

GDC MEMORIAL COLLEGE

BAHAL (BHIWANI)-127028



Lab Manual

PROGRAMMING IN C++ (B.Sc.(cs) 2nd year)

Department of Computer Science

B. Sc. (Computer Science) Semester-III
Programming in C++ Lab (Paper 3.2)

List of Programs:

1. WAP to swap two numbers using call by value and call by reference.
2. WAP to illustrate the concept of a class.
3. WAP to illustrate inline function.
4. WAP to illustrate the concept of default arguments.
5. WAP to compute the area of a triangle and a circle by overloading area() function.
6. WAP to illustrate static class members and member function.
7. WAP to illustrate object as function arguments.
8. WAP to illustrate friend function.
9. WAP that implements multiple constructors in a class.
10. WAP that overload unary operator using friend function.
11. WAP that overload binary operator using member function.
12. WAP to implement hybrid inheritance.
13. WAP to show the use of virtual base class.
14. WAP that illustrate the concept of 'this' pointer.
15. WAP to illustrate the concept of virtual function.

/* PROGRAM 1: WAP TO SWAP TWO NUMBERS USING CALL BY VALUE & CALL BY REFERENCE*/

```
#include<iostream.h>

#include<conio.h>

void call_by_value(int,int);

void call_by_reference(int &,int &);

void main()
{
int a,b;
clrscr();
cout<<"Entre the value of a & b:";
cin>>a>>b;
cout<<"1. Call by Value:";
cout<<"Values of a & b before swap:"
cout<<"a:"<<a<<"\t b:"<<b;
call_by_value(a,b);
cout<<"2. Call by Reference:";
cout<<"Values of a & b before swap:"
cout<<"a:"<<a<<"\t b:"<<b;
call_by_reference(a,b);
cout<<"\n Values of a & b after swap";
cout<<"a:"<<a<<"\t b:"<<b;
getch();
}

void call_by_value(int a,int b){
```

```
int temp;
temp=a;
a=b;
b=temp;
cout<<"\n Values of a & b after swap:";
cout<<"\na:"<<a<<"\t b:"<<b;}
```

```
void call_by_reference(int &x,int &y){
int temp;
temp=x;
x=y;
y=temp;}
```

/*PROGRAM 2: WAP TO ILLUSTRATE THE CONCEPT OF CLASS*/

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

```
class student
```

```
{
```

```
char name[50];
```

```
int age;
```

```
public:
```

```
void getdata()
```

```
{cout<<"\nEnter name:";
```

