**LAB 1**

1. Write a C++ program to display names, roll no and grades of 3 students who have appeared in examination. declare the class of name, roll no and grade. create an array of class objects read and display the contents of array.

Program:

#include<iostream>

using namespace std;

class Student\_info

{

int roll\_number;

char student\_name[50], grade[2];

public:

void read\_data(int count)

{

cout<<"\n\n -----------Enter Students "<<count+1<<"Information------------\n";

cout<<"Name of the student (max. 50 char only): ";

cin>>student\_name;

cout<<"Roll Number: ";

cin>>roll\_number;

cout<<"Grade(O, A+, A, B+, B, C, D, F): ";

cin>>grade;

cout<<"\n student Information with roll number "<<roll\_number<<"has saved!";

}

void display\_data(int count)

{

cout<<"\n\n \*\*\*\*\*\*\*\*\*\*Student"<<count+1<<"Information\*\*\*\*\*\*";

cout<<"\n Name of the student: "<<student\_name;

cout<<"\n Roll Number: "<<roll\_number;

cout<<"\n Grade secured: "<<grade;

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

}

};

int main()

{

Student\_info stud[3];

int i;

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

stud[i].read\_data(i);

cout<<"\n\n+++++++++++++++++++++++++++++++++++++\n";

cout<<"THe information of 3 students has been saved.";

cout<<"\n ++++++++++++++++++++++++++++++++++++++\n";

}

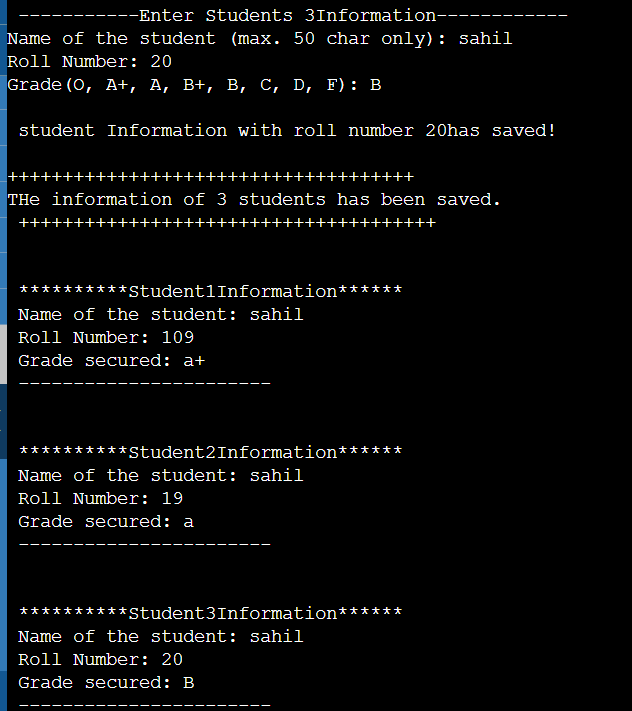
for(i=0; i<3; i++)

stud[i].display\_data(i);

return 0;

}

**Output:**

****

**2. Write a c++ program to declare struct, intialize and display contents of member varaiables.**

Program:

#include<iostream>

using namespace std;

struct college\_info

{

char college\_name[15];

char college\_code[2];

char dept[50];

int intake;

};

int main()

{

struct college\_info college;

cout<<"\n+++++++++Enter the collage name+++++++\n\n";

cout<<"Name of collage: ";

cin>>college.college\_name;

cout<<"Collage code: ";

cin>>college.college\_code;

cout<<"ENter the department: ";

cin>>college.dept;

cout<<"Department In\_take: ";

cin>>college.intake;

cout<<"\n\n\*\*\*\*\*\*\*\*\*Collage information\*\*\*\*\*\*\*\*\n\n";

cout<<"Name of collage: "<<college.college\_name<<endl;

cout<<"Univercity code: "<<college.college\_code<<endl;

cout<<"Name of Department: "<<college.dept<<endl;

