```
====================program for inheritance====================
#include<stdio.h>
#include<conio.h>
#include<iostream.h>
class student
{
	int no,marks[3],total;
	char nm[20];
	public:
		void getdata();
		void calculate();
		void display();
};
void student::getdata()
{
	cout<<"\nENTER STUDENT NAME:";
	gets(nm);
	cout<<"\nENTER STUDENT NUMBER:";
	cin>>no;
	for(int i=0;i<3;i++)
	{
		cout<<"\nENTER MARKS["<<i+1<<"]:";
		cin>>marks[i];
	}
}
void student::calculate()
{
	total=0;
	for(int i=0;i<3;i++)
	{
		total=total+marks[i];
	}
}
void student::display()
{
	cout<<"\nNAME:"<<nm;
	cout<<"\nNUMBER:"<<no;
	cout<<"\nTOTAL:"<<total;
}
class parent_data:public student
{
	char fname[20],mname[20];
	public:
		void GetParentData();
		void DisplayParentData();
};
void parent_data::GetParentData()
{
	cout<<"\nENTER FATHER NAME:";
	gets(fname);
	cout<<"\nENTER MOTHER NAME:";
	gets(mname);
}
void parent_data::DisplayParentData()
{
	student::display();
	cout<<"\nFATHER'S NAME:"<<fname;
	cout<<"\nMOTHER'S NAME:"<<mname;
}
int main()
{
	parent_data obj1;
	clrscr();
	obj1.getdata();
	obj1.GetParentData();
	obj1.calculate();
	clrscr();
	obj1.DisplayParentData();
	getch();
	return(0);
};
==================overloading unary minus==================
#include<iostream.h>
#include<stdio.h>
class space
{
	int x,y,z;
	public:
		void getdata()
		{
			cout<<"\nenter value of x:";
			cin>>x;
			cout<<"\nenter value of y:";
			cin>>y;
```

```
                cout<<"\nenter value of z:";
                cin>>z;
        }
        void display()
        {
                cout<<"\nvalue of x:"<<x;
                cout<<"\nvalue of y:"<<y;
                cout<<"\nvalue of z:"<<z;
        }
        void operator-()
        {
                x=-x;
                y=-y;
                z=-z;
        }
};
int main()
{
        space s;
        clrscr();
        s.getdata();
        cout<<"\nvalues of variable before operator overloading\n";
        s.display();
        -s;
        cout<<"\nvalues of variable after operator overloading\n";
        s.display();
        getch();
        return 0;
}
```

==========overloading binary operator==========

```
#include<stdio.h>
#include<iostream.h>
#include<conio.h>
class abc
{
        int x,y;
        public:
                void getdata()
                {
                        cout<<"\nenter value of x:" ;
                        cin>>x;
                        cout<<"\nenter value of y:" ;
                        cin>>y;
                }
                void display()
                {
                        cout<<"\nx="<<x<<" y="<<y;
                }
                abc operator+(abc c)
                {
                        abc temp;
                        temp.x=x+c.x;
                        temp.y=y+c.y;
                        return (temp);
                }
};
/*abc abc::operator+(abc c)
{
        abc temp;
        temp.x=x+c.x;
        temp.y=y+c.y;
        return(temp);
} */
main()
{
        abc obj1,obj2,obj3;
        clrscr();
        obj1.getdata();
        obj2.getdata();
        obj3=obj1+obj2;
        obj3.display();
        getch();
        return 0;
}
```

==========overloading binary operator using friend function==========

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
class abc
{
        int x,y;
        public:
                void getdata()
```