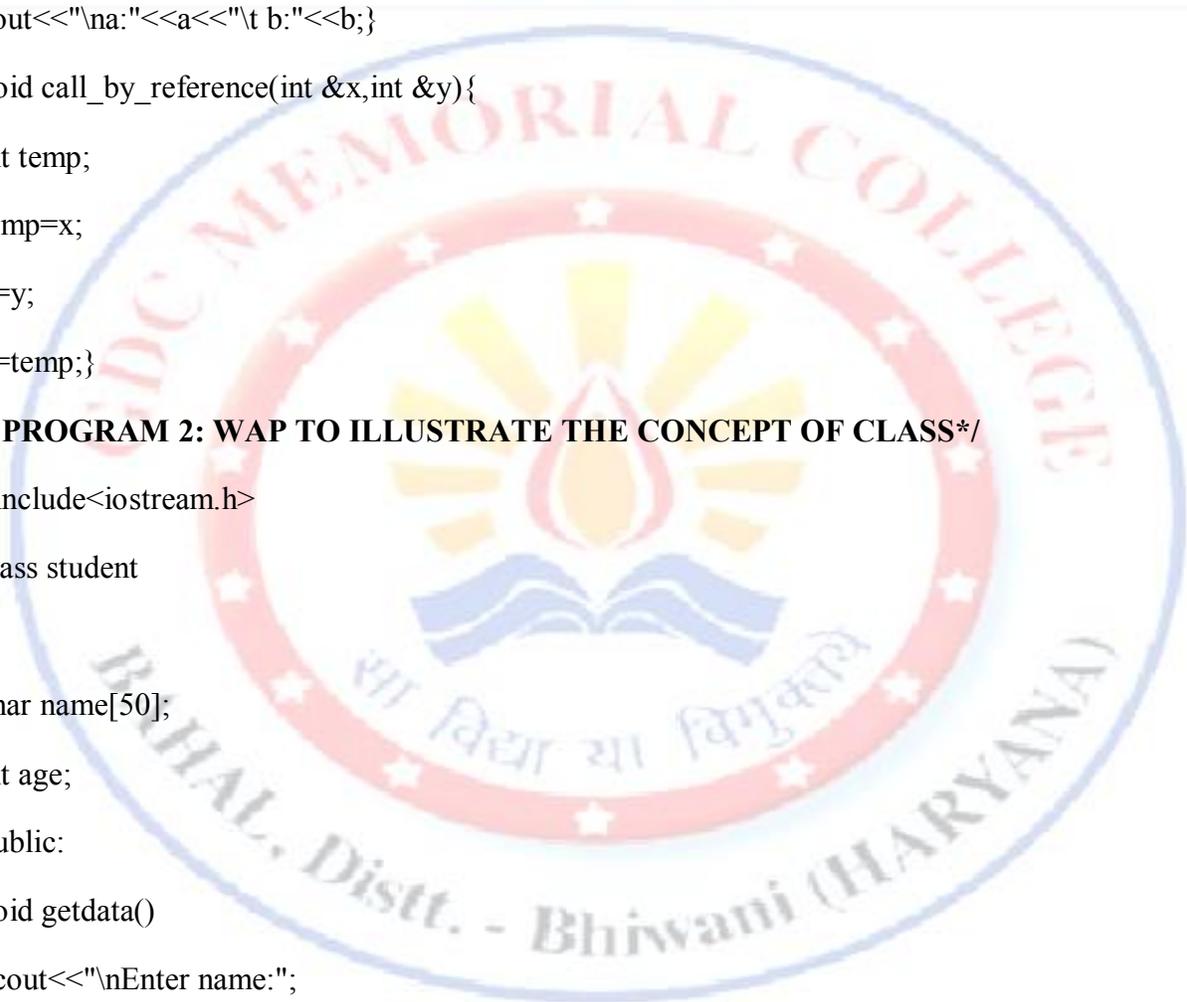
```
cin>>name;
```

```
cout<<"\nEnter age:";
```

```
cin>>age;
```

```
}
```

```
void putdata();
```



```
};  
void student::putdata()  
{  
cout<<"\nName:"<<name;  
cout<<"\nAge:"<<age;  
}
```

```
void main()  
{  
student s;  
s.getdata();  
s.putdata();  
}
```

/*PROGRAM 3: WAP TO ILLUSTRATE INLINE FUNCTION*/

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

```
inline float mul (float x,float y)
```