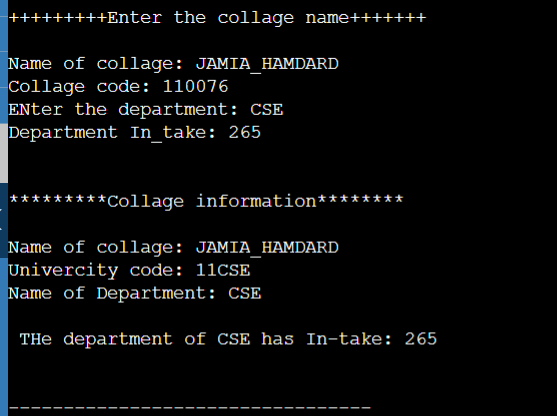
cout<<"\n THe department of "<<college.dept <<" has In-take: "<<college.intake<<endl;

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

return 0;

}

**Output:**

****

**3. Write a C++ program to declare a class, declare pointer to class, initailize and display contents of class member.**

Program:

#include<iostream>

using namespace std;

class RectrangleTest

{

public:

int length, breadth;

public:

void initialize(int len, int bre)

{

length = len;

breadth = bre;

}

int getArea()

{

return 2\*length\*breadth;

}

void display()

{

int area = getArea();

cout<<"\n \*\*\*\*\*\*\*Rectrangle Information\*\*\*\*\*\*\*\n";

cout<<"Length = "<<length;

cout<<"\n breadth = "<<breadth;

cout<<"\n Area = "<<area;

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

}

};

int main(){

RectrangleTest rect, \*class\_ptr;

Handle color = GetStdHandle(STD\_OUTPUT\_HANDLE);

class\_ptr = &rect;

//Accessing member function using class pointer

SetConsoleTextAttribute(int color, 10); //setting color green

cout<<"\n Using member function access";

SetConsoleTextAttribute(color, 7); //setting color white

class\_ptr->initialize(10, 5);

class\_ptr->display();

//Accessing data members using class pointer

SetConsoleTextAttribute(color, 10); //setting color green

cout<<"\n Using data members access";

SetConsoleTextAttribute(color, 7); //setting color white

class\_ptr->length = 2;

class\_ptr->breadth = 3;

class\_ptr->initialize(class\_ptr->length, class\_ptr->breadth);

class\_ptr->display();

return 0;

}

**LAB 2**

4. **Given that an employee class contains following members, data members, employee number, employee name ,basic, DA, IT, net salary and print data members.**

**Program:**

#include<iostream>

using namespace std;

class employee

{

int emp\_number;

char emp\_name[20];

float emp\_basic;

float emp\_da;

float emp\_it;

float emp\_net\_sal;

public:

void get\_emp\_details();

float find\_net\_salary(float basic, float da, float it);

void show\_emp\_details();

};

void employee :: get\_emp\_details()

{

cout<<"\n Enter emplyee Number: ";

cin>>emp\_number;

cout<<"\n Enter emplyee Name: ";

cin>>emp\_name;

cout<<"\n Enter employee Basic: ";

cin>>emp\_basic;

cout<<"\n Enter emplyee DA: ";

cin>>emp\_da;

cout<<"\n Enter emplyee IT: ";

cin>>emp\_it;

}

float employee :: find\_net\_salary(float basic, float da, float it)

{

return (basic + da)-it;

}

void employee :: show\_emp\_details()

{

cout<<"\n\n \*\*\*\*\*\*\*\*Details of Employee\*\*\*\*\*\*\*\*\n";

cout<<" Emplyee Name : "<<emp\_name<<endl;

cout<<" Emplyee Number : "<<emp\_number<<endl;

cout<<" Basic salary : "<<emp\_basic<<endl;

cout<<" Employee DA : "<<emp\_da<<endl;

cout<<" Income Tax : "<<emp\_da<<endl;

cout<<" Net SAlary : "<<find\_net\_salary(emp\_basic, emp\_da, emp\_it)<<endl;

