

```
#include<conio.h>
#include<iostream.h>
void main()
{
    clrscr();
    cout<<"dec:\t";
    cout<<dec<<10<<"\t"<<11<<"\t"<<12<<"\n";
    cout<<"oct:\t";
    cout<<oct<<10<<"\t"<<11<<"\t"<<12<<"\n";
    cout<<"hex:\t";
    cout<<hex<<10<<oct<<"\t"<<11<<hex<<"\t"<<12;
    getch();
}
```

/* prog for setw() manipulator*/

```
#include<conio.h>
#include<iostream.h>
#include<iomanip.h>
void main()
{
    int basic=12000,allowance=2000,total;
    total=basic+allowance;
    cout<<setw(10)<<"basic"<<setw(10)<<basic<<endl;
    cout<<setw(10)<<"allowance"<<setw(10)<<allowance<<endl;
    cout<<"_____ "<<endl;
    cout<<setw(10)<<"total"<<setw(10)<<total;
    getch();
}
```

```
#include<stdio.h>
#include<math.h>
#include<conio.h>
#include<iostream.h>
#include<iomanip.h>
void main()
{
    int basic=12000,allowance=2000,total;
    total=basic+allowance;
    clrscr();
    cout<<setw(10)<<"basic"<<setw(10)<<basic<<endl;
    cout<<setw(10)<<"allowance"<<setw(10)<<allowance<<endl;
    cout<<"_____ "<<endl;
    cout<<setw(10)<<"total"<<setw(10)<<total;
    cout<<endl<<setfill('s')<<setw(10)<<"fff";
    cout<<endl<<setprecision(0)<<sqrt(10);
    getch();
}
```

/*prog. For inline function*/

```
#include<stdio.h>
#include<conio.h>
#include<iostream.h>
inline int area(int s)
{
    return (s*s);
}
```

```

void main()
{
    int sides;
    clrscr();
    cout<<"enter the side of square:\t";
    cin>>sides;
    cout<<"\narea of square is :\t"<<area(sides);
    getch();
}

```

/*prog for reference variable */

```

#include<stdio.h>
#include<conio.h>
#include<iostream.h>
void main()
{
    clrscr();
    int sum=10;
    int &total=sum;          //reference variable
    cout<<"value of sum is :"<<sum;
    cout<<"\tvalue of total is:"<<total;
    sum=sum+10;
    cout<<"\nvalue of sum is :"<<sum;
    cout<<"\tvalue of total is:"<<total;
    total=total+10;
    cout<<"\nvalue of sum is :"<<sum;
    cout<<"\tvalue of total is:"<<total;
    getch();
}

```

```
#include<stdio.h>

#include<conio.h>

#include<iostream.h>

class abc{

    int x,y;

    public:

    void getdata()

    {

        this->x=10;

        y=20;

    }

    void putdata()

    {

        cout<<"value of x :-\t"<<this->x;

        cout<<"\nvalue of y :-\t"<<y;

        getch();

    }

};

void main()

{

    abc obj1;

    clrscr();

    obj1.getdata();

    obj1.putdata();

}
```

```
#include<iostream.h>
#include<conio.h>
#include<string.h>
#include<stdlib.h>
class abc{
    int marks;
    char name[20];
public:
    void getdata(int m,char *n)
    {
        strcpy(name,n);
        marks=m;
    }
    void putdata()
    {
        cout<<"\nname :-\t"<<name;
        cout<<"\nmarks :-\t"<<marks;
    }
    abc greater(abc temp)
    {
        if(marks>temp.marks)
            return *this;                //retur invoking object(o1)
        else if(marks<temp.marks)
            return temp;
        else
            // means both are equal
            cout<<"\nboth student has scored equal marks.";
        getch();
        exit(0);
    }
};
```

```
void main()
{
    int m;
    char name[20];
    abc o1,o2,o3;
    cout<<"enter name:-\t";
    cin>>name;
    cout<<"enter "<<name<<"s marks :-\t";
    cin>>m;
    o1.getdata(m,name);
    clrscr();
    cout<<"enter name:-\t";
    cin>>name;
    cout<<"enter "<<name<<"s marks :-\t";
    cin>>m;
    clrscr();
    o2.getdata(m,name);
    cout<<"given data are as:\n\n";
    o1.putdata();
    o2.putdata();
    getch();
    clrscr();
    o3=o1.greater(o2);
    cout<<"student having max. marks is :\n\n";
    o3.putdata();
    getch();
}
```

