1. **WRITE A PROGRAM THAT REVERSE A NUMBER USING REV() FUNCTION.**

Solution:-

```
//reverse a number using function
#include<iostream.h>
#include<conio.h>
int rev(int n)
{
        int i,r;
        for(i=n,r=0;i>0;r=r*10+i%10,i=i/10);
        return r;
}
void main()
{
        clrscr();
        int n,r;
        cout<<"\nenter a number:: ";
        cin>>n;
        r=rev(n);
        cout<<"\nthe reverse of "<<n<<" is "<<r;
        getch();
}
```

Output:-

```
enter a number:: 123
the reverse of 123 is 321
```

2. **WRITE A PROGRAM THAT CHECK A NUMBER IS ARMSTRONG OR NOT.**

Solution:-

```
//check whether a number is armstrong or not
#include<iostream.h>
#include<conio.h>
#include<math.h>
int armstrong(int n)
{
        int i,c,sum;
        for(i=n,c=0;i>0;i=i/10,c++);
        for(i=n,sum=0;i>0;sum=sum+pow(i%10,c),i=i/10);
        if(sum==n)
                return 1;
        else
```

```
                return 0;
}
void main()
{
        clrscr();
        int n,a;
        cout<<"\nenter a number:: ";
        cin>>n;
        a=armstrong(n);
        if(a==1)
                cout<<"\nthe number is armstrong";
        else
                cout<<"\nthe number is not armstrong";
        getch();
}
```

Output:-

```
enter a number:: 1634
the number is armstrong
```

3. <u>WRITE A PROGRAM THAT SWAP TWO NUMBERS USING CALL BY VALUE.</u>

Solution:-

```
        //swapping using call by value
#include<iostream.h>
#include<conio.h>
void swap(int p,int q)
{
        p=p+q;
        q=p-q;
        p=p-q;
        cout<<"\nnow the value of A is "<<p<<" and the value of B is "<<q;
}
void main()
{
        clrscr();
        int a,b;
        cout<<"\nenter the value of A and B:: ";
        cin>>a>>b;
        swap(a,b);
        getch();
}
```

Output:-
enter the value of A and B:: 50 60
now the value of A is 60 and the value of B is 50

4. WRITE A PROGRAM TO SWAP TWO NUMBERS USING CALL BY REFERENCE

Solution:-

```
        //swapping using call by value
#include<iostream.h>
#include<conio.h>
void swap(int &p,int &q)
{
        p=p+q;
        q=p-q;
        p=p-q;
        cout<<"\nnow the value of A is "<<p<<" and the value of B is "<<q;
}
void main()
{
        clrscr();
        int a,b;
        cout<<"\nenter the value of A and B:: ";
        cin>>a>>b;
        swap(a,b);
        getch();
}
```

Output:-

enter the value of A and B:: 10 20
now the value of A is 20 and the value of B is 10

5. WRITE PROGRAM TO DEMONSTRATE INLINE FUNCTION'S CONCEPT.

Solution:-

```
/*inline function*/
#include<iostream.h>
#include<conio.h>
inline int add(int a,int b)
{
        return a+b;
}
void main()
{
```

```
        clrscr();
        int x,y,z;
        cout<<"\nenter the two values:: ";
        cin>>x>>y;
        z=add(x,y);
        cout<<"Sum of "<<x<<" and "<<y<<" is "<<z;
        getch();
}
```

Output:-

```
        enter the two values:: 20 30
        sum of 20 and 30 is 50
```

6. WRITE A PROGRAM TO DEMONSTRATE DEFAULT ARGUMENT'S CONCEPT.

Solution:-

```
        //default argument
#include<iostream.h>
#include<conio.h>
int sum(int a=10,int b=40)
{
        return a+b;
}
void main()
{
        clrscr();
        int a,b,r;
        cout<<"\nenter the value of a and b:: ";
        cin>>a>>b;
        r=sum(a);
        cout<<"\nthe sum is :: "<<r;
        r=sum(a,b);
        cout<<"\nthe sum is :: "<<r;
        r=sum();
        cout<<"\nthe sum is :: "<<r;
        getch();
}
```

