

# **OOPS WITH C++ PROGRAMMING LABORATORY MANUAL**

**FOR 4<sup>TH</sup> SEM IS AND CS**

**(2011-2012)**

**BY**

**MISS. SAVITHA R  
LECTURER  
INFORMATION SCIENCE DEPTATMENT  
GOVERNMENT POLYTECHNIC  
GULBARGA**

FOR ANY FEEDBACK CONTACT TO  
EMAIL: [savitharamu@gmail.com](mailto:savitharamu@gmail.com)

**1. Write a C++ program to find the largest of three numbers using inline function.**

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

inline int largest(int &a,int &b,int &c)
{
    int big=0;
    if(a>b)
        big=a;
    else
        big=b;
    if(c>big)
        big=c;
    return big;
}

int main()
{
    int a,b,c;
    clrscr();
    cout<<"Enter Three Numbers To Find The Largest "<<endl;
    cout<<"a = ";
    cin>>a;
    cout<<"\nb = ";
    cin>>b;
    cout<<"\nc = ";
    cin>>c;
    int large=largest(a,b,c);
    cout<<"\n Largest of "<<a<<","<<b<<" and "<<c<<" is "<<large;
    getch();
    return(0);
}
```

\*\*\*\*\***OUTOUT**\*\*\*\*\*

Enter Three Numbers To Find The Largest  
a = 24  
b = 45  
c = 23  
Largest of 24,45 and 23 is 45

**2. Write a C++ program to sort an array of integer in ascending order using a function called exchange( ) which accepts two integer arguments by reference.**

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

void exchange(int (&a)[],int &n);

int main()
{
    int a[10],size;
    clrscr();
    cout<<"Enter the Array size : ";
    cin>>size;
    cout<<"Enter the Array elements :\n";
    for(int i=0;i<size;i++)
        cin>>a[i];
    exchange(a,size);
    cout<<"After sorting :\n";
    for(i=0;i<size;i++)
        cout<<a[i]<<endl;
    getch();
    return 0;
}

void exchange(int (&a)[],int &n)
{
    for(int i=0;i<n;i++)
        for(int j=0;j<n;j++)
            if(a[i]<a[j])
            {
                int temp=a[i];
                a[i]=a[j];
                a[j]=temp;
            }
}
```

\*\*\*\*\*OUTPUT\*\*\*\*\*

Enter the Array size : 10

Enter the Array elements :

15

46

89

62

-5

-78

0

5

45

9

After sorting :

-78

-5

0

5

9

15

45

46

62

89

**3. Write a C++ program to implement function overloading in order to compute power(m,n) where i) m is double and n is int ii) m and n are int.**

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

int power(int a,int b);
double power(int i, double d);

int main()
{
    int i1,i2,i3;
    double d1;
    clrscr();
    cout<<"\nEnter Two Integer Numbers To Calculate Power : ";
    cin >>i1>>i2;
    int p1=power(i1,i2);
    cout<<"\nThe Power of "<<i1<<" and "<<i2<<" is : "<<p1;
    cout<<"\nEnter One Integer And One Double To calculate power :";
    cin>>i3>>d1;
    double p2=power(i3,d1);
    cout<<"\nThe Power of "<<i3<<" and "<<d1<<" is : "<<p2;
    getch();
    return(0);
}

int power(int a,int b)
{
    return(pow(a,b));
}
double power(int i, double d)
{
    return(pow(i,d));
}
```

\*\*\*\*\*OUTPUT\*\*\*\*\*

Enter Two Integer Numbers To Calculate Power : 2 4

The Power of 2 and 4 is : 16

Enter One Integer And One Double To calculate power :2 2.5

The Power of 2 and 2.5 is : 5.656854

**4. Create a 'DISTANCE' class with :**

- feet and inches as data members
- member function to input distance
- member function to output distance
- member function to add two distance objects

**Write a main function to create objects of DISTANCE class. Input two distances and output the sum.**

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

```
class distance
{
    int feet,inch;
public:
    void getdistance();
    void putdistance();
    void adddistance(distance d1,distance d2);
};

void distance::getdistance()
{
    cout<<"\nEnter the feet : ";
    cin>>feet;
    cout<<"\nEnter the inches : ";
    cin>>inch;
}

void distance::putdistance()
{
    cout<<"\n\nfeet = "<<feet;
    cout<<"\ninch = "<< inch;
}

void distance::adddistance(distance d1,distance d2)
{
    inch=d1.inch+d2.inch;
    feet=inch/12;
    inch=inch%12;
    feet=feet+d1.feet+d2.feet;
}

int main()
{
    distance d1,d2,d3;
    clrscr();
    cout<< "\nDistance 1 \n";
    d1.getdistance();
    cout<< "\nDistance 2 \n";
    d2.getdistance();
    d3.adddistance(d1,d2);
```