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<iostream.h>
class student
{
    int rno;
    int marks1,marks2,total;
    char name[10];
public:
    void getinfo()
    {
        cout<<"enter std roll no:\t";
        cin>>rno;
        cout<<"enter std name:\t";
        cin>>name;
        cout<<"enter std english marks:\t";
        cin>>marks1;
        cout<<"enter std math marks:\t";
        cin>>marks2;
        cout<<"\nstudent data added.";
        getch();
        clrscr();
        return;
    }
    void displayinfo()
    {
```

```
total=marks1+marks2;

cout<<rno<<"\t ";
cout<<name<<"\t";
cout<<total<<"\n";

return;

}

};

void main()
{
    clrscr();
    student obj[10];
    int i=0;
    while(1)
    {
        clrscr();
        int choice;

        cout<<"<1> for add std info.\n";
        cout<<"<2> for display std info.\n";
        cout<<"<0> for exit.\n\n";
        cout<<"enter your chioce:\t";
        cin>>choice;

        switch(choice)
        {
            case 1:
                if(i>10)
                {
                    cout<<"\nlist is full.";
                    getch();
                }
            }
        }
    }
}
```



```
}  
else  
{  
    obj[i].getinfo();  
    i++;  
}  
break;
```

case 2:

```
if(i==0)  
{  
    cout<<"no data is added.";  
    getch();  
    break;  
}  
else  
{  
    cout<<"\n\nroll no\t name\t\tmarks\n";  
    for(int j=0;j<i;j++)  
        obj[j].displayinfo();  
    getch();  
    break;  
}
```

case 0: exit(0);

```
}  
  
}  
  
}
```

```
#include<stdio.h>

#include<conio.h>

#include<iostream.h>

class c_1{
    int m1,m2;
    friend void sum(c_1) // a friend is not a member function
    public:
    void getdata()
    {
        cout<<"enter any 2 integer value:\t";
        cin>>m1;
        cin>>m2;
        cout<<"\ndata is added.";
        getch();
        return;
    }
    void display()
    {
        cout<<"value of m1 and m2 is :\t"<<m1<<m2;
        getch();
        return;
    }
};
```

```
void sum(c_1 x)
{
    int s;
    s=x.m1+x.m2;
    cout<<"\nsum of m1 and m2 is :\t"<<s;
    getch();
    return;
}
void main()
{
    c_1 obj1;
    clrscr();
    obj1.getdata();
    clrscr();
    sum(obj1);
}
```

```
class c_2;           // forward declaration
```

```
class c_1{
    int m1;
    friend void sum(c_1,c_2)
    public:
    void getdata()
    {
        cout<<"enter any integer value:\t";
        cin>>m1;
        cout<<"\ndata is added.";
        getch();
        return;
    }
    void display()
    {
        cout<<"value of m1 is :\t"<<m1;
        getch();
        return;
    }
};
```

```
class c_2{
    int m2;
    public:
    friend void sum(c_1,c_2)
    void getdata()
    {
        cout<<"enter any integer value:\t";
        cin>>m2;
```

```
        cout<<"\ndata is added.";
        getch();
        return;
    }
    void display()
    {
        cout<<"value of m2 is :\t"<<m2;
        getch();
        return;
    }
};

void sum(c_1 x,c_2 y)
{
    int s;
    s=x.m1+y.m2;
    cout<<"\nsum of m1 and m2 is :\t"<<s;
    getch();
    return;
}

void main()
{
    c_1 obj1;
    c_2 obj2;
    obj1.getdata();
    clrscr();
    obj2.getdata();
    clrscr();
    sum(obj1,obj2);
}
```