Output:_

```
        enter the value of a and b:: 80 20
        the sum is:: 120
        the sum is :: 100
        the sum is :: 50
```

7. WRITE AN PROGRAM THAT DEMONSTRATE THE CLASS AND OBJECT CONCEPT.

Solution:-

```
        //concept of class and object
#include<iostream.h>
#include<conio.h>
class add//to define a class
{
        private: //access specifiers
                int a,b;//member data
        public: //access specifiers
                void input()//member function
                {
                        cout<<"\nenter two values:: ";
                        cin>>a>>b;
                }
                void display()//member function
                {
                        cout<<"\nthe sum is :: "<<a+b;
                }
};
void main()
{
        clrscr();
        add obj;//create an object obj of add class
        obj.input();
        obj.display();
        getch();
}
```

Output:-

```
        enter two values:: 3 2
        the sum is:: 5
```

8. WRITE A PROGRAM THAT DEMONSTRATE DATA ABSTRACTION.

Solution:-

```
//concept of data abstraction
#include<iostream.h>
```

```
#include<conio.h>

class add//to define a class
{
	private: //access specifiers
		int a,b;//member data
	public: //access specifiers
		void input()//member function
		{
			cout<<"\nenter two values:: ";
			cin>>a>>b;
		}
		void display()//member function
		{
			cout<<"\nthe sum is :: "<<a+b;
		}
};
void main()
{
	clrscr();
	add obj;//create an object obj of add class
	obj.input();
	obj.display();
	getch();
}
```

Output:-

```
	enter two values:: 3 2
	the sum is:: 5
```

9. WRITE A PROGRAM THAT DEMONSTRATE DATA ENCAPSULATION

Solution:-

```
//concept of data encapsulation
#include<iostream.h>
#include<conio.h>
class add//to define a class
{
	private: //access specifiers
		int a,b;//member data
	public: //access specifiers
		void input()//member function
		{
```

```
                cout<<"\nenter two values:: ";
                cin>>a>>b;
        }
        void display()//member function
        {
                cout<<"\nthe sum is :: "<<a+b;
        }
};
void main()
{
        clrscr();
        add obj;//create an object obj of add class
        obj.input();
        obj.display();
        getch();
}
Output:-
        enter two values:: 3 2
        the sum is:: 5
```

10. WRITE A PROGRAM THAT ACCESS DATA MEMBERS AND MEMBER FUNCTIONS.

```
Solution:-
        //access data members and member fnction
#include<iostream.h>
#include<conio.h>
class inc
{
        private:
                int x;
        public:
                void input()
                {
                        cout<<"\nenter the value of x:: ";
                        cin>>x;
                }
                void display()
                {
                        cout<<"\nafter increment:: "<<++x;
                }
};
void main()
{
```

```
        clrscr();
        inc obj;
        obj.input();
        obj.display();
        getch();
}
```

Output:-

```
enter the value of x:: 5
after increment:: 6
```

11. <u>WRITE A PROGRAM THAT DEMONSTRATE NESTING OF MEMBER FUNCTION.</u>

Solution:-

```
        //nesting of member function
#include<iostream.h>
#include<conio.h>
class rev
{
        private:
                int n,r;
        public:
                void input()
                {
                        cout<<"\nenter the value of n:: ";
                        cin>>n;
                        r=cal(n);
                        cout<<"\nreverse is:: "<<r;
                }
                int cal(int n)
                {
                        int i,r;
                        for(i=n,r=0;i>0;r=r*10+i%10,i=i/10);
                        return r;
                }
};
void main()
{
        clrscr();
        rev r;
        r.input();
        getch();
}
```

Output:-
enter the value of n:: 123
reverse is 321

12. WRITE A PROGRAM THAT DEMONSTRATE FUNCTION OVERLOADING

Solution:-

```
        //function overloading
#include<iostream.h>
#include<conio.h>

int add(int a,int b)
{
        return a+b;
}
float add(float a,float b)
{
        return a+b;
}
void main()
{
        clrscr();
        int a,b;
        float a1,b1;
        cout<<"\nenter two values in int:: ";
        cin>>a>>b;
        cout<<"The sum of two int val is :: "<<add(a,b);
        cout<<"\nenter two values in float:: ";
        cin>>a1>>b1;
        cout<<"The sum of two float val is :: "<<add(a1,b1);
        getch();
}
```

