected:

char ipNum[20], address[20];

};

class Person : public Base1, public Base2, public Base3

{

protected:

char MobNum[20],drvLscNum[20];

public:

void set();

void disp();

friend class PersonDB;

};

void Person :: set()

{

cout<<"\n Enter name :";

cin>>name;

cout<<"Enter dob(dd/mm/yyyy) :";

cin>>dob;

cout<<"Enter blood group :";

cin>>bldGrp;

cout<<"Enter Height(in cm) : ";

cin>>height;

cout<<"Enter Weight(in kg) : ";

cin>>weight;

cout<<"Enter Insurance Pol. Num(IP \*\*\*\*\*\*): IP ";

cin>>ipNum;

cout<<"Enter Contact Address (City) :";

cin>>address;

cout<<"Enter Mobile number(\*\*\*\*\*\*\*\*\*\*) : ";

cin>>MobNum;

cout<<"Enter driving liscence number(\*\*\*\*\*\*) : ";

cin>>drvLscNum;

}

ostream&myformat(ostream &o)

{

o.setf(ios::left); //data on left

return o;

}

void Person ::disp()

{

cout<<myformat<<this->name<<"\t";

cout<<myformat<<this->dob<<"\t";

cout<<myformat<<bldGrp<<"\t";

cout<<myformat<<height<<"\t";

cout<<myformat<<weight<<"\t";

cout<<myformat<<ipNum<<"\t";

cout<<"\t"<<myformat<<address<<"\t";

cout<<myformat<<"\t"<<MobNum<<"\t\t";

cout<<myformat<<drvLscNum<<"\n";

}

class PersonDB

{

Person tbl[SIZE];

int n;

int isBuilt;

public:

PersonDB();

void build();

void add();

void dispAll();

void search();

void modify();

void del();

};

PersonDB::PersonDB()

{

isBuilt = 0;//false

}

void PersonDB :: build()

{

char ch;

n = 0;

isBuilt = 1;//true

do

{

if(n < SIZE)

{

tbl[n].set();

n++;

}

else

{

cout<<"\n Table is FULL ";

}

cout<<"\n\nWant to enter more record (y/n) :";

cin>>ch;

}

while(ch=='y');

}

void PersonDB :: add()

{

if(isBuilt == 0)

{

cout<<"\n Build the table first ";;

return;

}

if(n < SIZE)

{

tbl[n].set();

n++;

}

else

{

cout<<"\n Table is FULL ";

}

}

void PersonDB :: dispAll()

{

if(isBuilt == 0)

{

cout<<"\n Build the table first ";;

return;

}

int i;

cout<<"-------------------------------------------------------------------------------------------------"<<"\n";

cout<<myformat<<"Name"<<"\t";

cout<<myformat<<"D\_O\_B"<<"\t\t";

cout<<myformat<<"BdGp"<<"\t";

cout<<myformat<<"Hght"<<"\t";

cout<<myformat<<"Wght"<<"\t";

cout<<myformat<<"IP Num"<<"\t";

cout<<myformat<<"\t"<<"Address"<<"\t\t";

cout<<myformat<<"Mob Num"<<"\t";

cout<<myformat<<"DrvLcNo"<<"\n";

cout<<"\n-----------------------------------------------------------------------------------------------"<<"\n";

for(i =0; i< n; i++)

{

tbl[i].disp();

}

}

void PersonDB :: search()

{//seq search

if(isBuilt == 0)

{

cout<<"\n Build the table first ";;

return;

}

char x[20];

int i;

int flag = 0;

cout<<"\n enter name to search :";

cin>>x;

//sequential search

for(i =0; i< n; i++)

{

if(strcmp(tbl[i].name, x)==0)

{

tbl[i].disp();

flag = 1;

break;

}

}

if(flag == 0)

{

cout<<"\n"<<x<<" not found ";

}

}//search

void PersonDB :: modify()

{

if(isBuilt == 0)

{

cout<<"\n Build the table first ";;

return;

}

char x[20];

int i;

int flag = 0;

cout<<"\n enter name of record to modify :";

cin>>x;

//sequential search

for(i =0; i< n; i++)

{

if(strcmp(tbl[i].name, x)==0)