```
        {
                cout<<"\nenter value of x:";
                cin>>x;
                cout<<"\nenter value of y:";
                cin>>y;
        }
        void display()
        {
                cout<<"\nvalue of x:"<<x<<" value of y:"<<y;
        }
        friend abc operator+(abc,abc);
};
abc operator+(abc a,abc b)
{
        abc temp;
        temp.x=a.x+b.x;
        temp.y=a.y+b.y;
        return (temp);
}
int main()
{
        abc obj1,obj2,obj3;
        clrscr();
        obj1.getdata();
        obj2.getdata();
        obj3=operator+(obj1,obj2);
        obj3.display();
        getch();
        return 0;
}
```

==========addition of array and strings using operator overloading=========

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<string.h>
class abc
{
        int arr[5],i,len;
        char *w;
        public:
                void getdata();
                void display();
                friend abc operator+(abc,abc);
};
void abc::getdata()
{
        for(i=0;i<5;i++)
        {
                cout<<"\nenter value of arr["<<i+1<<"]:";
                cin>>arr[i];
        }
        cout<<"\nenter any word:";
        cin>>w;
        len=strlen(w);
}
void abc::display()
{
        for(i=0;i<5;i++)
        {
                cout<<arr[i]<<" ";
        }
        cout<<endl<<w;
}
abc operator+(abc a,abc b)
{
        abc temp;
        int i;
        for(i=0;i<5;i++)
        {
                temp.arr[i]=a.arr[i]+b.arr[i];
        }
        temp.len=a.len+b.len;
        temp.w=new char[temp.len+1];
        strcpy(temp.w,b.w);
        strcat(temp.w,a.w);
        return(temp);
}
int main()
{
        abc obj1,obj2,obj3;
        clrscr();
        cout<<"\nenter value of obj1 array\n";
        obj1.getdata();
```

```
	cout<<"\nenter value of obj2 array\n";
	obj2.getdata();
	cout<<"\naddition of arrays\n\n";
	obj3=obj1+obj2;
	obj3.display();
	getch();
	return 0;
}
=======================program to write in file=====================
#include<fstream.h>
#include<conio.h>
#include<stdio.h>
int main()
{
	ofstream fout;
	fout.open("out.txt");
	char str[50]="this is my first program of file handling";
	fout<<str;
	fout.close();
	return(0);
}
==============program to read from txt file and display==============
#include<fstream.h>
#include<conio.h>
int main()
{
	ifstream fin;
	fin.open("out.txt");
	char ch;
	while(!fin.eof())
	{
		fin.get(ch);
		cout<<ch;
	}
	fin.close();
	getch();
	return(0);
}
===============program to write from keyboard in file==============
#include<fstream.h>
#include<conio.h>
#include<stdio.h>
int main()
{
	ofstream fout;   ////obj to write data in file
	ifstream fin;  ////obj to read from file

	char str[50],ch;
	clrscr();
	fout.open("out.txt");
	cout<<"\nenter text to write in file:\n";
	gets(str);
	fout<<str;
	fout.close();
	cout<<"\nDATA WRITEN IN FILE\n";
	fin.open("out.txt");
	while(!fin.eof())
	{
		fin.get(ch);
		cout<<ch;
	}
	fin.close();
	getch();
	return(0);
};
======================using fstream object======================
#include<fstream.h>
#include<conio.h>
#include<stdio.h>
int main()
{
	fstream obj;
	char str[50],ch;
	int count=0;
	obj.open("test.txt",ios::in|ios::out|ios::app);
	cout<<"\nenter text to write in file\n";
	gets(str);
	obj<<str;
	cout<<"\nDATA WRITTEN IN FILE";
	obj.seekp(ios::beg);
	while(!obj.eof())
	{
		obj.get(ch);
		count++;
```

```
        }
        cout<<"\n\ntotal letter file:"<<count;
        getch();
        return(0);
};
=================parameterezed constructor=====================
#include<stdio.h>
#include<conio.h>
#include<iostream.h>
class abc
{
        int x,y,z;
        public:
        abc()
        {
                x=1;y=2;z=3;
        }
        abc(int a)
        {
                x=a*10;y=a*20;z=a*30;
        }
        void getdata()
        {
                cout<<"\nx="<<x;
                cout<<"\ny="<<y;
                cout<<"\nz="<<z;
        }
};
int main()
{
        abc obj1;
        clrscr();
        obj1.getdata();
        abc obj2(3);
        obj2.getdata();
        getch();
        return(0);
}
====================copy constructor====================
#include<iostream.h>
#include<conio.h>
class abc
{
        int a;
        public:
                abc(int temp)
                {
                        a=temp;
                }
                abc(abc &c1)
                {
                        a=c1.a;
                }
                void print()
                {
                        cout<<"\nvalue of a:"<<a;
                }
};
int main()
{
        abc a1(10);
        abc a2(a1); //the statement meansf same as below
        abc a3=a1;
        clrscr();
        a1.print();
        a2.print();
        a3.print();
        getch();
        return(0);
}
====================use of virtual function=====================
#include<stdio.h>
#include<conio.h>
#include<iostream.h>
class A
{
        public:
                virtual void display1()
                {
                        cout<<"\nDISPLAY1 A";
                }
                virtual void display2()
                {
                        cout<<"\nDISPLAY2 A";
```

```
                }
};
class B:public A
{
        public:
                void display1()
                {
                        cout<<"\nDISPLAY1 B";
                }
                void display2()
                {
                        cout<<"\nDISPLAY2 B";
                }
};
class C:public B
{
        public:
                void display1()
                {
                        cout<<"\nDISPLAY1 C";
                }
};
int main()
{
        A a1;
        B b1;
        C c1;

        clrscr();

        A *ptr;

        ptr=&a1;
        ptr->display1();
        ptr->display2();

        ptr=&b1;
        ptr->display1();
        ptr->display2();


        ptr=&c1;
        ptr->display1();

        getch();
        return(0);
}
```