```
class abc{
    int num;
    public:
    void getdata(int n)
    {
        num=n;
    }
    /*the below funciton can't make any change class data if it tries to do so then
    complier shows an error message */
    void putdata_1()const
    {
        // num=num+10 :- its not possible because of constant fun.
        cout<<"\nnum is (const):\t "<<num;
    }
    void putdata_2()
    {
        num=num+10;
        cout<<"\nnum is :\t"<<num;
    }
};

void main()
{
    abc obj;
    clrscr();
    obj.getdata(1);
    obj.putdata_1();
    obj.putdata_2();
    getch();
}
```

```
#include<iostream.h>
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
class figures
{
    public:
    void area(float);
    void area(int,int);
    void area(int);
};
void figures::area(float r)
{
    float pi=3.14,a;
    a=pi*r*r;
    cout<<"area of circle is:\t"<<a;
    getch();
    return;
}
void figures::area(int l,int b)
{
    int a;
    a=l*b;
    cout<<"area of rectngle is:\t"<<a;
    getch();
    return;
}
```

```
void figures::area(int s)
{
    int a;
    a=s*s;
    cout<<"area of square is :\\t"<<a;
    getch();
    return;
}

void main()
{
    while(1)
    {
        clrscr();
        figures obj;
        int choice,s,l,b;
        float r;
        cout<<"<1> for area of circle:\\n";
        cout<<"<2> for area of rectangle:\\n";
        cout<<"<3> for area of square:\\n";
        cout<<"<0> for exit:\\n\\n";
        cout<<"enter your choice:\\t";
        cin>>choice;
        switch(choice)
        {
            case 1: cout<<"enter radius of circle:\\t";
                    cin>>r;
                    obj.area(r);
                    break;
```



```
case 2: cout<<"enter length of rectangle:\t";
        cin>>l;
        cout<<"\nenter breadth of rectangle:\t";
        cin>>b;
        obj.area(l,b);
        break;
case 3: cout<<"enter side of square:\t";
        cin>>s;
        obj.area(s);
        break;
case 0: exit(0);
default:cout<<"wrong input.....!";
        getch();
        break;
}
}
}
```

```
#include<stdio.h>
#include<conio.h>
#include<iostream.h>
class std
{
    int m1,m2;
public:
    std(){                //default constructor
        m1=10;
        m2=30;
    }
    ~std(){};            //destructor
    void display()
    {
        cout<<"\n\nmarks1=\t"<<m1;
        cout<<"\nmarks2=\t"<<m2;
        getch();
        return;
    }
};
void main()
{
    clrscr();
    std obj1;
    obj1.display();
}
```

/*prog for parameterized constructor*/

```
#include<stdio.h>
#include<conio.h>
#include<iostream.h>
class std
{
    int m1,m2;
public:
    std(int x,int y)
    {
        m1=x;
        m2=y;
    }
    ~std(){};           //destructor
    void display()
    {
        cout<<"\n\nmarks1=\t"<<m1;
        cout<<"\nmarks2=\t"<<m2;
        getch();
        return;
    }
};
```

```
void main()
{
    clrscr();
    int m1,m2;
    cout<<"enter marks of student:\t";
    cin>>m1;
    cout<<"\nenter marks of student:\t";
    cin>>m2;
    std obj1(m1,m2);          //para. contructor
    obj1.display();
}
```

```
#include<stdio.h>

#include<conio.h>

#include<iostream.h>

class factorial{

    int num;

    public:

        void getdata()

        {

            cout<<"enter no :\t";

            cin>>num;

        }

        factorial()

        {

        }

        factorial(factorial &o)

        {

            num=o.num;

        }

        void putdata()

        {

            for(int i=num-1;i!=0;i--)

                num=num*i;

            cout<<"factorial is :\t"<<num;

        }

};
```

```
void main()
```

```
{
```

```
clrscr();
```

```
factorial obj1;
```

```
obj1.getdata();
```

```
factorial obj2(obj1);
```

```
obj2.putdata();
```

```
getch();
```

```
}
```

```
#include<stdio.h>

#include<conio.h>

#include<iostream.h>

class u_overloading{

    int a,b,c;

    public:

    void getdata()

    {

        cout<<"\nenter value of a,b and c:\n";

        cin>>a>>b>>c;

        return;

    }

    void putdata()

    {

        cout<<"a="<<a<<" b="<<b<<" c="<<c;

        return;