```
cout<<"\nThe Sum Of two distance is : ";
d3.putdistance();
getch();
return(0);
}
```

\*\*\*\*\***OUTPUT**\*\*\*\*\*

Distance 1

Enter the feet : 13

Enter the inches :11

Distance 2

Enter the feet : 11

Enter the inches :11

The Sum Of two distance is :

feet = 25      inch = 10

**5. Create a class called 'EMPLOYEE' that has**

- **EMPCODE and EMPNAME as data members**
- **member function getdata( ) to input data**
- **member function display( ) to output data**

**Write a main function to create EMP, an array of EMPLOYEE objects. Accept and display the details of at least 6 employees.**

```
#include<iostream.h>
#include<conio.h>
#include<iomanip.h>
class Employee
{
    private: int empcode;
            char empname[10];
    public: void getdata();
            void display();
};
void Employee::getdata()
{
    cout<<"\nNAME :";
    cin>>empname;
    cout<<"\nCODE :";
    cin>>empcode;
}
void Employee::display()
{
    cout<<endl<<setw(20)<<empname<<setw(10)<<empcode;
}

int main()
{
    Employee Emp[6];
    clrscr();
    cout<< "Enter employee details:\n ";
    for(int i=0;i<6;i++)
    {
        cout<<"\nemployee "<<i+1<<endl;
        Emp[i].getdata();
    }
    cout<<"\nEmployee details are as follows :";
    cout<<"\n\n" <<setw(20)<<"NAME"<<setw(10)<<setiosflags(ios::right)<<"CODE";
    cout<<"\n-----";
    for(i=0;i<6;i++)
        Emp[i].display();
    getch();
    return(0);
}
```

\*\*\*\*\*OUTPUT\*\*\*\*\*

Enter employee details:

employee 1

NAME :ashok

CODE :111

employee 2

NAME :annapurna

CODE :112

employee 3

NAME :anupama

CODE :113

employee 4

NAME :anuradha

CODE :114

employee 5

NAME :ashraya

CODE :115

employee 6

NAME :akash

CODE :116

Employee details are as follows :

NAME	CODE
Ashok	111
annapurna	112
anupama	113
anuradha	114
ashraya	115
akash	116

**6. Create a class called 'TIME' that has**

- three integer data members for hours, minutes and seconds
- constructor to initialize the object to zero
- constructor to initialize the object to some constant value
- member function to add two TIME objects
- member function to display time in HH:MM:SS format

**Write a main function to create two TIME objects, add them and display the result in HH:MM:SS format.**

```
#include<iostream.h>
#include<conio.h>
class Time
{
    int hrs,min,sec;

public: Time()
{
    hrs=0;min=0;sec=0;
}

Time(int a,int b,int c)
{
    hrs=a; min=b;sec=c;
}

void addTime(Time tt1,Time tt2)
{
    sec=tt1.sec+tt2.sec;
    min=sec/60;
    sec=sec%60;
    min=min+tt1.min+tt2.min;
    hrs=min/60;
    min=min%60;
    hrs=hrs+tt1.hrs+tt2.hrs;
}

void display()
{
    cout<<hrs<<"."<<min<<"."<<sec<<endl;
}

};

int main()
{
    clrscr();
```

```
Time t1(10,48,30);
Time t2(2,22,35);
Time t3;
t3.addTime(t1,t2);
cout<<"\nTime 1 is :";
t1.display();
cout<<"\nTime 2 is :";
t2.display();
cout<<"\nTime 1 + Time 2 is :";
t3.display();
getch();
return(0);
}
```

**\*\*\*\*\*OUTPUT\*\*\*\*\***

Time 1 is :10:48:30

Time 2 is :2:22:35

Time 1 + Time 2 is :13:11:5

**7. Create a class 'COMPLEX' to hold a complex number. Write a friend function to add two complex numbers. Write a main function to add two COMPLEX objects.**