Output:-
enter two values in int:: 12 23
The sum of two int val is:: 35
enter two values in float:: 1.2 5.6
The sum of two float val is:: 6.8

13. WRITE A PROGRAM THAT COUNTS THE NUMBER OF OBJECT IN A PROGRAM BY USING STATIC DATA MEMBER AND STATIC MEMBER FUNCTION.

Solution:-

```
        //count object
#include<iostream.h>
```

```
#include<conio.h>
class count
{
	public:
		static int c;
		void show(int n)
		{
			cout<<"\nthe value is :: "<<n<<endl;
			c++;
		}
		static void display()
		{
			cout<<"\nwe call object "<<c<<" times";
		}
};
int count::c=0;
void main()
{
	clrscr();
	count obj1,obj2;
	obj1.show(15);
	obj2.show(60);
	count::display();
	getch();
}
```

Output:-

```
the value is :: 15
the value is :: 60
we call object 2 times
```

14. WRITE A PROGRAM THAT DEMONSTRATE FRIEND FUNCTION'S CONCEPT

Solution:-

```
	//friendly function
#include<iostream.h>
#include<conio.h>
class B;
class A
{
	private:
		int x;
	public:
		void input()
```

```
        {
            cout<<"\nenter a number:: ";
            cin>>x;
        }
        friend int add(A a,B b);
};
class B
{
    private:
        int y;
    public:
        void input()
        {
            cout<<"\nenter another number:: ";
            cin>>y;
        }
        friend int add(A a,B b);
};
int add(A a,B b)
{
    return a.x+b.y;
}
void main()
{
    clrscr();
    A a;
    B b;
    a.input();
    b.input();
    cout<<"The sum of two numbers is :: "<<add(a,b);
    getch();
}
```

Output:-

```
enter a number:: 5
enter another number:: 5
The sum of two numbers is :: 10
```

15. <u>WRITE A PROGRAM THAT DEMONSTRATE DEFAULT CONSTRUCTOR'S CONCEPT</u>

Solution:-

```
    //default constructor
#include<iostream.h>
```

```
#include<conio.h>

class construct
{
	public:
		int x,y;
		construct()
		{
			x=40;
			y=30;
		}
		void display()
		{
			cout<<"The sum is:: "<<x+y;
		}
};
void main()
{
	clrscr();
	construct c;
	c.display();
	getch();
}
```

Output:-
The sum is:: 70

16. <u>WRITE A PROGRAM THAT DEMONSTRATE PARAMETERIZED CONSTRUCTOR'S CONCEPT</u>

Solution:-

```
	//parameterized constructor
#include<iostream.h>
#include<conio.h>
class abhi
{
	public:
		int a,b;
		abhi(int x,int y)
		{
			a=x;
			b=y;
		}
		void display()
```

```
        {
            cout<<"\nThe sum is :: "<<a+b;
        }
};
void main()
{
    clrscr();
    abhi obj(2,4);
    obj.display();
    getch();
}
```

Output:-

The sum is:: 6

17. WRITE A PROGRAM THAT DEMONSTRATE COPY CONSTRUCTOR'S CONCEPT

Solution:-

```
    #include<iostream.h>
#include<conio.h>
class add
{
    private:
        int x,y,z;
    public:
        add()
        {
        }
        add(int a,int b)
        {
            x=a;
            y=b;
        }
        add(add &p)
        {
            x=p.x;
            y=p.y;
        }
        int cal()
        {
            z=x+y;
            return z;
        }
        void display()
```

```
        {
                int x1;
                x1=cal();
                cout<<"Result is "<<x1;
        }
};
void main()
{
        clrscr();
        add p(5,6);
        add q(p);
        q.display();
        getch();
}
```