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

}

int main()

{

employee emp;

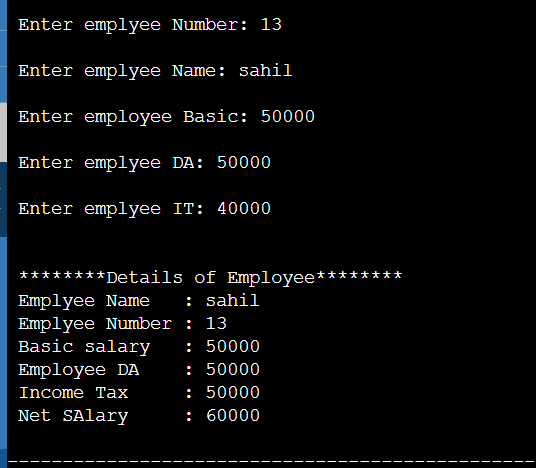
emp.get\_emp\_details();

emp.show\_emp\_details();

return 0;

}

**Output:**

****

**5. Write a C++ program to read the data of N employee and compute net salary of each employee (DA=52%of basic and income tax(IT)=30% of gross salary.**

Program:  
#include<iostream>

using namespace std;

class Employee

{

char emp\_name[30];

int emp\_number;

float basic, da, it, gross\_salary, net\_salary;

public:

void read\_emp\_details(int count)

{

cout<<"\n\n\*\*\*\*Enter Emplyee "<<count<<" Details\*\*\*\*\*";

cout<<"\n Employee Number: ";

cin>>emp\_number;

cout<<" Emplyee Name: ";

cin>>emp\_name;

cout<<"basic Salary: ";

cin>>basic;

cout<<"\n -----Emplyee "<<count<<" Details are saved-----\n\n";

}

float find\_net\_salary()

{

da = basic\*0.52;

gross\_salary = basic+da;

it = gross\_salary \* 0.30;

net\_salary = (basic+da) - it;

return net\_salary;

}

void display\_emp\_details(int count)

{

cout<<"\n\n \*\*\*\*\*\*EMployee "<<count<<" Details\*\*\*\n";

cout<<"\n Employee Nmuber : "<<emp\_number;

cout<<"\n Employee Name : "<<emp\_name;

cout<<"\n Net Salery : "<<net\_salary;

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

}

};

int main()

{

Employee emp[100];

int number\_of\_emp, count;

cout<<"\n Please enter the number of Employees(max.100): ";

cin>>number\_of\_emp;

for(count=0; count<number\_of\_emp; count++)

{

emp[count].read\_emp\_details(count+1);

}

for(count=0; count<number\_of\_emp; count++)

{

emp[count].find\_net\_salary();

}

for(count=0; count<number\_of\_emp; count++)

{

emp[count].display\_emp\_details(count+1);