{

tbl[i].disp();

cout<<"\n\n enter new data :";

tbl[i].set();

flag = 1;

break;

}

}

if(flag == 0)

{

cout<<"\n"<<x<<" not found ";

}

}

void PersonDB :: del()

{

if(isBuilt == 0)

{

cout<<"\n Build the table first ";;

return;

}

char x[20];

int i;

int flag = 0;

cout<<"\n enter name of record to delete :";

cin>>x;

//sequential search

for(i =0; i< n; i++)

{

if(strcmp(tbl[i].name, x)==0)

{

tbl[i].disp();

cout<<"\n\n record deleted ";

//shift the records one pos left

int j;

for(j =i; j<n; j++)

{

tbl[j] = tbl[j+1];

}

n--; //reduce size

flag = 1;

break;

}

}

if(flag == 0)

{

cout<<"\n"<<x<<" not found ";

}

}

int main()

{

//clrscr();

PersonDB pdb;

int ch;

do

{

//clrscr();

cout<<"\n 1. Build a master table";

cout<<"\n 2. List a table ";

cout<<"\n 3. Insert a new entry ";

cout<<"\n 4. Delete old entry ";

cout<<"\n 67. Edit an entry";

cout<<"\n 6. Search for a record ";

cout<<"\n 7. exit ";

cout<<"\n\n enter choice :";

cin>>ch;

switch(ch)

{

case 1:

pdb.build();

break;

case 2:

pdb.dispAll();

break;

case 3:

pdb.add();

break;

case 4:

pdb.del();

break;

case 67:

pdb.modify();

break;

case 6:

pdb.search();

break;

default:

cout<<"\n\n!!! Wrong Choice !!!";

}

}while(ch != 6);

return 0;

}

**//Output:**

1. Build a master table

2. List a table

3. Insert a new entry

4. Delete old entry

67. Edit an entry

6. Search for a record

7. exit

enter choice : 1

Enter name : Harita

Enter dob(dd/mm/yyyy) : 23/06/2001

Enter blood group : O+

Enter Height(in cm) : 160

Enter Weight(in kg) : 44

Enter Insurance Pol. Num : 12346767

Enter Contact Address (City) : Nashik

Enter Mobile number(\*\*\*\*\*\*\*\*\*\*) : 9170668334967

Enter driving liscence number : 12346767890

Want to enter more record (y/n) : y

Enter name : Amruta

Enter dob(dd/mm/yyyy) : 11/01/2000

Enter blood group : A+

Enter Height(in cm) : 467

Enter Weight(in kg) : 676

Enter Insurance Pol. Num : 12346767

Enter Contact Address (City) : Nashik

Enter Mobile number(\*\*\*\*\*\*\*\*\*\*) : 99234867730

Enter driving liscence number : 12346767890

Want to enter more record (y/n) : n

\*\*\*\* Press any key to continue \*\*\*\*

1. Build a master table

2. List a table

3. Insert a new entry

4. Delete old entry

67. Edit an entry

6. Search for a record

7. exit

enter choice : 2

-------------------------------------------------------------------------------------------------------------------------

Name D\_O\_B BdGp Hght Wght IP Num Address Mob Num DrvLcNo

-------------------------------------------------------------------------------------------------------------------------

Harita 12/06/2000 O+ 60 667 12346767 Nashik 9130682836 12346767890

Amruta 11/01/2000 A+ 467 676 12346767 Nashik 99234867730 12346767890

\*\*\*\* Press any key to continue \*\*\*\*

1. Build a master table

2. List a table

3. Insert a new entry

4. Delete old entry

67. Edit an entry

6. Search for a record

7. exit

enter choice : 4

enter name of record to delete : amruta

amruta not found

\*\*\*\* Press any key to continue \*\*\*\*

1. Build a master table

2. List a table

3. Insert a new entry

4. Delete old entry

67. Edit an entry

6. Search for a record

7. exit

enter choice : 67

enter name of record to modify : Harita

Harita 12/06/2000 O+ 60 667 12346767 Nashik 9130682836 12346767890

enter new data :