    }

    void operator -()

    {

        a=-a;

        b=-b;

        c=-c;

        return;

    }

};
```

```
void main()
{
    u_overloading obj;
    clrscr();
    obj.getdata();
    cout<<"value before operator overloading:\n";
    obj.putdata();
    -obj; //activates overloading function
    cout<<"\n\nvalues after operator overlading:\n";
    obj.putdata();
    getch();
}
```



```
#include<stdio.h>
#include<conio.h>
#include<iostream.h>
class overloading{
    int num;
    public:
    void getdata()
    {
        cout<<"\nenter num:\t";
        cin>>num;
    }
    void putdata()
    {
        cout<<"\n"<<num;
    }
    overloading operator-(overloading x)
    {
        overloading temp;
        temp.num=num+x.num;
        return (temp);
    }
};
```

```
void main()
{
    overloading obj1,obj2,obj3;
    clrscr();
    obj1.getdata();
    obj2.getdata();
    obj3=obj1-obj2;           //or obj3=obj1operator+(obj2)
    obj1.putdata();
    obj2.putdata();
    cout<<"\nsum two numbers by using '-' operator:\t";
    obj3.putdata();
    getch();
}
```

```
#include<stdio.h>

#include<conio.h>

#include<iostream.h>

class student{

    int rollno;

    char name[20];

    public:

    void getinfo()

    {

        clrscr();

        cout<<"input roll no:\t";

        cin>>rollno;

        cout<<"\nenter name:\t";

        gets(name);

        clrscr();

        return;

    }

    void putinfo()

    {

        cout<<name<<" roll no. is "<<rollno;

        return;

    }

};

class result:private student{

    int m1,m2,m3;

    int t;

    float per;
```

```
public:
```

```
void getmarks()
```

```
{
```

```
    getinfo();
```

```
    cout<<"\nenter marks of 2 subjects:\n";
```

```
    cin>>m1>>m2;
```

```
    clrscr();
```

```
    return;
```

```
}
```

```
void putmarks()
```

```
{
```

```
    putinfo();
```

```
    t=m1+m2;
```

```
    per=(float)t/200*100;
```

```
    cout<<"\ntotal marks is "<<t<<" and percentage is "<<per<<"%";
```

```
    return;
```

```
}
```

```
};
```

```
void main()
```

```
{
```

```
    result obj;
```

```
    obj.getmarks();
```

```
    obj.putmarks();
```

```
    getch();
```

```
}
```

```
/*prog for single level inheritance with protected data members (publicly) */
```

```
#include<conio.h>
```

```
#include<iostream.h>
```

```
class student{
```

```
    protected:
```

```
    int rollno;
```

```
    char name[20];
```

```
};
```

```
class result:public student{
```

```
    int m1,m2,m3;
```

```
    int t;
```

```
    float per;
```

```
/*here is protected members of base class under visibility label of protected
```

```
    public:
```

```
    void getinfo()
```

```
{
```

```
    clrscr();
```

```
    cout<<"input roll no:\t";
```

```
    cin>>rollno;
```

```
    cout<<"\nenter name:\t";
```

```
    gets(name);
```

```
    clrscr();
```

```
    return;
```

```
}
```

```
void putinfo()
```

```
{  
    cout<<name<<" roll no. is "<<rollno;  
    return;  
}
```

```
void getmarks()
```

```
{  
    cout<<"\nenter marks of 2 subjects:\n";  
    cin>>m1>>m2;  
    clrscr();  
    return;  
}
```

```
void putmarks()
```

```
{  
    t=m1+m2;  
    per=(float)t/200*100;  
    cout<<"\ntotal marks is "<<t<<" and percentage is "<<per<<"%";  
    return;  
}
```

```
};
```

```
void main()
```

```
{  
    result obj;  
    obj.getinfo();  
    obj.getmarks();  
    obj.putinfo();  
    obj.putmarks();  
    getch();  
}
```

```
#include<stdio.h>

#include<conio.h>

#include<iostream.h>

class student{

    int rollno;

    char name[20];

    public:

    void getinfo()

    {

        clrscr();

        cout<<"input roll no:\t";

        cin>>rollno;

        cout<<"\nenter name:\t";

        gets(name);

        clrscr();

        return;