Output:-
Result is 11

18. <u>WRITE A PROGRAM THAT DEMONSTRATE OBJECT AS FUNCTION ARGUMENT CONCEPT</u>

Solution:-

```
        //object as function argument
#include<iostream.h>
#include<conio.h>

class add
{
        public:
                int a,b;
                void input()
                {
                        cout<<"\nenter two numbers:: ";
                        cin>>a>>b;
                }
                void cal(add q)
                {
                        a=a+q.a;
                        b=b+q.b;
                }
                void display()
                {
                        cout<<"The sum is :: "<<a<<" and "<<b;
                }
```

```
};
void main()
{
        add a1,a2;
        a1.input();
        a2.input();
        a1.cal(a2);
        a1.display();
        getch();
}
```

Output:-

```
enter two numbers:: 5  7
enter two numbers::  8   9
The sum is:: 13   16
```

19. <u>WRITE A PROGRAM THAT DEMONSTRATE POINTER TO OBJECT CONCEPT</u>

Solution:-

```
        //pointer object
#include<iostream.h>
#include<conio.h>
class A
{
        int x,y;
        public:
                void input()
                {
                        cout<<"\nenter any two numbers:: ";
                        cin>>x>>y;
                }
                void display()
                {
                        cout<<"\nthe sum of two numbers is "<<x+y;
                }
};
void main()
{
        A *a;
        a->input();
        a->display();
        getch();
}
```

Output:-
enter two any number:: 5 8
sum of two numbers is :: 13

20. WRITE A PROGRAM THAT DEMONSTRATE UNARY OPERATOR OVERLOADING CONCEPT

Solution:-

```
        //unary operator overloading
#include<iostream.h>
#include<conio.h>
class increment
{
        private:
                int x;
        public:
                void input()
                {
                        cout<<"\nenter a number:: ";
                        cin>>x;
                }
                void operator ++()
                {
                        ++x;
                }
                void display()
                {
                        cout<<"\nthe result is :: "<<x;
                }
};
void main()
{
        clrscr();
        increment i;
        i.input();
        ++i;
        i.display();
        getch();
}
```

Output:-
enter a number:: 9
the result is :: 10

21. WRITE A PROGRAM THAT DEMONSTRATE BINARY OPERATOR OVERLOADING CONCEPT

Solution:-

```
//binary operator overloading
#include<iostream.h>
#include<conio.h>
class mult
{
    private:
        int f,s;
    public:
        mult()
        {
            f=s=0;
        }
        mult(int x,int y)
        {
            f=x;
            s=y;
        }
        void input()
        {
            cout<<"\nenter two values :: ";
            cin>>f>>s;
        }
        void display()
        {
            cout<<"\nresult is :: "<<f<<" and "<<s;
        }
        mult operator *(mult m)
        {
            int a,b;
            a=f*m.f;
            b=s*m.s;
            return mult(a,b);
        }
};
void main()
{
    clrscr();
    mult m1,m2,m3;
    m1.input();
```

```
        m2.input();
        m3=m1*m2;
        m3.display();
        getch();
}
```

Output:-

```
enter two values:: 5    7
enter two values:: 9   8
result is :: 45    56
```

22. <u>WRITE A PROGRAM THAT CONCATENATE TWO STRINGS</u>

Solution:-

```
        //concatenate two strings
#include<iostream.h>
#include<conio.h>
#include<string.h>
class string
{
        private:
                char str[50];
        public:
                string()
                {
                        strcpy(str,"");
                }
                string(char *ch)
                {
                        strcpy(str,ch);
                }
                void input()
                {
                        cout<<"\nenter the string:: ";
                        cin>>str;
                }
                void display()
                {
                        cout<<str;
                }
                string operator +(string q)
                {
                        char abhi[50];
```

```
                strcpy(abhi,str);
                strcat(abhi,q.str);
                return string(abhi);
        }
};
void main()
{
        clrscr();
        string s1,s2,s3;
        s1.input();
        s2.input();
        s3=s1+s2;
        cout<<"\nafter concatenation string:: ";
        s3.display();
        getch();
}
```

