

**01.) Write a C++ program to display Names , Roll No., & grades of 3 students who have appeared in examination by using array of class objects.**

#include <iostream>

using namespace std;

class stu{

char name[30];

int rollNo;

int grade;

public:

void getInfo(); /\* Member Function To Get Student Data \*/

void putInfo(); /\* Member Function To Display Student Data \*/

};

/\* Define Member Function\*/

void stu::getInfo()

{

cout<<"Enter your name :-"<<"\t";

cin>>name;

cout<<"Enter your roll no. :-"<<"\t";

cin>>rollNo;

cout<<"Enter your total marks out of 500 :-"<<"\t";

cin>>grade; cout<<endl<<endl; }

void stu::putInfo(){

cout<<"Your name :-"<<"\t"<<name<<endl<<endl;

cout<<"Your roll no."<<"\t"<<rollNo<<endl<<endl;

cout<<"Your total marks "<<"\t"<<grade<<"/"<<"500"<<endl<<endl;

}

main(){

stu st[3];

for(int i=0;i<3;i++)

{

cout<<"\t\tSubmit Details of "<<(i+1)<<" student"<<endl; cout<<"==========================================="<<endl;

st[i].getInfo();

}

for(int i=0;i<3;i++)

{

cout<<"\t\tDetails of "<<(i+1)<<" student"<<endl<<endl;

cout<<"=========================================="<<endl;

st[i].putInfo();

}

}

**OUTPUT:-**

Submit Details of 1 student

=========================================================================

Enter your name :- Abdus

Enter your roll no. :- 02

Enter your total marks out of 500 :- 456

Submit Details of 2 student

========================================================================

Enter your name :- Hritik

Enter your roll no. :- 03

Enter your total marks out of 500 :- 654

Submit Details of 3 student

========================================================================

Enter your name :- Honey

Enter your roll no. :- 04

Enter your total marks out of 500 :- 256

*continue……..*

Details of 1 student

================================================================

Your name :- Abdus

Your roll no. 2

Your total marks 456/500

Details of 2 student

================================================================Your name :- Hritik

Your roll no. 3

Your total marks 654/500

Details of 3 student

===============================================================

Your name :- Honey

Your roll no. 4

Your total marks 256/500

**02.) Write a C++ program to declare struct. Initialise & display contents of members variables**.

#include<iostream>

using namespace std;

struct stu{

int rollNo;

int marks;

char name[20];

};

main(){

struct stu s;

cout<<"Enter Your Name"<<"\t";

gets(s.name);

cout<<"Enter Your Roll No."<<"\t";

cin>>s.rollNo;

cout<<"Enter Your Grades"<<"\t";

cin>>s.marks;

cout<<endl<<"DISPLAYING INFORMATION"<<endl<<endl;

cout<<"Name"<<"\t" <<"\t"<<s.name<<endl;

cout<<"Roll No."<<"\t" <<"\t"<<s.rollNo<<endl;

cout<<"Grades"<<"\t" <<"\t"<<s.marks<<end;

}

OUTPUT

Enter Your Name:- Abdus Samad

Enter Your Roll No.:- 02

Enter Your Grades:- 456

DISPLAYING INFORMATION

Name :- Abdus Samad

Roll No. :- 2

Grades :- 456

**03.) Write a program to declare a class. Declare pointer to class. Initialize & display the content of class member.**

#include<iostream>

using namespace std;

class si{

double principle;

double rate;

double time;

public:

si(double p=2.0,double r=2.0,double t=2.0)

{

cout<<"Constructor called."<<endl;

principle=p;

rate=r;

time=t;

}

double siFormula()

{

return (principle\*rate\*time)/100;

}

};

main( ){

si s1(10000,8,5); //create objects

si s2(12000,10,8);

si \*ptrSi1=&s1;

si \*ptrSi2=&s2;

cout<<"Simple Interest of 1st Query:-\t"<<ptrSi1->siFormula()<<endl;

cout<<"Simple Interest of 2nd Query:-\t"<<ptrSi2->siFormula()<<endl;

}

**OUTPUT:-**

Constructor called.

Constructor called.