```
#include<iostream.h>
#include<conio.h>
class complex
{
    float real,imag;
public: void get_complex();
        void show_complex();
        friend complex add_complex(complex c1,complex c2);
};

void complex::get_complex()
{
    cout<<"Enter real number :";
    cin>> real;
    cout<<"Enter Imaginary number :";
    cin>> imag;
}
void complex::show_complex()
{
    cout<<real<<"+"<<imag;
}
complex add_complex(complex c1,complex c2)
{
    complex c;
    c.real=c1.real+c2.real;
    c.imag=c1.imag+c2.imag;
    return c;
}

int main()
{
    clrscr();
    complex c1,c2,c3;
    c1.get_complex();
    c2.get_complex();
    c3=add_complex(c1,c2);
    cout<<"\nComplex Number 1 = ";
    c1.show_complex();
    cout<<"\nComplex Number 2 = ";
    c2.show_complex();
    cout<<"\nSum of Complex Number 1 and 2 = ";
    c3.show_complex();
    getch();
    return 0;
}
```

\*\*\*\*\*OUTPUT\*\*\*\*\*

Enter real number :12  
Enter Imaginary number :10  
Enter real number :3  
Enter Imaginary number :5

Complex Number 1 = 12+i10  
Complex Number 2 = 3+i5  
Sum of Complex Number 1 and 2 = 15+i15

**8. Create a 'MATRIX' class of size m X n. Overload the '+' operator to add two MATRIX objects. Write a main function to implement it.**

```
#include<iostream.h>
#include<conio.h>
class mat
{
    int m,n,a[20][20];
public:
    mat(int x,int y);
    void readmat();
    mat operator +(mat);
    void display();
};

mat :: mat(int x,int y)
{
    m=x;n=y;
    for(int i=0;i<m;i++)
    {
        for(int j=0;j<n;j++)
            a[i][j]=0;
    }
}

void mat :: readmat()
{
    cout<<"\nEnter matrix elements\n";
    for(int i=0;i<m;i++)
        for(int j=0;j<n;j++)
            cin>>a[i][j];
}

mat mat:: operator +(mat obj)
{
    mat temp(m,n);
    for(int i=0;i<m;i++)
        for(int j=0;j<n;j++)
    {
        temp.a[i][j]=a[i][j]+obj.a[i][j];
    }
    return temp;
}

void mat:: display()
{
    int i,j;
    for(i=0;i<m;i++)
    {
        cout<<"\n\n";
        for(j=0;j<n;j++)
            cout<<"\t"<<a[i][j];
    }
}
```

```
int main()
{
    int m1,n1;
    clrscr();
    cout<<"\nEnter the size(m,n) of matrix: ";
    cin>>m1>>n1;
    mat a(m1,n1),b(m1,n1),c(m1,n1);
    cout<<"\nEnter matrix 1: ";
    a.readmat();
    cout<<"\nEnter matrix 2: ";
    b.readmat();
    c=a.operator +(b);
    cout<<"\nFirst Matrix :\n";
    a.display();
    cout<<"\nSecond Matrix :\n";
    b.display();
    cout<<"\nmatrix 1+matrix 2: ";
    c.display();
    getch();
    return 0;
}
```

\*\*\*\*\*OUTPUT\*\*\*\*\*

Enter the size(m,n) of matrix: 2 2

Enter matrix 1: enter matrix elements

3 3  
3 3

Enter matrix 2: enter matrix elements

4 4  
4 4

First Matrix :

3 3  
3 3

Second Matrix :

4 4  
4 4

matrix 1 + matrix 2:

7 7  
7 7

**9. Derive a class ‘MAT’ from MATRIX class created in program No. 8. Add a member function to overload ‘\*’ operator to multiply two objects. (Single Inheritance)**

```
#include<iostream.h>
#include<conio.h>
class matrix
{
public: int m,n;
    int a[20][20];
    void mat_to_zero(int x,int y);
    void readmat();
    void display();
};

void matrix::mat_to_zero(int x,int y)
{
    m=x;
    n=y;
    for(int i=0;i<m;i++)
        for(int j=0;j<n;j++)
            a[i][j]=0;
}

void matrix::readmat()
{
    cout<<"enter the matrix element";
    for(int i=0;i<m;i++)
        for(int j=0;j<n;j++)
            cin>>a[i][j];
}

void matrix::display()
{
    for(int i=0;i<m;i++)
    {
        cout<<"\n";
        for(int j=0;j<n;j++)

```