Output:-

```
enter the string:: Good
enter the string:: Morning
after concatenation string:: GoodMorning
```

23. <u>WRITE A PROGRAM THAT CHECK WHETHER BOTH STRINGS ARE COMMON OR NOT TO OVERLOAD == OPERATOR</u>

Solution:-

```
#include<iostream.h>
#include<conio.h>
#include<string.h>
class string
{
        public:
                char str[60];
                string()//non-parametrized constructor
                {
                        strcpy(str,"");
                }
                string(char *ch)//parameterizd constructor
                {
                        strcpy(str,ch);
                }
                void input()
                {
```

```
                cout<<"\nenter the string:: ";
                cin>>str;
        }
        int operator ==(string q)//compare
        {
                int i;
                i=strcmp(str,q.str);
                return i;
        }
};
void main()
{
        clrscr();
        int i;
        string s1,s2,s3;
        s1.input();
        s2.input();
        i=s1==s2;
        if(i==0)
                cout<<"\nboth strings are equal";
        else
                cout<<"\nboth strings are not equal";
        getch();
}
```

Output:-

```
enter the string:: RICIS
enter the string:: RICIS
both strings are equal
```

24. <u>WRITE A PROGRAM THAT DEMONSTRATE DESTRUCTOR</u>

Solution:-

```
//destructor
#include<iostream.h>
#include<conio.h>
class des
{
        public:
                int a;
                des()
                {
                        a=5;
```

```
                cout<<"\nvalue in constructor:: "<<++a;
        }
        ~des()
        {
                cout<<"\nvalue in destructor is :: "<<--a;
        }
};
void main()
{
        des d1,d2;
        getch();
}
```

Output:-

```
value in constructor:: 6
value in constructor:: 6
value in destructor:: 5
value in destructor:: 5
```

25. WRITE A PROGRAM THAT DEMONSTRATE CLASS TO BASIC TYPE CONCEPT

Solution:-

```
        //class to basic type
#include<iostream.h>
#include<conio.h>

class add
{
        public:
                int a,b;
                void input()
                {
                        cout<<"\nenter two values:: ";
                        cin>>a>>b;
                }
                operator int()
                {
                        return a+b;
                }
};
void main()
{
```

```
        clrscr();
        add a;
        int b;
        a.input();
        b=a;
        cout<<"\nthe result is :: "<<b;
        getch();
}
```

Output:-

```
enter two values:: 5    7
the result is :: 12
```

26. <u>WRITE A PROGRAM THAT PERFORM COMPLEX NUMBER OPERATION TO OVERLOAD +,- AND * OPERATOR</u>

Solution:-

```
        //complex number add,sub and mult
#include<iostream.h>
#include<conio.h>
class complex
{
        private:
                int rel,img;
        public:
                complex()
                {
                        rel=img=0;
                }
                complex(int a,int b)
                {
                        rel=a;
                        img=b;
                }
                void input_1()
                {
                        cout<<"\nenter 1st real and imaginary part :: ";
                        cin>>rel>>img;
                }
                void input_2()
                {
                        cout<<"\nenter 2nd real and imaginary part:: ";
                        cin>>rel>>img;
```

```
        }
        void display()
        {
                cout<<"\n("<<rel<<")"<<"+"<<"("<<img<<")"<<"i";
        }
        complex operator +(complex c)
        {
                int p,q;
                p=rel+c.rel;
                q=img+c.img;
                return complex(p,q);
        }
        complex operator -(complex c)
        {
                int p,q;
                p=rel-c.rel;
                q=img-c.img;
                return complex(p,q);
        }
        complex operator *(complex c)
        {
                int p,q;
                p=(rel*c.rel)-(img*c.img);
                q=(rel*c.img)+(img*c.rel);
                return complex (p,q);
        }
};
void main()
{
        clrscr();
        complex c1,c2,c3;
        c1.input_1();
        c2.input_2();
        cout<<"\naddition of two complex number is"<<endl;
        c3=c1+c2;
        c3.display();
        cout<<"\nsubtraction of two complex number is"<<endl;
        c3=c1-c2;
        c3.display();
        cout<<"\nmultiplication of two complex number is"<<endl;
        c3=c1*c2;
        c3.display();
        getch();
```

```
}
```