Simple Interest of 1st Query:- 4000

Simple Interest of 2nd Query:- 9600

**04.) Given that an Employee class contain following members:-name,id,basic salary,DA,IT,Net Salary & print data members**.

#include <iostream>

using namespace std;

class emp{

int empId;

char name[20];

float emp\_basicSal;

float emp\_da;

float emp\_it;

float emp\_netSal;

public:

void getInfo(); //Member function to get info.

void cal(); //Member function to calculate da,it etc.

void putInfo(); //Member function to display info.

};

void emp::getInfo(){

cout<<"Enter the employee ID :-"<<"\t";

cin>>empId;

cout<<"Enter the employee name :-"<<"\t";

cin>>name;

cout<<"Enter the basic employee salary :-"<<"\t";

cin>>emp\_basicSal;

}

void emp::cal(){

emp\_da=0.52\*emp\_basicSal;

emp\_it=0.30\*(emp\_basicSal+emp\_da);

emp\_netSal=(emp\_da+emp\_basicSal)-emp\_it;

}

void emp::putInfo(){

cout<<endl<<endl;

cout<<"\tEMPLOYEE DATAILS"<<endl<<endl;

cout<<"Employee ID :-"<<"\t"<<empId<<endl;

cout<<"Employee name :-"<<"\t"<<name<<endl;

cout<<"Employee Basic Salary :-"<<"\t"<<emp\_basicSal<<endl;

cout<<"Employee Dearness Allowance :-"<<"\t"<<emp\_da<<endl;

cout<<"Employee Income Tax :-"<<"\t"<<emp\_it<<endl;

cout<<"Employee Net Salary :-"<<"\t"<<emp\_netSal<<endl;

}

main(){

emp e; e.getInfo(); e.cal(); e.putInfo();

}

**OUTPUT:-**

Enter the employee ID :- 99

Enter the employee name :- Binod

Enter the basic employee salary :- 555000

EMPLOYEE DATAILS

Employee ID :- 99

Employee name :- Binod

Employee Basic Salary :- 555000

Employee Dearness Allowance :- 288600

Employee Income Tax :- 253080

Employee Net Salary :- 590520

**05.)Write a program to read the data of N employee & compute Net Salary of each employee.(Given DA=0.52\*Basic Salary,Income Tax=30% of Gross Salary).**

#include <iostream>

using namespace std;

class emp{

int empId;

char name[20];

float emp\_basicSal;

float emp\_da;

float emp\_grossSal;

float emp\_it;

float emp\_netSal;

public:

void getInfo();

void cal();

void putInfo();

};

//Define Member Function.

void emp::getInfo(){

cout<<"Enter the employee ID :-"<<"\t";

cin>>empId;

cout<<"Enter the employee name :-"<<"\t";

cin>>name;

cout<<"Enter the basic employee salary :-"<<"\t";

cin>>emp\_basicSal;

cout<<endl<<endl;

}

void emp::cal(){

emp\_da=0.52\*emp\_basicSal;

emp\_grossSal=emp\_basicSal+emp\_da;

emp\_it=0.30\*(emp\_grossSal);

emp\_netSal=(emp\_grossSal)-emp\_it;

}

void emp::putInfo(){

cout<<endl<<endl;

cout<<"\tEMPLOYEE DATAILS"<<endl<<endl;

cout<<"Employee ID :-"<<"\t"<<empId<<endl;

cout<<"Employee name :-"<<"\t"<<name<<endl;

cout<<"Employee Basic Salary :-"<<"\t"<<emp\_basicSal<<endl;

cout<<"Employee Dearness Allowance :-"<<"\t"<<emp\_da<<endl; cout<<"Employee Gross Salary :-"<<"\t"<<emp\_grossSal<<endl; cout<<"Employee Income Tax :-"<<"\t"<<emp\_it<<endl;

cout<<"Employee Net Salary :-"<<"\t"<<emp\_netSal<<endl;

}

main(){

int n;

cout<<"Enter no. of employee whose details to be inserted:-\t";

cin>>n;

emp e[n];

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

cout<<"\tEnter "<<i+1<<" employee details."<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

e[i].getInfo();

e[i].cal();