```
    cout<<" "<<a[i][j];  
}  
}  
class mat:public matrix  
{  
public: mat operator * (mat obj);  
};  
mat mat::operator *(mat obj)  
{  
    mat temp;  
    temp.mat_to_zero(m,n);  
    for(int i=0;i<m;i++)  
        for(int j=0;j<n;j++)  
    {  
        temp.a[i][j]=0;  
        for(int k=0;k<n;k++)  
        {  
            temp.a[i][j]=temp.a[i][j]+a[i][j]*obj.a[j][i];  
        }  
    }  
    return temp;  
}  
int main()  
{  
    int m1,n1;  
    clrscr();  
    cout<<"enter the order of matrix: ";  
    cin>>m1>>n1;  
    mat a,b,c;  
    a.mat_to_zero(m1,n1);  
    b.mat_to_zero(m1,n1);  
    cout<<"enter the matrix 1: \n";  
    a.readmat();
```

```
cout<<"enter the matrix 2: \n";
b.readmat();
c=a*b;
cout<<"\n matrix 1: ";
a.display();
cout<<"\n matrix 2: ";
b.display();
cout<<"\n sum of two matrix: ";
c.display();
getch();
return 0;
}
```

\*\*\*\*\*OUTPUT\*\*\*\*\*

```
enter the order of matrix: 2 2
enter the matrix 1:
enter the matrix element2 2 2 2
enter the matrix 2:
enter the matrix element2 2 2 2
```

```
matrix 1:
2 2
2 2
matrix 2:
2 2
2 2
sum of two matrix:
8 8
8 8
```

**10. Write a c++ program :****a) to illustrate multilevel inheritance.**

```
#include<iostream.h>
#include<conio.h>
class student
{
protected : int rollno;
public : void get_num();
           void put_num();
};

void student::get_num()
{
    cout<<"\nEnter the roll number:\t";
    cin>>rollno;
}

void student::put_num()
{
    cout<<"Rollnumber: "<<rollno;
}

class test:public student
{
protected : float sub1,sub2;
public:
    void get_marks()
    {
        cout<<"\nEnter the sub1 marks: ";
        cin>>sub1;
        cout<<"\nEnter the sub2 marks: ";
        cin>>sub2;
    }

    void put_marks()
    {
        cout<<"\nSub1="<<sub1;
```

```
cout<<"\nSub2="<<sub2;
    }
};

class result : public test
{
    float total;

public:
    void display()
    {
        total=sub1+sub2;
        put_num();
        put_marks();
        cout<<"\nTotal= "<<total;
    }
};

int main()
{
    clrscr();
    result r;
    r.get_num();
    r.get_marks();
    r.display();
    getch();
    return 0;
}
```

\*\*\*\*\***OUTPUT**\*\*\*\*\*

Enter the roll number: 11111

Enter the sub1 marks: 67

Enter the sub2 marks: 56

Rollnumber: 11111

Sub1=67

Sub2=56

Total= 123

**10. Write a c++ program :****b) to illustrate multiple inheritance**

```
#include<iostream.h>
#include<conio.h>
class m
{
protected : int m;
public:
void getm()
{
    cout<<"\nEnter the value for m : ";
    cin>>m;
}
};

class n
{
protected : int n;
public:
void getn()
{
    cout<<"\nEnter the value for n : ";
    cin>>n;
}
};

class p : public m , public n
{
public:
void display()
{
    cout<<"\nM=" <<m;
    cout<<"\nN=" <<n;
    cout<<"\nM*N=" <<m*n;
}
};
```

```
int main()
{
    clrscr();
    p p1;
    p1.getm();
    p1.getn();
    p1.display();
    getch();
    return 0;
}
```

\*\*\*\*\***OUTPUT**\*\*\*\*\*

Enter the value for m : 12

Enter the value for n : 10

M=12

N=10

M\*N=120

**11. Create a 'STRING' class which overloads '==' operator to compare two STRING objects.**

```
#include<iostream.h>
#include<conio.h>
#include<string.h>
class string
{
    char *p;
    int len;
public:
    string() { }
    string( char *s)
    {
        len=strlen(s);
        p=new char[len+1];
        strcpy(p,s);
    }
    friend int operator ==( string &s,string &t);
};

int operator == ( string &s1,string &s2)
{
    if(strcmp(s1.p,s2.p)==0)
        return(1);
    else
        return(0);
}

int main()
{
    string s1,s2;
    char str1[20],str2[20];
    clrscr();
    cout<<"\nEnter the first string : ";
    cin>>str1;
    cout<<"\nEnter the second string : ";
    cin>>str2;
    s1=str1;
    s2=str2;
    if(s1==s2)
        cout<<"\nStrings are equal";
    else
        cout<<"\nStrings are not equal";
    getch();
    return 0;
}
```

\*\*\*\*\***OUTPUT 1**\*\*\*\*\*

Enter the first string : cpc

Enter the second string : cs

Strings are not equal

\*\*\*\*\***OUTPUT 2**\*\*\*\*\*

Enter the first string : CPC

Enter the second string : CPC

Strings are equal

**12. Write a C++ program to illustrate ‘this’ pointer and pointers to derived classes.**