Output:-

enter 1st real and imaginary part:: 1 2
enter 2nd real and imaginary part:: 3 4
addition of two complex number is
(4)+(6)i
Subtraction of two complex number is
(-2)+(-2)i
Multiplication of two complex number is
(-5)+(10)i

27. WRITE A PROGRAM TO DEMONSTRATE SINGLE INHERITANCE

Solution:-

```
        //single inheritance
#include<iostream.h>
#include<conio.h>
class base
{
        protected:
                int a,b;
        public:
                void input()
                {
                        cout<<"\nenter the values:: ";
                        cin>>a>>b;
                }
};
class derive:public base
{
        public:
                void display()
                {
                        cout<<"\nresult is:: "<<a+b;
                }
};
void main()
{
        clrscr();
        derive obj;
        obj.input();
        obj.display();
```

```
        getch();
}
```

Output:-

```
enter the values:: 5    6
result is:: 11
```

28. <u>WRITE A PROGRAM TO DEMONSTRATE MULTIPLE INHERITANCE</u>

Solution:-

```
        /*multiple inheritance*/
#include<iostream.h>
#include<conio.h>

class A
{
        protected:
                int a;
};
class B
{
        protected:
                int b;
};
class C
{
        protected:
                int c;
};
class D:public A,public B,public C
{
        public:
        void input()
        {
                cout<<"\nenter three values:: ";
                cin>>a>>b>>c;
        }
        void display()
        {
                cout<<"A is :: "<<a<<endl
                   <<"B  is :: "<<b<<endl
                   <<"C  is :: "<<c;
        }
```

```
};
void main()
{
        D obd;
        obd.input();
        obd.display();
        getch();
}
Output:-
enter three values:: 5   6   7
A is :: 5
B is :: 6
C is :: 7
```

29. WRITE A PROGRAM TO DEMONSTRATE HIERARCHICAL INHERITANCE

```
Solution:-
        /*hierarchical inheritance*/
#include<iostream.h>
#include<conio.h>

class Base
{
        protected:
                int mult;
};
class Derive_1:public Base
{
        public:
                void cal()
                {
                        int i;
                        for(i=1,mult=1;i<=5;i++)
                        {
                                mult=mult*i;
                        }
                }
                void display()
                {
                        cout<<"\nmultiplication of first 5 number is :: "<<mult;
                }
};
class Derive_2:public Base
```

```
{
	public:
	void cal()
	{
		int i;
		for(i=1,mult=1;i<=6;i++)
		{
			mult=mult*i;
		}
	}
	void display()
	{
		cout<<"\nmultiplication of first 6 number is :: "<<mult;
	}
};
class Derive_3:public Base
{
	public:
	void cal()
	{
		int i;
		for(i=1,mult=1;i<=7;i++)
		{
			mult=mult*i;
		}
	}
	void display()
	{
		cout<<"\nmultiplication of first 7 number is :: "<<mult;
	}
};
void main()
{
	clrscr();
	Derive_1 obd1;
	Derive_2 obd2;
	Derive_3 obd3;
	obd1.cal();
	obd1.display();
	obd2.cal();
	obd2.display();
	obd3.cal();
	obd3.display();
```

```
        getch();
}
```

Output:-
multiplication of first 5 number is:: 120
multiplication of first 6 number is:: 720
multiplication of first 7 number is:: 5040