}

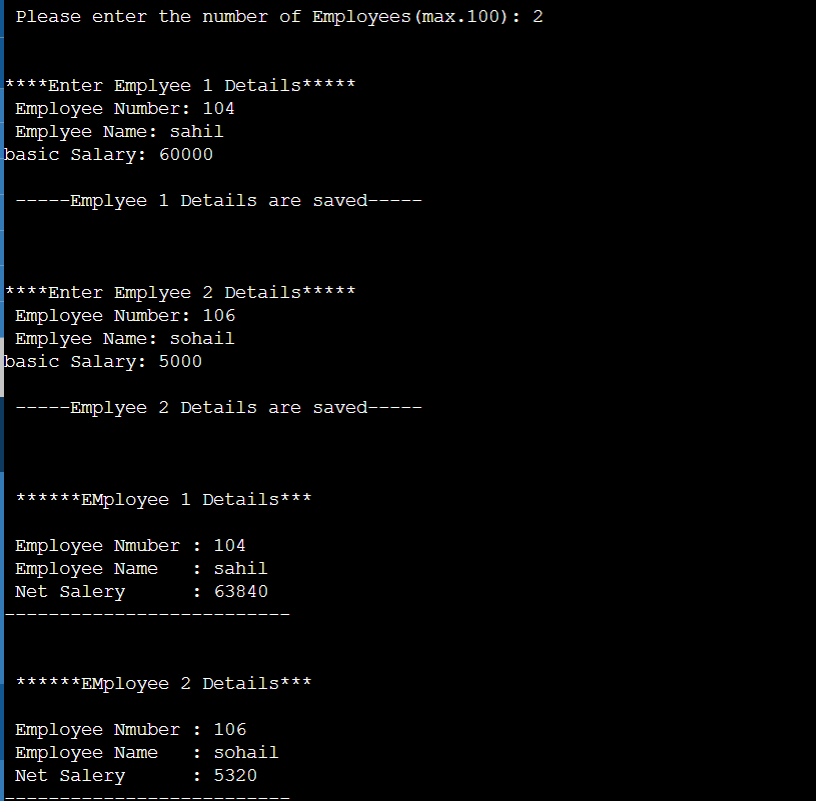
cout<<"\n press any key to close!!!!";

getc;

return 0;

}

**Output:**



**LAB 3**

Q1 Write a Program using class to process Shopping List for a Departmental Store. The list includes details such as the Code No and Price of each itemand perform the operations like Adding, Deleting Items to the list andPrinting the Total value of an Order.

**Program:**

#include<iostream>

using namespace std;

const tm = 50;

class Items

{

int ItemCode[tm];

float ItemPrice[tm];

int count;

public:

void

CNT(void){

count=0;

}

void getitem(void);

void displaySum(void);

void remove(void);

void displayItems(void);

};

void Items::getitem(void)

{

cout<<"Enter item code : ";

cin>>ItemCode[count];

cout<<"Enter item cost; ";

cin>>ItemPrice[count];

count++;

}

void Items::displaySum(void)

{

float sum=0;

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

sum = sum+ItemPrice[i];

cout<<"\n Total value: "<<sum<<endl;

}

}

void Items::remove(void)

{

int a;

cout<<"Enter item code: ";

cin>>a;

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

if(ItemCode[i]==a){

ItemPrice[i]=0;

}

}

void Items::displayItems(void)

{

cout<<"\n Code price: \n";

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

{

cout<<"\n "<<ItemCode[i];

cout<<" "<<ItemPrice[i];

}

cout<<"\n";

}

int main()

{

Items order;

order.CNT();

int x;

do{

cout<<"\n You can do the following: "<<"ENter appropriate number\n";

cout<<"\n1 : Add an items";

cout<<"\n2 : Display total value";

cout<<"\n3 : Delete an item";

cout<<"\n4 : Quit";

cout<<"\n\n What is your option?";

cin>>x;

switch(x)

{

case 1 : order.getitem();

break;

case 2: order.displaySum();

break;

case 3 : order.remove();

break;

case 4 ; order.displayItems();

break;

default:cout<<"Error in input";

}

}wile(x!=5)

return 0;

}

**2. Write a program to maintain the records of person with details (Name and Age) and find the eldest among them. The program must use this pointer to return the result.**

**Program:**

#include<iostream>

#include<string.h>

class person

{

char name[20];

float age;

public:

person(char\*s, float a){

strcpy(name, s);

age = a;

}

person & person :: greater(person & x)

{

if(x.age>=age)

return x;

else

return \*this;

}

void display(void){

cout<<"Name: "<<name<<"\n "<<"Age: "<<age<<endl;

}

};

int main(){

person p1("sahil", 37.50),

p2("rajik", 29.0),

p3("hozaifa", 40.5);

person p = p1.greater(p3);

cout<<"Elder Person is: \n";

p.display();

return 0;

}

**LAB 4**

1**. Write a C++ Program to illustrate default constructor, parameterized constructor and copy constructors.**