```
#include<iostream.h>
#include<conio.h>
class BC
{
public:
int b;
void show()
{
    cout<<"b= "<<b<<endl;
}
BC findlarge(BC obj)
{
    if(b>obj.b)
        return *this;
    else
        return obj;
}
class DC:public BC
{
public:
int d;
void show()
{
    cout<<"b= "<<b<<endl;
    cout<<"d= "<<d<<endl;
}
int main()
{
    clrscr();
    BC b1,b2;
    b1.b=10;
    b2.b=20;
    BC Large=b1.findlarge(b2);
    cout<<"\n Largest is :";
    Large.show();
    BC *bptr;
    BC base;
    bptr=&base;
    bptr->b=100;
    cout<<"Base pointer to base class\n";
    bptr->show();
    DC derived;
    bptr=&derived;
    bptr->b=200;
    cout<<"Base pointer to base class\n";
```

```
bptr->show();
DC *dptr;
dptr=&derived;
dptr->d=300;
cout<<"Derived pointer to derived class\n";
dptr->show();
((DC*)bptr)->d=400;
cout<<"Type conversion\n";
((DC*)bptr)->show();
getch();
return 0;
}
```

\*\*\*\*\*OUTPUT\*\*\*\*\*

```
Largest is :b= 20
Base pointer to base class
b= 100
Base pointer to base class
b= 200
Derived pointer to derived class
b= 200
d= 300
Type conversion
b= 200
d= 400
```

**13. Create a base class called 'SHAPE' having**

- two data members of type double
- member function *get-data()* to initialize base class data members
- pure virtual member function *display-area()* to compute and display the area of the geometrical object.

**Derive two specific classes 'TRIANGLE' and 'RECTANGLE' from the base class.**

**Using these three classes design a program that will accept dimension of a triangle / rectangle interactively and display the area.**

```
#include<iostream.h>
#include<conio.h>
class shape
{
protected:double x, y;
public:void getdata(double a, double b)
{
    x=a;
    y=b;
}
virtual void display_area()=0;
};
class triangle:public shape
{
    double triangle_area;
    void display_area()
    {
        triangle_area=(1*x*y)/2;
        cout<<"area of triangle is:"<<triangle_area<<endl;
    }
};
class rectangle:public shape
{
    double rectangle_area;
    void display_area()
    {
        rectangle_area=x*y;
        cout<<"area of rectangle is:"<<rectangle_area;
    }
};
int main()
{
    clrscr();
    shape *p;
    triangle t;
    rectangle r;
```

```
p=&t;
p->getdata(10,30);
p->display_area();
p=&r;
p->getdata(20,30);
p->display_area();
getch();
return 0;
}
```

\*\*\*\*\***OUTPUT**\*\*\*\*\*

area of triangle is:150  
area of rectangle is:600

**14. Write a C++ program to read a list containing item name, item code and cost interactively and display the data in a tabular format as shown below:**

NAME	CODE	COST
------	------	------

```
#include<iostream.h>
#include<conio.h>
#include<iomanip.h>
int main()
{
    int n,itcode[10];
    char item[10][10];
    float itcost[10];
    clrscr();
    cout<<"Enter the no of item:";
    cin>>n;
    for(int i=0;i<n;i++)
    {
        cout<<"Enter the item details:"<<i+1;
        cout<<"\nEnter name:";
        cin>>item[i];
        cout<"Enter code:";
        cin>>itcode[i];
        cout<<"Enter cost:";
        cin>>itcost[i];
    }
    cout<<endl<<setw(15)<<"Item name"<<setw(15)<<"item code"<<setw(15)<<"cost"<<endl;
    cout<<"\n-----\n";
    for(i=0;i<n;i++)
    {
        cout.fill( ' - ' );
        cout.setf(ios::left,ios::adjustfield);
```