30. WRITE A PROGRAM TO DEMONSTRATE MULTILEVEL INHERITANCE

Solution:-

```
/*multilevel inheritance*/
#include<iostream.h>
#include<conio.h>
#include<string.h>
class A
{
        protected:
                char name[50];
        public:
                void getdata()
                {
                        cout<<"\nenter your name here:: ";
                        cin>>name;
                }
                void display()
                {
                        cout<<"Name of Student is :: "<<name;
                }
};
class B:public A
{
        protected:
                int roll;
        public:
                void getdata()
                {
                        A::getdata();
                        cout<<"\nenter your roll number here:: ";
                        cin>>roll;
                }
                void display()
                {
```

```
                A::display();
                cout<<"\nRoll of the student is :: "<<roll;
        }
};
class C:public B
{
    protected:
        int marks;
    public:
        void getdata ()
        {
            B::getdata();
            cout<<"\nenter your marks here:: ";
            cin>>marks;
        }
        void display()
        {
            B::display();
            cout<<"\nStudents's Marks is :: "<<marks;
        }
};
void main()
{
    clrscr();
    C obc;
    obc.getdata();
    obc.display();
    getch();
}
```

Output:-

```
enter your name here:: MS Dhoni
enter your roll number here:: 07
enter your marks here:: 183
Name of student is:: MS Dhoni
Roll of the student is:: 07
Student's marks is:: 183
```

31. <u>WRITE A PROGRAM THAT DEMONSTRATE HYBRID INHERITANCE</u>

Solution:-

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

```
class arithmetic
{
protected:
int num1, num2;
public:
void getdata()
{
cout<<"For Addition:";
cout<<"\nEnter the first number: ";
cin>>num1;
cout<<"\nEnter the second number: ";
cin>>num2;
}
};
class plus:public arithmetic
{
protected:
int sum;
public:
void add()
{
sum=num1+num2;
}
};
class minus
{
protected:
int n1,n2,diff;
public:
void sub()
{
cout<<"\nFor Subtraction:";
cout<<"\nEnter the first number: ";
cin>>n1;
cout<<"\nEnter the second number: ";
cin>>n2;
diff=n1-n2;
}
};
class result:public plus, public minus
{
public:
void display()
```

```
{
cout<<"\nSum of "<<num1<<" and "<<num2<<"= "<<sum;
cout<<"\nDifference of "<<n1<<" and "<<n2<<"= "<<diff;
}
};
void main()
{
clrscr();
result z;
z.getdata();
z.add();
z.sub();
z.display();
getch();
}
```

Output:-

```
For Addition:
Enter the first number:   5
Enter the second number:  10
For Subtraction:
Enter the first number:   20
Enter the second number:   10
Sum of 5 and 10=15
Subtraction of 20 and 10= 10
```

32. <u>WRITE A PROGRAM TO DEMONSTRATE VIRTUAL BAS CLASS</u>

Solution:-

```
        //virtual base class
#include<iostream.h>
#include<conio.h>

class A
{
        public:
                int a;
};
class B:virtual public A
{
        public:
                int b;
};
class C:virtual public A
```

```
{
        public:
                int c;
};
class D:public B,public C
{
        public:
                void input()
                {
                        cout<<"\nenter two values:: ";
                        cin>>a>>b;
                }
                void display()
                {
                        c=a+b;
                        cout<<"\nSum is "<<c;
                }
};
void main()
{
        clrscr();
        D obj;
        obj.input();
        obj.display();
        getch();
}
```

Output:-

```
enter two values:: 5   10
sum is:: 15
```

33. <u>WRITE A PROGRAM TO DEMONSTRATE VIRTUAL FUNCTION</u>

Solution:-

```
        //virtual function
#include<iostream.h>
#include<conio.h>
class base
{
        public:
                int x;
                virtual void display()
                {
```

```
                x=5;
                cout<<"\nThe value of x in base class is :: "<<x;
            }
};
class derive:public base
{
    public:
            int x;
        void display()
            {
                x=10;
                cout<<"\nThe value of x in derive class is :: "<<x;
            }
};
void main()
{
    clrscr();
    base obb,*abhi;
    derive obd;
    abhi=&obb;
    abhi->display();
    abhi=&obd;
    abhi->display();
    getch();
}
```

Output:-

```
The value of x in base class is:: 5
The value of x in derive class is:: 10
```