    }

    void putinfo()

    {

        cout<<name<<" roll no. is "<<rollno;

        return;

    }

};

class marks:public student{

    int m1,m2,m3;

    public:

    int t;
```

```
void getmarks()
{
    cout<<"\nenter marks of 2 subjects:\n";
    cin>>m1>>m2;
    t=m1+m2;
    clrscr();
    return;
}

};
```

```
class result:public marks{
    char grade;
    float per;
public:
    void putresult()
    {
        per=(float)t/200*100;
        if(per<=32)
        {
            cout<<"fail";
            return;
        }
        else
        {
            if(per>90)
                grade='A';
            if(per>75)
                grade='B';
```



```
        if(per>55)
            grade='C';
        if(per>32)
            grade='D';
        }
        cout<<"\ngrade is :"<<grade;
        cout<<"\n%age is :"<<per;
        return;
    }
};
```

```
void main()
{
    result obj;
    obj.getinfo();
    obj.getmarks();
    obj.putinfo();
    obj.putresult();
    getch();
}
```

```
#include<stdio.h>
```

```
#include<conio.h>
```

```
#include<iostream.h>
```

```
class student{
```

```
    int rollno;
```

```
    char name[20];
```

```
    public:
```

```
    void getinfo()
```

```
    {
```

```
        clrscr();
```

```
        cout<<"input roll no:\t";
```

```
        cin>>rollno;
```

```
        cout<<"\nenter name:\t";
```

```
        gets(name);
```

```
        clrscr();
```

```
        return;
```

```
    }
```

```
    void putinfo()
```

```
    {
```

```
        cout<<name<<" roll no. is "<<rollno;
```

```
        return;
```

```
    }
```

```
};
```

```
class marks:public student{
```

```
    int m1,m2,m3;
```

```
    public:
```

```
    int t;
```

```
void getmarks()
{
    cout<<"\nenter marks of 2 subjects:\n";
    cin>>m1>>m2;
    t=m1+m2;
    clrscr();
    return;
}

};
```

```
class result:private marks{
    char grade;
    float per;

    // & public function of marks....

public:
    void getfun()
    {
        getinfo();
        getmarks();
    }
    void putfun()
    {
        putinfo();
    }
    void putresult()
    {
        per=(float)t/200*100;
```

```
        if(per<=32)
        {
            cout<<"\nresult is fail.";
            return;
        }
        else
        {
            if(per>90)
                grade='A';
            if(per>75)
                grade='B';
            if(per>55)
                grade='C';
            if(per>32)
                grade='D';
        }
        cout<<"\ngrade is :"<<grade;
        cout<<"\n%age is :"<<per;
        return;
    };

void main()
{
    result obj;
    obj.getfun();
    obj.putfun();
    obj.putresult();
    getch();
}
```

```
/* prog for multi-level inheritance ( privately with return function for addition) */
```

```
#include<stdio.h>
```

```
#include<conio.h>
```

```
#include<iostream.h>
```

```
class one{
```

```
    int x;
```

```
    public:
```

```
    void getdata_x()
```

```
    {
```

```
        clrscr();
```

```
        cout<<"enter value of x : ";
```

```
        cin>>x;
```

```
        clrscr();
```

```
        return;
```

```
    }
```

```
    int getx(){ return(x); }
```

```
    void putdata_x()
```

```
    {
```

```
        cout<<"value of x is :\t"<<x;
```

```
        return;
```

```
    }
```

```
};
```

```
class two:public one{
```

```
    int y;
```

```
    public:
```

```
void getdata_y()
{
    cout<<"\nenter value of y:\t";
    cin>>y;
    clrscr();
    return;
}
int gety(){ return(y); }
void putdata_y()
{
    cout<<"\nvalue of y is : "<<y;
}

};
```

```
class three:private two{
```

```
    int s;
    public:
    void getfun()
    {
        getdata_x();
        getdata_y();
    }
    void putfun()
    {
        putdata_x();
        putdata_y();
    }
}
```

```
void sum()  
{  
    s=getx()+gety();  
    cout<<"\nsum is : "<<s;  
    return;  
}  
};
```