```
cout.width(15);
cout<<item[i];
cout.setf(ios::left,ios::adjustfield);
cout.width(15);
cout<<itcode[i];
cout.setf(ios::right,ios::adjustfield);
cout<<itcost[i]<<endl;
}
getch();
return 0;
}
*****OUTPUT*****
```

Enter the no of item:3

Enter the item details:1

Enter name:Rice

Enter code:111

Enter cost:35.4

Enter the item details:2

Enter name:Sugar

Enter code:112

Enter cost:12.3

Enter the item details:3

Enter name:Soap

Enter code:113

Enter cost:60.4

Item name	item code	cost
-----------	-----------	------

---

Rice-----	111-----	35.400002
-----------	----------	-----------

Sugar-----	112-----	12.3
------------	----------	------

Soap-----	113-----	60.400002
-----------	----------	-----------

**15. Design your own manipulator to provide the following output specification for printing money value:**

- 1) 10 columns width**
- 2) The character '\$' at the beginning**
- 3) Showing '+' sign.**
- 4) Two digits precision**
- 5) Filling of unused spaces with '\*'**
- 6) Trailing zeros shown**

```
#include<iostream.h>
#include<conio.h>
#include<iomanip.h>
ostream &moneyshow(ostream &output)
{
    cout<<'$';
    cout.fill('*');
    cout.setf(ios::showpos);
    cout.setf(ios::showpoint);
    cout.width(10);
    cout.precision(2);
    return output;
}
void main()
{
    double amount;
    clrscr();
    cout<<"enter the value:";
    cin>>amount;
    cout<<"\nyour money value:";
    cout<<moneyshow<<amount;
    getch();
}
```

\*\*\*\*\***OUTPUT 1**\*\*\*\*\*

enter the value:12.2

your money value:\$\*\*\*\*+12.20

\*\*\*\*\***OUTPUT 2**\*\*\*\*\*

enter the value:34

your money value:\$\*\*\*\*+34.00

**16. Write a C++ program that uses a single file for both reading and writing the data.**

```
#include<iostream.h>
#include<conio.h>
#include<fstream.h>
int main()
{
    clrscr();
    ofstream outf("ITEM.txt");
    cout<<"Enter the filename:";
    char name[30];
    cin>>name;
    outf<<name<<"\n";
    cout<<"Enter ITEM cost:";
    float cost;
    cin>>cost;
    outf<<cost;
    outf.close();
    ifstream inf("ITEM.txt");
    inf>>name;
    inf>>cost;
    cout<<"The name of the item is:"<<name;
    cout<<"\nItem cost is :"<<cost;
    inf.close();
    getch();
    return 0;
}
```

\*\*\*\*\*OUTPUT\*\*\*\*\*

```
Enter the ITEM name:vicks
Enter ITEM cost:20.3
The name of the item is:vicks
Item cost is :20.299999
```

**17. A file contains a list of names and telephone numbers in the following form:**

**Name              Tel. No.**

**Write a C++ program to read the file and output the list in the tabular format. The name should be left-justified and numbers right-justified. Use a class object to store each set of data.**

```
#include<iostream.h>
#include<conio.h>
#include<iomanip.h>
#include<fstream.h>
class phone
{
public:
    char name[20];
    long int phone_no;
    void getdata()
    {
        cout<<"Enter the name and phone_no"<<"\n";
        cin>>name>>phone_no;
    }
    void putdata()
    {
        cout<<endl<<setw(20)<<setiosflags(ios::left)<<name<<setw(10)
            <<setiosflags(ios::right)<<phone_no<<endl;
    }
};
void main()
{
    fstream f;
    phone ph;
    int n;
    f.open("phonefile",ios::out|ios::in|ios::app);
```

```
clrscr();
cout<<"\nEnter the total no_of records:";
cin>>n;
for(int i=0;i<n;i++)
{
    ph.getdata();
    f.write((char *)&ph,sizeof(ph));
}
f.seekg(0);
cout<<"\n\nThe content of the file is:\n\n";
cout<<setw(20)<<setiosflags(ios::left)<<"NAME"
    <<setw(10)<<setiosflags(ios::right)<<"TELEPHONE_NO";
while(f.read((char *)&ph,sizeof(ph)))
    ph.putdata();
f.close();
getch();
}
```

**\*\*\*\*\*OUTPUT\*\*\*\*\***

Enter the total no\_of records:3

Enter the name and phone\_no

RAGHU

2456570

Enter the name and phone\_no

RAJEEV

2457859

Enter the name and phone\_no

RANJU

2451230

The content of the file is:

NAME	TELEPHONE_NO
RAGHU	2456570

RAJEEV	2457859
--------	---------

RANJU	2451230
-------	---------

**18. Write an interactive, menu-driven program that will access the file created in program No.17 and implement the following tasks:**

- i) To determine the telephone numbers of the specified person.**
- ii) To determine the name if a telephone number is given.**
- iii) To update the telephone number whenever there is a change.**