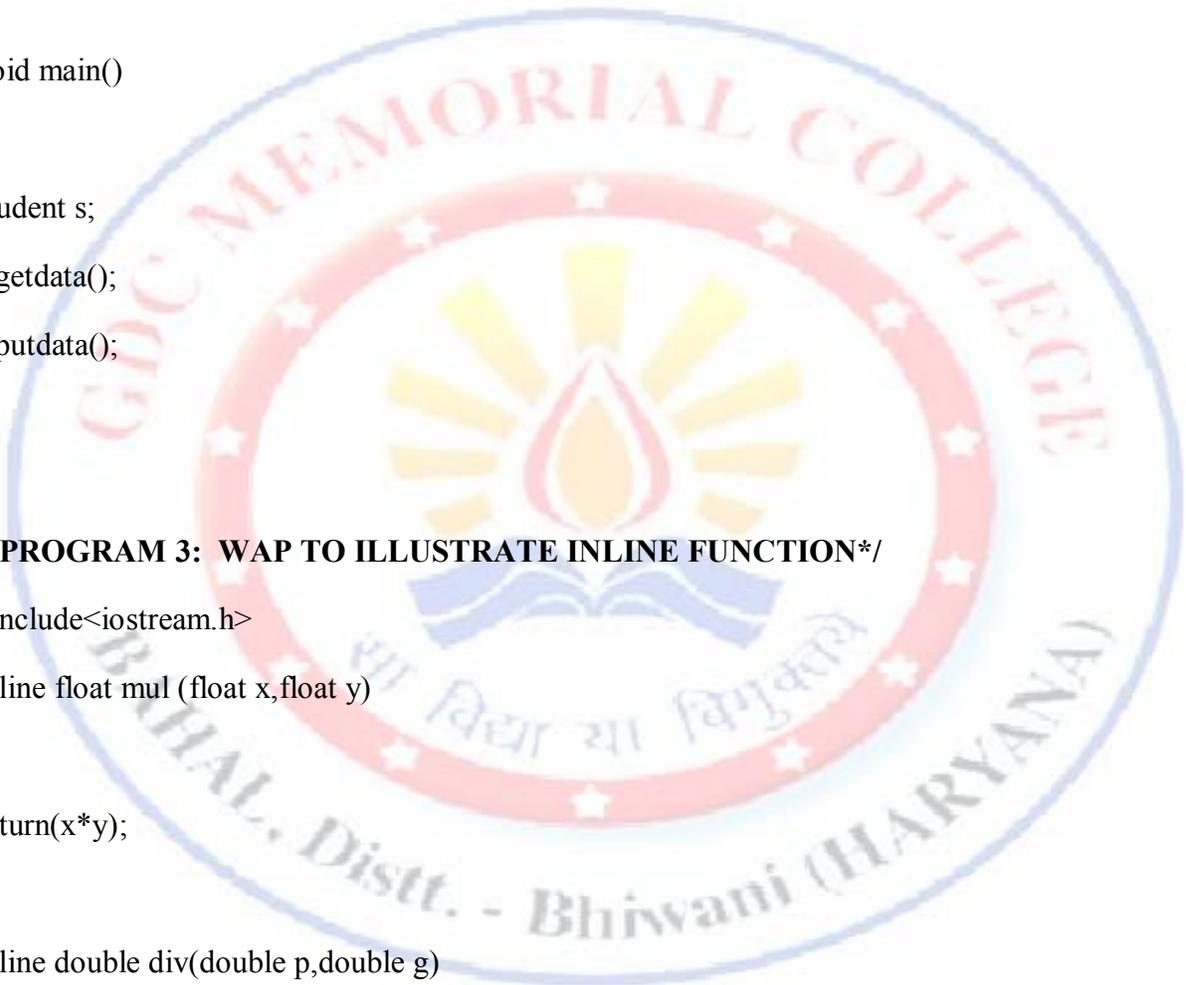
```
{  
return(x*y);  
}
```

```
inline double div(double p,double g)
```

```
{  
return(p/g);  
}
```

```
int main()
```

```
{float a=23.8;
```



```

float b=6.9 ;
cout<<"Multiplication:"<<mul(a,b)<<"\n";
cout<<"Division:"<<div(a,b)<<"\n";
return 0;
}

```

**/*PROGRAM 4: WAP TO ILLUSTRATE THE CONCEPT OF DEFAULT ARGUMENT
*/**

```

#include <iostream.h>
#include<conio.h>
int main()
{
float amount;
float value(float p,int n,float r=0.52);
void printline(char ch='*',int len=40);
printline();
amount=value(5000.00,5);
cout<<"\n frind value="<<amount<<"\n\n";
printline('=');
getch();
return 0;
}

float value(float p,int n,float r)
{
int year=1;