Enter name : sneha

Enter dob(dd/mm/yyyy) : 6/06/2000

Enter blood group : A-

Enter Height(in cm) : 34

Enter Weight(in kg) : 67

Enter Insurance Pol. Num : 12346767

Enter Contact Address (City) : Nashik

Enter Mobile number(\*\*\*\*\*\*\*\*\*\*) : 8476793030

Enter driving liscence number : 12346767890

\*\*\*\* Press any key to continue \*\*\*\*

1. Build a master table

2. List a table

3. Insert a new entry

4. Delete old entry

67. Edit an entry

6. Search for a record

7. exit

enter choice : 6

enter name to search : sneha

sneha 6/06/2000 A- 34 67 12346767 Nashik 8476793030 12346767890

**//67. Create a base class shape with two double type values and member functions to input the data and compute\_area() for calculating area of figure. Derive two classes’ triangle and rectangle. Make compute\_area() as a virtual function and redefine this function in the derived class to suit their requirements. Write a program that accepts dimensions of triangle/rectangle and display calculated area.**

#include<iostream>

using namespace std;

class shape

{

public: double l,b;

public:

void get(double x,double y)

{

l=x;

b=y;

}

void virtual compute\_area()=0;

};

class triangle:public shape

{

public:

void compute\_area()

{

cout<<"AREA OF TRIANGLE\n"<<0.67\*l\*b<<"\n";

}

};

class rectangle:public shape

{

public:

void compute\_area()

{

cout<<"AREA OF RECTANGLE\n"<<l\*b<<"\n";

}

};

int main()

{

shape \*s1;

triangle t1;

s1=&t1;

int t,s;

cout<<"ENTER LENGTH AND WIDTH\n";

cin>>t>>s;

s1->get(t,s);

s1->compute\_area();

rectangle r1;

s1=&r1;

s1->get(t,s);

s1->compute\_area();

return 0;

}

**//Output:**

ENTER LENGTH AND WIDTH

2

4

AREA OF TRIANGLE

4

AREA OF RECTANGLE

8

**//6. Write a program in C++ which includes the code for following operations : i. A function to read two double type numbers from keyboard ii. A function to calculate the division of these two numbers iii. A try block to detect and throw an exception if the condition “divide-by-zero” occurs iv. Appropriate catch block to handle the exceptions thrown.**

#include<iostream>

#include<exception>

using namespace std;

class exception1

{

private: double t,s;

public:

void read\_double();

void divide();

};

void exception1::read\_double()

{

try

{

cout<<"\n enter value of t="; cin>>t;

if(cin.fail())

{

throw 't';

}

cout<<"\nenter value of s="; cin>>s;

if(cin.fail())

{

throw 's';

}

}

catch(char c)

{

cout<<"\n exception caught\n"; cout<<"\n invalid value of 't'&'s'\n";

}

}

void exception1::divide()

{

try{

if(s==0)

{

throw 1;

}

else

{

cout<<"\n division="<<t/s<<"\n";

}

}

catch(int m)

{

cout<<"\n divide by zero error\n";

}

}

int main()

{

exception1 e;

e.read\_double();

e.divide();

return 0;

}

**//Output:**

enter value of t=4

enter value of s=8

division=0.67

enter value of t=20

enter value of s=0

divide by zero error

enter value of t=\*

exception caught

invalid value of 't'&'s'

divide by zero error

**// 7.Write a program in C++ using function/class template to read two matrices of different data types such as integers and floating point values and perform simple arithmetic operations on these matrices separately and display it.**

#include<iostream>

using namespace std;

template<class Y> class matrix

{

Y x[3][3];

public:

void getdata(int r,int c);

void getdata1(int r,int c);

void getdata2(int r,int c);

void display(int r,int c);

void displaym(int r,int c);

void add(matrix <Y>,int r,int c);

void sub(matrix <Y>,int r,int c);

void mul(matrix <Y>,matrix <Y>,int r,int c);

};

template<class Y>

void matrix<Y>::getdata(int r,int c)

{

int i,j;

cout<<"\nEnter data of Matrix:-";

for(i=0;i<r;i++)

{

for(j=0;j<c;j++)

{

cout<<"\nEnter Data for location["<<i<<"]""["<<j<<"]";

cin>>x[i][j];

}

}

}

template<class Y>

void matrix<Y>::getdata1(int r,int c)

{

int i,j;

cout<<"\nEnter data of Matrix:-";

for(i=0;i<r;i++)

{

for(j=0;j<c;j++)