```
#include<iostream.h>
#include<fstream.h>
#include<iomanip.h>
#include<conio.h>
#include<string.h>
#include<stdlib.h>

class phone
{
public: long int no;
char name[10];
void getdata()
{
    cout<<endl<<"enter the name :";
    cin>>name;
    cout<<endl<<"enter the telephone number :";
    cin>>no;
}
void putdata()
{
    cout<<endl<<setw(10)<<name<<endl<<setw(10)<<no<<endl;
}
};

int main()
{
    phone ph;
    fstream f;
    clrscr();
```

```
int n;
f.open("pp1.txt",ios::trunc|ios::in|ios::out|ios::ate);
cout<<"Enter the number of records :";
cin>>n;
for(int i=0;i<n;i++)
{
    ph.getdata();
    f.write((char *)&ph,sizeof(ph));
}
f.seekg(0);
cout<<endl<<"contents of the file are as follows ";
while(f.read((char *)&ph,sizeof(ph)))
    ph.putdata();
int choice;
while(1)
{
    cout<<"\n1.get name from telephone number";
    cout<<"\n2.get telephone number from name";
    cout<<"\n3.alter the telephone number";
    cout<<"\n4.EXIT";
    cout<<"\nEnter ur choice ";
    cin>>choice;
    f.seekg(0);
    f.clear();
    switch(choice)
    {
        case 1:cout<<"\nEnter the telephone number to get the name ";
        int no;
        cin>>no;
        f.seekg(0);
        f.clear();
        while(f.read((char *)&ph,sizeof(ph)))
        {

```

```
if(ph.no==no)
{
    ph.putdata();
}
break;

case 2:cout<<"\nEnter the name to get the telephone number";
char search_name[10];
cin>>search_name;
while(f.read((char *)&ph,sizeof(ph)))
{
    if(strcmp(ph.name,search_name)==0)
        ph.putdata();
}
break;

case 3:cout<<"\nEnter the name to modify the telephone number ";
char searchname[10];
cin>>searchname;
while(f.read((char *)&ph,sizeof(ph)))
{
    if(strcmp(ph.name,searchname)==0)
    {
        ph.putdata();
        cout<<endl<<"Enter the new details to modify :";
        ph.getdata();
        int loc(sizeof(ph));
        int p=f.tellp();
        cout<<endl<<p;
        f.seekp(p-loc);
        f.write((char *)&ph,sizeof(ph));
    }
}
cout<<endl<<"After modification the content of the file is as follows ";
```

```
f.clear();
f.seekg(0);
while(f.read((char *)&ph,sizeof(ph)))
    ph.putdata();
break;
case 4:exit(0);
}
{
}
```

\*\*\*\*\*OUTPUT\*\*\*\*\*

Enter the number of records :3

enter the name :aaa

enter the telephone number :11111

enter the name :bbb

enter the telephone number :22222

enter the name :ccc

enter the telephone number :33333

contents of the file are as follows

aaa

11111

bbb

22222

ccc

33333

1.get name from telephone number

2.get telephone number from name

3.alter the telephone number

4.EXIT

enter ur choice 1

Enter the telephone number to get the name 11111

found

aaa

11111

not foundnot foundnot found

1.get name from telephone number

2.get telephone number from name

3.alter the telephone number

4.EXIT

enter ur choice 2

Enter the name to get the telephone numberccc

ccc

33333

1.get name from telephone number

2.get telephone number from name

3.alter the telephone number

4.EXIT

enter ur choice 3

Enter the name to modify the telephone number ccc

ccc

33333

Enter the new details to modify :

enter the name :ccc

enter the telephone number :44444

42

After modification the content of the file is as follows

aaa

11111

bbb

22222

ccc

44444

1.get name from telephone number

2.get telephone number from name

3.alter the telephone number

4.EXIT

enter ur choice 4

**19. Write a C++ program that displays the size (in bytes) of a given file. The name of the file is specified as command line argument.**