34. WRITE A PROGRAM TO DEMONSTRATE PURE VIRTUAL FUNCTION

Solution:-

```
#include<iostream.h>
#include<conio.h>
class BaseClass      //Abstract class
{
   public:
     virtual void Display1()=0;    //Pure virtual function or abstract function
     virtual void Display2()=0;    //Pure virtual function or abstract function
```

```
        void Display3()
        {
            cout<<"\n\tThis is Display3() method of Base Class";
        }
    };
     class DerivedClass : public BaseClass
  {
      public:
       void Display1()
       {
            cout<<"\n\tThis is Display1() method of Derived Class";
       }
       void Display2()
       {
            cout<<"\n\tThis is Display2() method of Derived Class";
       }
  };
  void main(
{
     DerivedClass D;
     D.Display1();
      D.Display2();
      D.Display3();
}
```

Output :

```
This is Display1() method of Derived Class
This is Display2() method of Derived Class
This is Display3() method of Base Class
```

35. <u>WRITE A PROGRAM TO DEMONSTRATE FUNCTION TEMPLATE</u>

Solution:-

```
#include<iostream.h>
#include<conio.h>
template <class t>
void sort(t arr[],int n)
{
      t i,j,x;
      for(i=1;i<n;i++)
      {
```

```
                for(j=i-1,x=arr[i];j>=0&&x<arr[j];j--)
                {
                        arr[j+1]=arr[j];
                }
                arr[j+1]=x;
        }
}
void main()
{
        clrscr();
        int i,n;
        int arr[50];
        float arr1[50];
        cout<<"\nhow many number you want to enter?\n";
        cin>>n;
        cout<<"\nenter the numbers:: ";
        for(i=0;i<n;i++)
                cin>>arr[i];
        cout<<"\nbefore int sort the values are"<<endl;
        for(i=0;i<n;i++)
                cout<<arr[i]<<"\t";
        sort(arr,n);
        cout<<"\nafter int sort the values are"<<endl;
        for(i=0;i<n;i++)
                cout<<arr[i]<<"\t";
        cout<<"\nhow many number you want to enter?\n";
        cin>>n;
        cout<<"\nenter the numbers:: ";
        for(i=0;i<n;i++)
                cin>>arr1[i];
        cout<<"\nbefore float sort the values are"<<endl;
        for(i=0;i<n;i++)
                cout<<arr1[i]<<"\t";
        sort(arr1,n);
        cout<<"\nafter float sort the values are"<<endl;
        for(i=0;i<n;i++)
                cout<<arr1[i]<<"\t";
        getch();
}

Output:-
how many number you want to enter?
3
```

```
enter the numbers:: 5
4
3
before int sort the values are
5   4   3
after int sort the values are
3   4   5
how many number you want to enter?
3
enter the numbers:: 5.5
4.4
3.3
before float sort the values are
5.5   4.4   3.3
after float sort the values are
3.3   4.3   5.5
```

36. WRITE A PROGRAM TO DEMONSTRATE CLASS TEMPLATE

Solution:-

```
#include<iostream.h>
#include<conio.h>
template<class t1,class t2>
class test
{
t1 a;
t2 b;
public:
test(t1 x,t2 y)
{
a=x;
b=y;
}
void show()
{
cout<<"\n\nThe values are=";
cout<<a<<"\t"<<b<<endl;
}
};
void main()
{
        test<float,int>t1(2.3,10);
        test<float,char>t2(5.5,'q');
        test<int,float>t3(70,12);
```

```
        t1.show();
        t2.show();
        t3.show();
        getch();
}
```

Output:-

```
 The values are= 2.3    10
The values are= 5.5     q
The values are= 70      12
```

37. <u>WRITE A PROGRAM TO DEMONSTRATE EXCEPTION HANDLING</u>

Solution:-

```
//exception handling
#include<iostream.h>
#include<conio.h>
void main()
{
        int a,b,c;
        cout<<"\nenter two values:: ";
        cin>>a>>b;
        try
        {
                if(b!=0)
                {
                        c=a/b;
                        cout<<"\ndivision result is :: "<<c;
                }
                else
                        throw(b);
        }
        catch(int a)
        {
                cout<<"\ndivided by "<<a<<" is not allowed";
        }
        getch();
}
```

Output:-

```
enter two values:: 5  0
divided by 0 is not allowed
```