**Program:**

#include<iostream>

using namespace std;

class code

{

int id;

int count;

public:

code(){

cout<<"Default constructor called: \n";

id = 0;

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

}

code(int a)

{

cout<<"parameterized constructor called; \n";

id = a;

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

}

code(code &x)

{

cout<<"copy constructor called: \n";

id = x.id;

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

}

void display(){

}

~code(){

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

cout<<"Object Destroyed"<<endl;

}

};

int main(){

code a(10); //calls parameterized constructor

code b(a); //calls copy constructor

code c(a); //calls copy constructor

code d; //calls default constructor

cout<<"\n For object d id= "; d.display();

cout<<"\n For object a id= "; a.display();

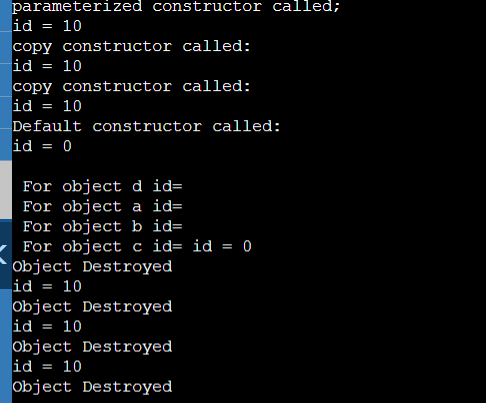
cout<<"\n For object b id= "; d.display();

cout<<"\n For object c id= "; d.display();

return 0;

}

**Output:**

****

**Q2. a)Write a program to demonstrate the i)Operator Overloading**

**ii)Function Overloading.**

**i)Operator Overloading:-**The mechanism of giving a special meaning to an

operator is called operator overloading. This can be achieved by special

function “operator”.

**Program:**

#include<iostream>

using namespace std;

class complex

{

float real,img;

public:

complex();

complex(float x, float y);

void read\_complex();

complex operator +(complex);

complex operator -(complex);

void display();

};

complex :: complex()

{

real=img=0;

}

complex :: complex(float x, float y)

{

real=x;

img=y;

}

void complex :: display()

{

char sign;

if(img<0){

}

else{

sign='-';

img =-img;

sign ='+';

cout<<real<<sign<<" i"<<img<<endl;

}

}

complex complex :: operator + (complex c){

complex r;

r,real = real+c.real;

r.img = img+c.img;

return r;

}

complex complex :: operator -(complex c){

complex r;

r.real = real-c.real;

r.img = img-c.img;

return r;

}

void complex :: read\_complex(){

cout<<"Enter real part of complex number: ";

cin>>real;

cout<<"Enter Imaginary part of complex number: ";

cin>>img;

}

int main(){

complex a;

a.read\_complex();

complex b;

b.read\_complex();

complex c;

cout<<"After Addition of two complex number ";

c = a+b;

c.display()

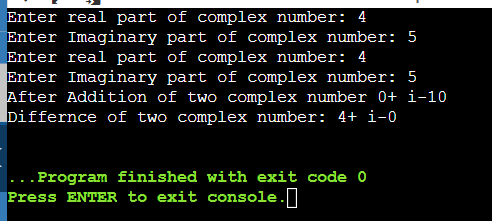
cout<<"Differnce of two complex number: ";

c = a-b;

c.display();

}

**Output:**

****

**LAB 5**

Single Inheritance:-

**Program:**

#include<iostream>

using namespace std;

class worker

{

public:

int age;

char name[20];

void get();

void show();

};

void worker :: get()

{

cout<<"Enter your Name: ";

cin>>name;

cout<<"Enter your age: ";

cin>>age;

}

void worker :: show()

{

cout<<"\nMy name is: "<<name<<endl<<" My age is: "<<age<<endl;

}

class manager : public worker

{

int now;

public:

void get();

void show();

};

void manager :: get()

{

worker :: get();

cout<<"Number of worker under you: ";

cin>>now;