}

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

cout<<"Detail of "<<i+1<<" employee."<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

e[i].putInfo();

}

}

**OUTPUT :-**

Enter no. of employee whose details to be inserted:- 3

Enter 1 employee details.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Enter the employee ID :- 41

Enter the employee name :- Binod

Enter the basic employee salary :- 12000

Enter 2 employee details.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Enter the employee ID :- 42

Enter the employee name :- Vikas

Enter the basic employee salary :- 18000

Enter 3 employee details.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Enter the employee ID :- 43

Enter the employee name :- Abdul

Enter the basic employee salary :- 25000

*continue………*

Detail of 1 employee.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Employee ID :- 41

Employee name :- Binod

Employee Basic Salary :- 12000

Employee Dearness Allowance :- 6240

Employee Gross Salary :- 18240

Employee Income Tax :- 5472

Employee Net Salary :- 12768

Detail of 2 employee.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Employee ID :- 42

Employee name :- Vikas

Employee Basic Salary :- 18000

Employee Dearness Allowance :- 9360

Employee Gross Salary :- 27360

Employee Income Tax :- 8208

Employee Net Salary :- 19152

Detail of 3 employee.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Employee ID :- 43

Employee name :- Abdul

Employee Basic Salary :- 25000

Employee Dearness Allowance :- 13000

Employee Gross Salary :- 38000

Employee Income Tax :- 11400

Employee Net Salary :- 26600

**07.) Write a C++ program to use scope resolution operator. Display the various values of the same variables declared at different scope levels.**

#include<iostream>

using namespace std;

class test{

private: // By default it would be private.

void pvtFun(){

cout<<"Hello from Pvt. member Function"<<endl;

}

public:

void getData(){

cout<<"Hello from Public member function.:-"<<endl<<endl;

cout<<"From public invoking Pvt. Member Function"<<endl<<endl;

pvtFun();

}

protected:

void putData(){

cout<<"Hello from Protected Member function.:-"<<endl<<endl;

}

};

class der:public test{ // Inherit der class from test

public:

void getProt(){

cout<<"Now from Derived class Public Function invoking Base Class Protected Member Function.:-"<<endl;

putData();

}

};

main(){

test t;

der d;

t.getData();

d.getProt();

}

Hello from Public member function.:-

From public invoking Pvt. Member Function

Hello from Pvt. member Function

Now from Derived class Public Function invoking Base Class Protected Member Function.:-

Hello from Protected Member function.

OUTPUT :-

**08.)Write a C++ program to allocate memory using new operator.**

#include<iostream>

using namespace std;

void singleArray();

void mdArray();

main()

{

singleArray();

mdArray();

}

void singleArray(){

int \*p,n;

cout<<"Enter the size"<<"\t";

cin>>n;

p=new int[n];

cout<<"Enter the elements"<<endl;

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

{

cin>>p[i]; //cin>>&(p+i); //not work here.

}

cout<<"The elements in 1-D array are"<<endl;

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

{

cout<<"[";

cout<<\*(p+i);

cout<<"]";

if(i<n-1){

cout<<",";

}

}

}

void mdArray(){

int \*\*p,nr,nc;

cout<<endl<<endl<<"CREATE MULTI-DIMENSIONAL ARRAY"<<endl<<endl;

cout<<"Enter the no. of rows & column :-"<<endl;

cin>>nr>>nc;

p=new int \*[nr];

for(int r=0;r<nr;r++)

{

p[r]=new int[nc];

}

cout<<"Enter the element in matrix :-"<<endl;

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

{

cout<<"Enter "<<i+1<<" row:-"<<endl;

for(int j=0;j<nc;j++)

{

cin>>p[i][j];

}

}

cout<<endl<<"The element in matrix are :-"<<endl;

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

{

for(int j=0;j<nc;j++)

{

cout<<p[i][j]<<"\t"<<ends;

}

cout<<endl;