```
#include<iostream.h>
#include<fstream.h>
#include<conio.h>
void main(int argc,char*argv[])
{
    int bcount=0;
    char cl;
    ifstream testfile;
    clrscr();
    cout<<"enter the file name\n";
    cin>>argv[1];
    testfile.open(argv[1],ios::in);
    while(testfile)
    {
        testfile.read((char*)&cl,sizeof(cl));
        ++bcount;
    }
    testfile.close();
    cout<<"size of the given file:"<<argv[1]<<"is "<<bcount<<"bytes"<<endl;
    getch();
}
```

\*\*\*\*\***OUTPUT**\*\*\*\*\*

enter the file name

ph.cpp

size of the given file:is2308bytes

**20. Define a function template for finding the minimum value contained in an array.  
Write main( ) function to find the minimum value of integer array and minimum value  
of floating point numbers in an array.**

```
#include<iostream.h>
#include<conio.h>
template<class T>
T minimum(T a[],int size)
{
    T min=a[0];
    for(int i=0;i<size;i++)
    {
        if(a[i]<min)
            min=a[i];
    }
    return min;
}
int main()
{
    int a[10],size,i,min1;
    float b[10],min2;
    clrscr();
    cout<<"enter the size value:\n";
    cin>>size;
    cout<<"enter the integer aray elements\n";
    for(i=0;i<size;i++)
        cin>>a[i];
    cout<<"enter the floating array elements\n";
    for(i=0;i<size;i++)
        cin>>b[i];
    min1=minimum(a,size);
    min2=minimum(b,size);
    cout<<"The minimum integer elements is:\n";
    cout<<min1;
```

```
cout<<"\nThe minimum floating elements is :\n";
cout<<min2;
getch();
return 0;
}
```

\*\*\*\*\***OUTPUT**\*\*\*\*\*

enter the size value:

5

enter the integer array elements

20

30

10

38

28

enter the floating array elements

20.4

19.7

14.8

1.7

2.6

The minimum integer elements is:

10

The minimum floating elements is :

1.7

**21. Write a class template to represent a generic vector. Include member functions to perform the following tasks:**

- 1) To create the vector.**
- 2) To modify the value of a given element.**
- 3) To multiply the vector by a scalar value.**
- 4) To display the vector in the form (10, 20, 30,.....)**

```
#include<iostream.h>
#include<string.h>
#include<conio.h>
template<class T>
class vector
{
    T *v;
    int size;
public:
    vector(int m);
    void create(T *a);
    void modify(int);
    void multiply(int);
    void display();
};

template<class T>
vector< T > :: vector(int m)
{
    v=new T[size=m];
    for(int i=0;i<size;i++)
        v[i]=0;
}

template<class T>
void vector<T>::create(T*a)
{
    for(int i=0;i<size;i++)
    {
        cin>>a[i];
        v[i]=a[i];
    }
}

template<class T>
void vector<T>::modify(int k)
{
    v[k]=v[k]+10;
}

template<class T>
```

```
void vector<T>::multiply(int k)
{
    for(int i=0;i<size;i++)
        v[i]=v[i]*k;
}

template<class T>
void vector<T>::display()
{
    cout.setf(ios::showpoint);
    cout<<"\n";
    for(int i=0;i<size;i++)
        cout<<v[i]<<", ";
    cout<<")\n";
}

int main()
{
    vector <float> v1(5);
    vector <int> v2(5);
    float *x;
    int *y;
    int i;
    int s;
    cout<<"Enter the float vector element :\n";
    v1.create(x);
    cout<<"Enter the Interger vector element :\n";
    v2.create(y);
    cout<<"enter the element u want to modify in float vector :";
    cin>>i;
    v1.modify(i);
    v1.display();
    cout<<"\nEnter the element u want to modify in int vector :";
    cin>>i;
    v2.modify(i);
    v2.display();
    cout<<"\nEnter the number to calculate the scalar product :";
    cin>>s;
    v1.multiply(s);
    v2.multiply(s);
    cout<<"\nthe Float vector after scalar product is as follows :";
    v1.display();
    cout<<"\nthe integer vector after scalar product is as follows :";
    v2.display();
    getch();
    return(0);
}
```

\*\*\*\*\*OUTPUT\*\*\*\*\*

Enter the float vector element :

1.1  
2.2  
3.3  
4.4  
5.5

Enter the Integer vector element :

10  
10  
30  
40  
50

enter the element u want to modify in float vector :3

(1.100000, 2.200000, 3.300000, 14.400000, 5.500000, )

enter the element u want to modify in int vector :2

(10, 10, 40, 40, 50, )

enter the number to calculate the scalar product :2

the Float vector after scalar product is as follows :

(2.200000, 4.400000, 6.600000, 28.799999, 11.000000, )

the integer vector after scalar product is as follows :

(20, 20, 80, 80, 100, )