```
void main()  
{  
    three obj;  
    obj.getfun();  
    clrscr();  
    obj.putfun();  
    obj.sum();  
    getch();  
}
```

```
#include<stdio.h>

#include<conio.h>

#include<iostream.h>

class student{

    int rollno;

    char name[20];

    public:

    void getinfo()

    {

        cout<<"input roll no:\t";

        cin>>rollno;

        cout<<"\nenter name:\t";

        gets(name);

        clrscr();

        return;

    }

    void putinfo()

    {

        cout<<name<<" roll no. is "<<rollno;

        return;

    }

};

class marks:public student{

    int m1,m2,m3;

    public:

    int t;
```



```
void getmarks()
{
    cout<<"\nenter marks of 2 subjects:\n";
    cin>>m1>>m2;
    t=m1+m2;
    clrscr();
    return;
}

};
```

```
class result:public marks{
    char grade;
    float per;
public:
    void putresult()
    {
        per=(float)t/200*100;
        if(per<=32)
        {
            cout<<"\nresult is fail.";
            return;
        }
        else
        {
            if(per>90)
                grade='A';
            if(per>75)
                grade='B';
```

```
if(per>55)
    grade='C';
if(per>32)
    grade='D';
}

cout<<"\ngrade is :"<<grade;
cout<<"\n%age is :"<<per;
return;
}

};
```

```
class fee:public student{
    int f1,f2,f3;
public:
    void getfee()
    {
        cout<<"enter fees of 3 subjects.";
        cin>>f1>>f2>>f3;
        return;
    }
    void putfee()
    {
        cout<<"\nfees for 3 subjects areas:\n"<<f1<<"\t"<<f2<<"\t"<<f3;
        return;
    }
};
```

```
void main()
{
    clrscr();
    printf("\nenter records for dept. of result:\n\n");
    result objr;
    objr.getinfo();
    objr.getmarks();
    printf("\nenter records for dept. of admin.:\n\n");
    fee objf;
    objf.getinfo();
    objf.getfee();

    //printing both class data...

    clrscr();
    printf("records of dept. of result are as:\n");
    objr.putinfo();
    objr.putresult();
    printf("\n\n\nrecords of dept. of admin. are as:\n");
    objf.putinfo();
    objf.putfee();
    getch();
}
```

```
#include<conio.h>
#include<iostream.h>
class b{
    public:
    void display()
    {
        cout<<"\n\ndisplay of base class....";
    }
};
class d:public b
{
    public:
    void display()
    {
        b::display();           //calling base class display()
        cout<<"\ndisplay of derived class....\n";
    }
};
void main()
{
    d obj1;
    obj1.display();           //calling its own display() where we call base class display()
    obj1.d::display();        //-----do-----//
    obj1.b::display();        //calling base class display()
    getch();
}
```

```
#include<stdio.h>

#include<conio.h>

#include<iostream.h>

class b1{
    public:
    void display()
    {
        cout<<"\n\ndisplay of base class 1....";
    }
};

class b2{
    public:
    void display()
    {
        cout<<"\n\ndisplay of base class 2.....";
    }
};

class d:public b1,public b2
{
    public:
    void display()
    {
        cout<<"display of derived class....\n";
    }
};
```

```
void main()
{
    d obj1;
    clrscr();
    obj1.display();      //calling its own display()
    obj1.d::display();   //-----do-----//
    obj1.b1::display();  //calling base class 1 display()
    obj1.b2::display();  //calling base class 2 dispaly()
    getch();
}
```

/*prog for calling member function with the help of pointers */ p.no:47