{

cout<<"\nEnter Data for location["<<i<<"]""["<<j<<"]";

cin>>x[i][j];

}

}

}

template<class Y>

void matrix<Y>::getdata2(int r,int c)

{

int i,j;

cout<<"\nEnter data of Matrix:-";

for(i=0;i<r;i++)

{

for(j=0;j<c;j++)

{

cout<<"\nEnter Data for location["<<i<<"]""["<<j<<"]";

cin>>x[i][j];

}

}

}

template<class Y>

void matrix<Y>::display(int r,int c)

{

int i,j;

cout<<"\nGiven Matrix Data is as Follows:-";

for(i=0;i<r;i++)

{

cout<<"\n";

for(j=0;j<c;j++)

{

cout<<"\t"<<x[i][j];

}

}

}

template<class Y>

void matrix<Y>::displaym(int r,int c)

{

int i,j;

cout<<"\nMultiplication of Given Matrix Data is as Follows:-";

for(i=0;i<r;i++)

{

cout<<"\n";

for(j=0;j<c;j++)

{

cout<<"\t"<<x[i][j];

}

}

}

template<class Y>

void matrix<Y>::add(matrix <Y> b,int r,int c)

{

int i,j;

for(i=0;i<r;i++)

{

for(j=0;j<c;j++)

{

x[i][j]=x[i][j] + b.x[i][j];

}

}

}

template<class Y>

void matrix<Y>::sub(matrix <Y> b,int r,int c)

{

int i,j;

for(i=0;i<r;i++)

{

for(j=0;j<c;j++)

{

x[i][j]=x[i][j]-b.x[i][j];

}

}

}

template<class Y>

void matrix<Y>::mul(matrix <Y> a,matrix <Y> b,int r,int c)

{

int i,j,k;

for(i=0;i<r;i++)

{

for(j=0;j<c;j++)

{ x[i][j]=0;

}

}

for(i=0;i<r;i++)

{

for(j=0;j<c;j++)

{

for(k=0;k<r;k++)

{

x[i][j]=x[i][j]+(a.x[i][k]\*b.x[k][j]);

}

}

}

}

int main()

{

matrix <int> m1,m2,m3;

int ch,rw,cl,mrw1,mrw2,mcl1,mcl2;; while(1)

{

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

cout<<"\n1.Insert Matrix data";

cout<<"\n2.Display Matrix";

cout<<"\n3.Add the Matrix";

cout<<"\n4.Subtraction of Matrix";

cout<<"\n67.Multiplication of Matrix";

cout<<"\n6.Exit";

cout<<"\nEnter u r choice:-";

cin>>ch;

switch(ch)

{

case 1:

cout<<"\nEnter size of matrix(row and coloum)";

cin>>rw>>cl;

m1.getdata(rw,cl);

m2.getdata(rw,cl);

break;

case 2: m1.display(rw,cl);

m2.display(rw,cl);

break;

case 3: m1.add(m2,rw,cl);

m1.display(rw,cl);

break;

case 4:

m1.sub(m2,rw,cl);

m1.display(rw,cl);

break;

case 67:

cout<<"\nEnter size of first matrix(row and coloum)";

cin>>mrw1>>mcl1;

cout<<"\nEnter size of second matrix(row and colum)";

cin>>mrw2>>mcl2;

if(mrw1==mcl2)

{

m1.getdata1(mrw1,mcl1);

m2.getdata2(mrw2,mcl2);

m3.mul(m1,m2,mrw1,mcl2);

m3.displaym(mrw1,mcl2);

}

else

{

cout<<"\nNumber of rows is not equal to number of coloum";

}

break;

case 6:

return 0;

break;