}

}

Enter the size 5

Enter the elements

78

79

55

65

45

The elements in 1-D array are

[78],[79],[55],[65],[45] ***continue………***

CREATE MULTI-DIMENSIONAL ARRAY

Enter the no. of rows & column :-

3

3

Enter the element in matrix :-

Enter 1 row:-

40

41

48

Enter 2 row:-

65

68

67

Enter 3 row:-

99

98

97

The element in matrix are :-

40 41 48

65 68 67

99 98 97

**OUTPUT :-**

CREATE MULTI-DIMENSIONAL ARRAY

Enter the no. of rows & column :-

3

3

Enter the element in matrix :-

Enter 1 row:-

40

41

48

Enter 2 row:-

65

68

67

Enter 3 row:-

99

98

97

The element in matrix are :-

40 41 48

65 68 67

99 98 97

**09.)Write a C++ program to create multilevel inheritence.**

#include<iostream>

using namespace std;

class base{

protected:

int x;

public:

void getValue1(){

cout<<"Hello from base class"<<endl<<endl;

cout<<"Enter value of x"<<"\t";

cin>>x;

}

};

class der1:public base{

protected:

int y;

public:

void getValue2(){

cout<<endl<<"Hello from der1 class derived from base class"<<endl<<endl;

cout<<"Enter value of y\t"; cin>>y;

}

};

class der2:public der1{

protected:

int z;

public:

void getValue3(){

cout<<endl<<"Hello from der2 class derived from der1 class"<<endl<<endl;

cout<<"Enter value of z\t";

cin>>z;

}

};

class der3:public der2{

protected:

int a;

public:

void getValue4(){

cout<<endl<<"Hello from der3 class derived from der2 class"<<endl<<endl;

cout<<"Enter value of a\t";

cin>>a;

}

void pro(){

cout<<endl<<"Product of value "<<x<<" \* "<<y<<" \* "<<z<<" \* "<<a<<" = "<<" "<<a<<(x\*y\*z\*a);

}

};

main(){

der3 d;

d.getValue1();

d.getValue2();

d.getValue3();

d.getValue4();

d.pro();

}

***OUTPUT :-***

Hello from base class

Enter value of x 98

Hello from der1 class derived from base class

Enter value of y 70

Hello from der2 class derived from der1 class

Enter value of z 64

Hello from der3 class derived from der2 class

Enter value of a 51 Product of value 98 \* 70 \* 64 \* 51 = 5122391040

**10.) Write a program to create an array of pointers.Invoke using array of objects.**

#include<iostream>

using namespace std;

class A{

public:

/\*Have to add virtual keyword to avoid its duplication in its inherited classes.\*/

virtual void show(){

cout<<"Hello from class A"<<endl;

}

};

class B:public A{

public:

void show(){

cout<<"Hello from class B"<<endl;

}

};

class C:public A{

public:

void show(){

cout<<"Hello from class C"<<endl;

}

};

class D:public A{

public:

void show(){

cout<<"Hello from class D"<<endl;

}

};

class E:public A{

public:

void show(){

cout<<"Hello from class E"<<endl;

}

};

class F:public A{

public:

void show(){

cout<<"Hello from class F"<<endl;

}

};

main(){

A a;

B b;

C c;

D d;

E e;

F f;

A \*p[]={&a,&b,&c,&d,&e,&f}; //Create array of refernces of every objects;

for(int i=0;i<6;i++){

p[i]->show();

}

}

***OUTPUT :-***

**Hello from class A**

**Hello from class B**

**Hello from class C**

**Hello from class D**

**Hello from class E**

**Hello from class F**

**11.) Write a C++ program to use pointer for both base & derived classes & call the member function.Use Virtual keyword.**

#include<iostream>

using namespace std;

class base{

public:

virtual void print(){

cout<<"Hello from print() of base class use virtual keyword"<<endl;

}

void show(){

cout<<"Hello from show() of base class don't use virtual keyword"<<endl;

}

};

class der:public base{

public:

void print(){

cout<<"Hello from print() function of derived class."<<endl;

}

void show(){

cout<<"Hello from show() function of derived class."<<endl;

}

};

main(){

base \*p;

der d;

p=&d;

p->show();

p->print();

}

***OUTPUT :-***

Hello from show() of base class don't use virtual keyword

Hello from print() function of derived class.