```
#include<iostream.h>
```

```
#include<conio.h>
```

```
class abc{
```

```
    int a,b;
```

```
    public:
```

```
    void getdata(int n,int m)
```

```
    {
```

```
        a=n;
```

```
        b=m;
```

```
    }
```

```
    void putdata()
```

```
    {;
```

```
        cout<<"\nb =\t"<<b;
```

```
        getch();
```

```
    }
```

```
};
```

```
void main()
```

```
{
```

```
    abc obj1,*ptr;
```

```
    ptr=&obj1;
```

```
    cout<<"\na =\t"<<a
```

```
    obj1.getdata(10,20);
```

```
    obj1.putdata();
```

```
    cout<<"\n\nnow calling through arrow operator.\n";
```

```
    ptr->getdata(30,40); //here ptr_getdata->is equal to obj1.getdata
```

```
    ptr->putdata();
```

```
}
```

```
class b{
    public:
    virtual void putdata()
    {
        cout<<"\ndisplay of base class...";
    }
};

class d:public b{
    public:
    void putdata()
    {
        cout<<"\ndisplay of deriver class...";
    }
};

void main()
{
    b obj1;
    d obj2;
    b *ptr1;
    ptr1=&obj1;
    clrscr();
    ptr1->putdata();
    ptr1=&obj2;
    //here complier selects on the bases of content rather than type becaues of virtual fun.
    ptr1->putdata();
    getch();
}
```



```
int s_no=1;
class student{
    int rno;
    char name[20];
    public:
    void getinfo()
    {
        cout<<"\nenter name of student no "<<s_no<<" :\t";
        cin>>name;
        cout<<"\n\nenter "<<name<<"'s roll no:\t";
        cin>>rno;
    }
    void putinfo()
    {
        cout<<"\n"<<s_no<<"\t"<<rno<<"\t"<<name<<"\t";
    }
};

class marks:virtual public student
{
    int m1,m2,m3,total;
    public:
    void getmarks()
    {
        cout<<endl<<"enter marks for m1,m2,m3 subjects:\n";
        cin>>m1>>m2>>m3;
        total=m1+m2+m3;
    }
}
```

```

void putmarks()
{
    cout<<m1<<"\t"<<m2<<"\t"<<m3<<"\t"<<total;
}
};

```

```

class catagory:virtual public student

```

```

{
    char cat[5];
    public:                //contains base class getinfo() & putinfo function
    void getcat()
    {
        cout<<"OP :-for open.\n";
        cout<<"SC :-for schedule cast.\n";
        cout<<"ST :-for schedule tribe.\n";
        cout<<"otr:-for other catagories.\n";
        cout<<"\n\nenter your catagories:\t";
        cin>>cat;
    }
    void putcat()
    {
        cout<<cat<<"\t";
    }
};

```

```

class result:public marks,public catagory

```

```

{
    public:
    void getdetails()
    {
        getinfo();
    }
}

```

```
        clrscr();
        getcat();
        clrscr();
        getmarks();
        clrscr();
    }
    void putresult()
    {
        putinfo();
        putcat();
        putmarks();
    }
};

void main()
{
    result obj[10];
    clrscr();
    cout<<"enter no of student:-\t";
    int num;
    cin>>num;
    for(int i=0;i<num;i++,s_no++)
        obj[i].getdetails();
    clrscr();
    s_no=1;
    cout<<"S.NO\tROLL NO\tNAME\tCAT.\tM1\tM2\tM3\tTOTAL\n\n";
    for(i=0;i<num;i++,s_no++)
        obj[i].putresult();
    getch();
}
```

/* prog for abstract classes / pure virtual function */ p.no:52

```
class b{           //abstract class because it contain atleast one pure virtual
function
    public:
        virtual void putdata()=0;  //pure virtual function or do nothing fun.
};

class d1:public b{
    public:
        void putdata()
        {
            cout<<"\ndisplay of deriver class 1...";
        }
};

class d2:public b{
    public:
        void putdata()
        { cout<<"\ndisplay of derived class 2....";      }
        };

void main()
{
    d1 obj1,*ptr1;  // we can't make an object of class 'b' because it is abstract class
    d2 obj2,*ptr2;
    ptr1=&obj1;
    ptr1->putdata();
    ptr2=&obj2;
    ptr2->putdata();
    getch();
}
```

```
#include<stdio.h>
#include<conio.h>
#include<iostream.h>
class person{
    int idno;
    char name[20];
public:
    void getinfo()
    {
        cout<<"enter name :-\t";
        cin>>name;
        cout<<"enter "<<name<<"s id no:-\t";
        cin>>idno;
    }
    void putinfo()
    {
        cout<<"\n"<<idno<<"\t"<<name<<"\t";
    }
    virtual void getdata()=0;
    virtual int isoutstanding()=0;//pure virtual function or do nothing fun.
};
class student:public person{
    int p_age;
public:
```