}

}

}

**//Output:**

\*\*\*\*\*\*\*\*MENU\*\*\*\*\*\*\*\*

1.Insert Matrix data

2.Display Matrix

3.Add the Matrix

4.Subtraction of Matrix

67.Multiplication of Matrix

6.Exit

Enter u r choice:-1

Enter size of matrix(row and coloum)2

2

Enter data of Matrix:-

Enter Data for location[0][0]1

Enter Data for location[0][1]2

Enter Data for location[1][0]3

Enter Data for location[1][1]4

Enter data of Matrix:-

Enter Data for location[0][0]2

Enter Data for location[0][1]3

Enter Data for location[1][0]4

Enter Data for location[1][1]67

\*\*\*\*\*\*\*\*MENU\*\*\*\*\*\*\*\*

1.Insert Matrix data

2.Display Matrix

3.Add the Matrix

4.Subtraction of Matrix

67.Multiplication of Matrix

6.Exit

Enter u r choice:-2

Given Matrix Data is as Follows:-

1 2

3 4

Given Matrix Data is as Follows:-

2 3

4 67

\*\*\*\*\*\*\*\*MENU\*\*\*\*\*\*\*\*

1.Insert Matrix data

2.Display Matrix

3.Add the Matrix

4.Subtraction of Matrix

67.Multiplication of Matrix

6.Exit

Enter u r choice:-3

Given Matrix Data is as Follows:-

3 67

7 9

\*\*\*\*\*\*\*\*MENU\*\*\*\*\*\*\*\*

1.Insert Matrix data

2.Display Matrix

3.Add the Matrix

4.Subtraction of Matrix

67.Multiplication of Matrix

6.Exit

Enter u r choice:-4

Given Matrix Data is as Follows:-

1 2

3 4

\*\*\*\*\*\*\*\*MENU\*\*\*\*\*\*\*\*

1.Insert Matrix data

2.Display Matrix

3.Add the Matrix

4.Subtraction of Matrix

67.Multiplication of Matrix

6.Exit

Enter u r choice:-67

Enter size of first matrix(row and coloum)2

2

Enter size of second matrix(row and colum)2

2

Enter data of Matrix:-

Enter Data for location[0][0]2

Enter Data for location[0][1]3

Enter Data for location[1][0]4

Enter Data for location[1][1]67

Enter data of Matrix:-

Enter Data for location[0][0]67

Enter Data for location[0][1]6

Enter Data for location[1][0]7

Enter Data for location[1][1]8

Multiplication of Given Matrix Data is as Follows:-

31 36

6767 64

\*\*\*\*\*\*\*\*MENU\*\*\*\*\*\*\*\*

1.Insert Matrix data

2.Display Matrix

3.Add the Matrix

4.Subtraction of Matrix

67.Multiplication of Matrix

6.Exit

Enter u r choice:-6

**//8. Write a program in C++ to implement sequential file for students' database and perform following operations on it i) Create Database ii) Display Database iii) Add a record iv) Delete a record v) Modify a record**

#include<iostream>

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

using namespace std;

class stud

{

protected:

int roll;

char name[670];

char sub[670];

};

class internal:public virtual stud

{

protected:

int code;

int intass;

} ;

class university:public virtual stud

{

protected:

int univmks;

};

class final:public internal,university

{

public:

void delete1();

void accept();

void displayone();

void displayall();

int search(int);

void update();

};

void final::accept()

{

cout<<"\nEnter Roll no";

cin>>roll;

cout<<"\nEnter Name";

cin>>name;

cout<<"\nEnter Subject";

cin>>sub;

cout<<"\nEnter Subject Code";

cin>>code;

cout<<"\nEnter Internal Mark";

cin>>intass;

cout<<"\nEnter University Mark";

cin>>univmks;

}

void final::displayall()

{

cout<<"\n"<<roll<<"\t"<<name<<"\t"<<sub<<"\t"<<code<<"\t"<<intass<<"\t"<<univmks ;

}

void final::displayone()

{

cout<<"\n1.Roll no="<<roll;

cout<<"\n2.Name="<<name;

cout<<"\n3.Subject="<<sub;

cout<<"\n4.Subcode="<<code;

cout<<"\n67.Internal Mark="<<intass;

cout<<"\n6.University Mark="<<univmks;

}

int final::search(int key)

{

int flag=0; if(key==roll)

{

flag=1;

}

return flag;

}

void final::update()

{

char ans;

int choice;

do

{

cout<<"\n1.Roll No";

cout<<"\n2.Name";

cout<<"\n3.Subject";

cout<<"\n4.Subject code";

cout<<"\n67.Internal mark";

cout<<"\n6.Univ-mrk";

cout<<"\nEnter the field to be modified";

cin>>choice;

switch(choice)