}

void manager :: show()

{

worker :: show();

cout<<"No. of workers under me are: "<<now;

}

int main()

{

worker W1;

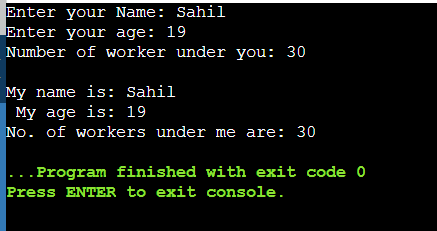
manager M1;

M1.get();

M1.show();

}

**Output:**

****

**MULTI-LEVEL INHERITANCE:-**

**Program:**

#include<iostream>

using namespace std;

class worker

{

int age;

char name[20];

public:

void get();

void show();

};

void worker :: get()

{

cout<<"Enter your name: ";

cin>>name;

cout<<"Enter your age: ";

cin>>age;

}

void worker :: show()

{

cout<<"My name is : "<<name <<"My age is : "<<age<<endl;

}

class manager : public worker

{

int now;

public:

void get();

void show();

};

void manager :: get()

{

worker :: get();

cout<<"No. of worker under you: ";

cin>>now;

}

void manager :: show()

{

worker :: show();

cout<<"NO. of workers under me are: "<<now;

}

class ceo : public manager

{

int num;

public:

void get();

void show();

};

void ceo :: get()

{

manager :: get();

cout<<"NO. of manager under me are: ";

cin>>num;

}

void ceo :: show()

{

cout<<"The number of manager under me are: ";

cout<<num;

}

int main()

{

ceo c;

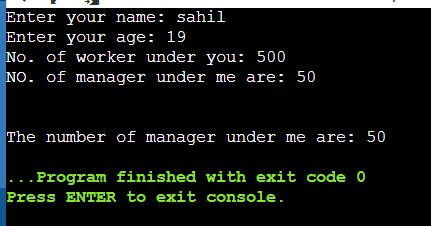
c.get();

cout<<"\n\n";

c.show();

}

Output:



**MULTIPLE INHERITANCE:-**

**Program:**

#include<iostream>

using namespace std;

class father

{

int age;

char name[20];

public:

void get();

void show();

};

void father :: get()

{

cout<<"Enter your father name: ";

cin>>name;

cout<<"Enter the age: ";

cin>>age;

}

void father :: show()

{

cout<<"My father's name is: "<<name<<"\n My father's age is: "<<age<<endl;

}

class mother

{

char name[20];

int age;

public:

void get()

{

cout<<"Enter your mother's name: ";

cin>>name;

cout<<"Enter your mother's age: ";

cin>>age;

}

void show()

{

cout<<"\n My mother's name is : "<<name;

cout<<"\n My mother's age is : "<<age;

}

};

class son : public father, public mother

{

char name[20];

int age;

public:

void get();

void show();

};

void son :: get()

{

father :: get();

mother :: get();

cout<<"child's Name: ";

cin>>name;

cout<<"child's age: ";

cin>>age;

}

void son :: show()

{

father::show();

mother::show();

cout<<"\n My name is: "<<name;

cout<<"\n My age is: "<<age;

}

int main()

{

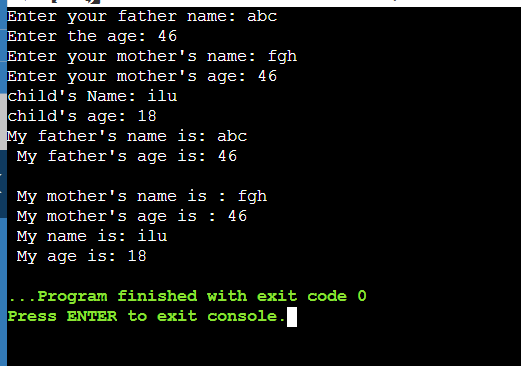
son s1;

s1.get();

s1.show();

}

**Output:**

****