```
void getdata()
{
    person::getinfo();
    cout<<"enter student result percentage:-\t";
    cin>>p_age;
    clrscr();
}

int isoutstanding()
{
    return(p_age>85)?1:0;
}

};
```

```
class professor:public person{
    int no_pub;                //no of books published
    public:
    void getdata()
    {
        person::getinfo();
        cout<<"enter no. professor's publication:-\t";
        cin>>no_pub;
        clrscr();
    }

    int isoutstanding()
    {
        return(no_pub>20)?1:0;
    }

};
```

```
void main()
{
    person *obj[100];
    int i=0;
    char choice;
    clrscr();
    do{
        cout<<"enter students or professor (s/p):-\t";
        cin>>choice;
        if(choice=='s')
            obj[i]=new student;
        else
            obj[i]=new professor;
        obj[i++]->getdata();
        cout<<"\ndo you want to continue (y/n):-\t";
        cin>>choice;
    }while(choice=='y');
    clrscr();
    cout<<"\nID NO\tNAME\tPERFORMANEC\n";
    for(int j=0;j<i;j++)
    {
        obj[j]->putinfo();
        if(obj[j]->isoutstanding())
            cout<<"outstanding performance.";
        else
            cout<<"normal performance.";
    }
    getch();
}
```

```
#include<iostream.h>
#include<conio.h>
void main()
{
    char c1,c2;
    clrscr();
    cout<<"\nenter a character:-\t";
    cin.get(c1);           // by cin.get(char *) version / by char version
                           // or we can also use:- c=cin.get(); by void version
    clrscr();
    cout<<"\ngiven character is:-\t"<<c1;
    getch();
}
```

/* prog for creating a files */

```
include<iostream.h>
#include<conio.h>
#include<fstream.h>
void main()
{
    ofstream myfile;
    clrscr();
    myfile.open("newfile.txt");
    myfile<<"this is my first file program in c++";
    myfile.close();
    getch();
}
```



```
#include<iostream.h>
#include<fstream.h>
#include<conio.h>
void main()
{
    char name[20];
    int cost;
    ofstream outf("item.txt");
    cout<<"\n\nenter data for ""item.txt""file:\n\n";
    cout<<"\nenter item name:\t";
    cin>>name;
    outf<<name<<"\n";
    clrscr();
    cout<<"\nenter item price:\t";
    cin>>cost;
    outf<<cost<<"\n";
    outf.close();

    /*****/
    clrscr();
    cout<<"\nreading data from file ""item.txt"" :";
    ifstream inf("item");
    inf>>name;
    inf>>cost;
    cout<<"\n\nitem name:\t"<<name;
    cout<<"\nitem cost:\t"<<cost;
    inf.close();
    getch();
}
```

```
#include<iostream.h>
#include<fstream.h>
#include<conio.h>
#include<stdlib.h>
void main()
{
    clrscr();
    ofstream fout;                                /*inserting data into country.txt file*/
    fout.open("country.txt");
    fout<<"\nindia";
    fout<<"\nusa";
    fout<<"\nnuk";
    fout.close();
    /*****/
    /*inserting data into capital.txt file*/
    fout.open("capital.txt");
    fout<<"\ndelhi";
    fout<<"\n washington";
    fout<<"\n london";
    fout.close();
    /*****/
    /* reading data from file country */
    const int n=80;
    char line[80];

    ifstream fin;
    fin.open("country.txt");
    cout<<"\ncontents of file 'country.txt':\n";
```

while(1)

p.no:59

{

if(fin.eof()!=0)

//if eof occurs then it returns 0 else 1

break;

else

{

fin.getline(line,n);

cout<<line<<"\n";

}

}

fin.close();

*/*******

*/*reading data from file capital */*

fin.open("capital.txt");

cout<<"\ncontents of file 'capital.txt':\n";

while(1)

{

if(fin.eof()!=0)

break;

else

{

fin.getline(line,n);

cout<<line<<"\n";

}

}

fin.close();

getch();

}