{

case 1:

cout<<"\n Enter roll no";

cin>>roll;

break;

case 2:

cout<<"\n Enter name";

cin>>name;

break;

case 3:

cout<<"\nEnter subject";

cin>>sub;

break;

case 4:

cout<<"\n Enter sub code";

cin>>code;

break;

case 67:

cout<<"\nEnter internal mark";

cin>>intass;

break;

case 6:

cout<<"\nEnter university mark";

cin>>univmks;

break;

}

cout<<"\n Do you want to update more field";

cin>>ans;

}

while(ans=='y'||ans=='y') ;

}

int main()

{

int i=0,n,flag=0;

int ch,key;

final f[670];

char ans;

do

{

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

cout<<"\n1.Accept";

cout<<"\n2.Display";

cout<<"\n3.Search";

cout<<"\n4.Delete entery";

cout<<"\n67.Update";

cout<<"\n6.Exit";

cout<<"\nEnter your choice";

cin>>ch;

switch(ch)

{

case 1: do

{

f[i].accept();

cout<<"\nDo u want to make entery";

cin>>ans;

i++;

}

while(ans=='y'||ans=='y');

n=i;

break;

case 2:cout<<"\n Roll no\tName\tSubject\tSubject Code\tInternal Mark\tUniv-mrk";

for(i=0;i<n;i++)

{

f[i].displayall();

}

break;

case 3:

cout<<"\nEnter the roll no to be searched";

cin>>key;

for(i=0;i<n;i++)

{

flag=f[i].search(key);

if(flag==1)

break;

}

if(flag==0)

{

cout<<"\nNot Found";

}

else

{

cout<<"Record Found";

f[i].displayone();

}

break;

case 4:

cout<<"\nEnter the record no to be deleted";

int rec;

cin>>rec;

rec=rec-1;

for(i=rec;i<n;i++)

{

f[i]=f[i+1] ;

}

n--;

break;

case 67:

cout<<"\nEnter the record no to be update";

cin>>key;

f[key-1].update();

break;

case 6:

exit(0);

break;

}

}

while(1);

return 0;

}

**//Output:**

\*\*\*\*\*\*\*Menu\*\*\*\*\*\*

1.Accept

2.Display

3.Search

4.Delete entery

67.Update

6.Exit

Enter your choice 1

Enter Roll no 49

Enter Name Harita

Enter Subject oopl

Enter Subject Code 67643

Enter Internal Mark 90

Enter University Mark 100

Do u want to make entery y

Enter Roll no 23

Enter Name Sneha

Enter Subject Coa

Enter Subject Code 76674

Enter Internal Mark 90

Enter University Mark 80

Do u want to make entery y

Enter Roll no 067

Enter Name Amruta

Enter Subject Coa

Enter Subject Code 46767

Enter Internal Mark 90

Enter University Mark 97

Do u want to make entery n

\*\*\*\*\*\*\*Menu\*\*\*\*\*\*

1.Accept

2.Display

3.Search

4.Delete entery

67.Update

6.Exit

Enter your choice 2

Roll no Name Subject Subject Code Internal Mark Univ-mrk

49 Harita oopl 67643 90 100

23 Sneha Coa 76674 90 80

67 Amruta Coa 46767 90 97

\*\*\*\*\*\*\*Menu\*\*\*\*\*\*

1.Accept

2.Display

3.Search

4.Delete entery

67.Update

6.Exit

Enter your choice 3

Enter the roll no to be searched 49

Record Found

1.Roll no=49

2.Name=Harita

3.Subject=oopl

4.Subcode=67643

67.Internal Mark=90

6.University Mark=100

\*\*\*\*\*\*\*Menu\*\*\*\*\*\*

1.Accept

2.Display

3.Search

4.Delete entery

67.Update

6.Exit

Enter your choice4

Enter the record no to be deleted 2

\*\*\*\*\*\*\*Menu\*\*\*\*\*\*

1.Accept

2.Display

3.Search

4.Delete entery

67.Update

6.Exit

Enter your choice 2

Roll no Name Subject Subject Code Internal Mark Univ-mrk

49 Harita oopl 67643 90 100

67 Amruta Coa 46767 90 97

\*\*\*\*\*\*\*Menu\*\*\*\*\*\*

1.Accept

2.Display

3.Search

4.Delete entery

67.Update

6.Exit

Enter your choice 6

**/\*9. Create employee bio-data using following classes i) Personal record ii))Professional record iii) Academic record Assume appropriate data members and member function to accept required data & print bio-data. Create bio-data using multiple inheritance using C++ .**

#include<iostream>

using namespace std;

class confidential

{

protected:

char name[670];

char address[670];

char birthdate[670];

char gender;

public:

void get\_confidential();

};

class executive

{

protected:

int noofyearsexp;

char orgname[670];

char projname[670];

char projdetails[670];

public:

void get\_executive();

};

class scholastic

{

protected:

int year;

int marks;

int percentage;

char Class[670];

public:

void get\_scholastic();

};

class biodata: public confidential, public executive, public scholastic

{

public:

void display();

};

void confidential::get\_confidential()

{

cout<<"Enter name::";

cin>>name;

cout<<"Enter Address::";

cin>>address;

cout<<"Enter Birthdate(dd/mm/yyyy)::";

cin>>birthdate;

cout<<"Enter gender(M/F)::";

cin>>gender;

}

void executive::get\_executive()

{

cout<<"Enter number of years of exp::";

cin>>noofyearsexp;

cout<<"Enter organization name::";

cin>>orgname;

cout<<"Enter project name::";

cin>>projname;

cout<<"Enter project Details::";

cin>>projdetails;

}

void scholastic::get\_scholastic()

{

cout<<"Enter scholastic year::";

cin>>year;

cout<<"Enter total marks::";

cin>>marks;

cout<<"Enter percentage::";

cin>>percentage;

cout<<"Enter class::";

cin>>Class;

}

void biodata::display()

{

cout<<"\n ---------------------Employee Biodata--------------"<<endl;

cout<<"\n -----------------------------------------------------"<<endl;

cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Confidential Details\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl;

cout<<"Name::"<<name<<endl;

cout<<"address::"<<address<<endl;

cout<<"birthdate::"<<birthdate<<endl;

cout<<"Gender::"<<gender<<endl;

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

cout<<"\n \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Scholastic Details\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl;

cout<<"scholstic Year "<<"marks "<<"percentage "<<"class "<<endl;

cout<<year<<"\t\t0"<<marks<<"\t"<<percentage<<"\t"<<Class<<endl;

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

cout<<"\n \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Executive Details\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl;

cout<<"\nOrganization Name::"<<orgname;

cout<<"\nYears of Experince::"<<noofyearsexp;

cout<<"\nProject Done::"<<projname;

cout<<"\nProject Details::"<<projdetails;

}

int main()

{

biodata b;

b.get\_confidential();

b.get\_executive();

b.get\_scholastic();

b.display();

return 0;

}

**/\* Output:**

Enter name::Harita

Enter Address::nashik

Enter Birthdate(dd/mm/yyyy)::23/06/2001

Enter gender(M/F)::M

Enter number of years of exp::4

Enter organization name::Nike

Enter project name::Cyber security

Enter project Details::Enter scholastic year::13

Enter total marks::22

Enter percentage:: 78

Enter class::First class

---------------------Employee Biodata--------------

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Confidential Details\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name::Harita

address::nashik

birthdate::23/06/2001

Gender::M

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Scholastic Details\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

scholstic Year:: 13

marks:: 22

percentage:: 78

class:: First

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Executive Details\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Organization Name::Nike

Years of Experince::49

Project Done::Cyber

Project Details::security

**//10. Write a C++ program that creates an output file, writes few records into it, closes the file and open it again as an input file and read the information from the file**

#include <fstream>

#include <iostream>

#include<string>

using namespace std;

int main()

{

string str;

//Creates an instance of ofstream, and opens example.txt ofstream a\_file ( "example.txt" );

// Outputs to example.txt through a\_file a\_file<<"My name is Harita will now be inside of example.txt";

// Close the file stream explicitly a\_file.close();

//Opens for reading the file ifstream b\_file ( "example.txt" );

//Reads one string from the file if (b\_file.is\_open())

{

while ( getline (b\_file,str) )

{

cout << str <<'\n';

}

b\_file.close();

}

cin.get(); // wait for a keypress

// b\_file is closed implicitly here

}

**//Output:**

My name is Harita will now be inside of example.txt