```

```
float sum=p;
while(year<=n)
{
sum=sum*(1+r);
year=year+1;
}
```

/*PROGRAM 5: WAP TO COMPUTE THE AREA OF SQUARE, RECTANGLE, CIRCLE AND TRIANGLE BY OVERLOADING AREA() FUNCTION.*/

```
#include <iostream.h>
#include<conio.h>
int area(int);
int area(int,int);
float area(float);
float area(float,float);
int main()
{
    int s,l,b;
    float r,bs,ht;
    cout<<"Enter side of a square:";
    cin>>s;

    cout<<"Enter length and breadth of rectangle:";
    cin>>l>>b;

    cout<<"Enter radius of circle:";
    cin>>r;

    cout<<"Enter base and height of triangle:";
```

```

    cin>>bs>>ht;
    cout<<"Area of square is"<<area(s);
    cout<<"\nArea of rectangle is "<<area(l,b);
    cout<<"\nArea of circle is "<<area(r);
    cout<<"\nArea of triangle is "<<area(bs,ht);
}
int area(int s)
{
    return(s*s);
}
int area(int l,int b)
{
    return(l*b);
}
float area(float r)
{
    return(3.14*r*r);
}
float area(float bs,float ht)
{
    return((bs*ht)/2);
}

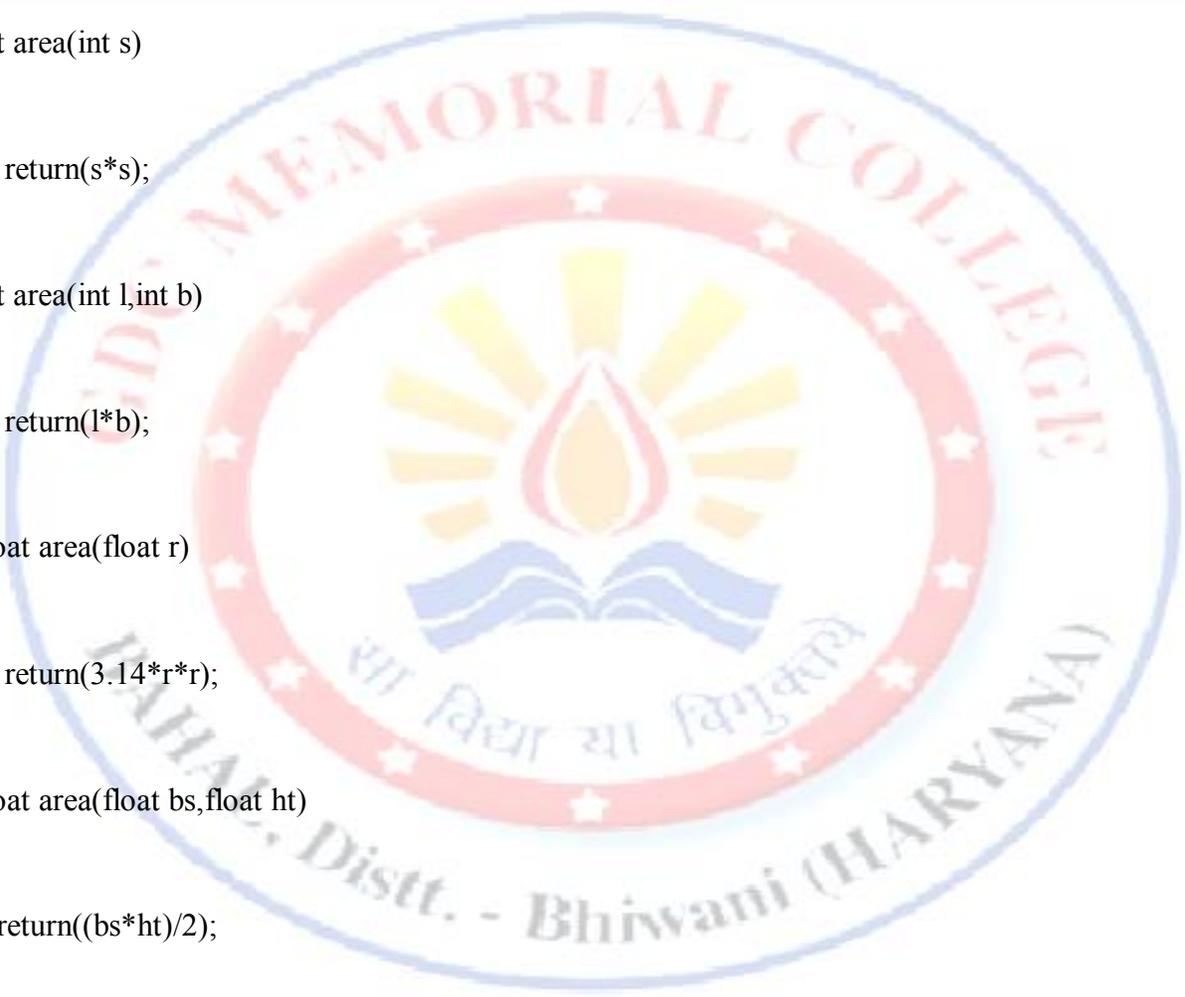
```

/*PROGRAM 6: WAP TO CONCEPT TO STATIC CLASS MEMBER*/

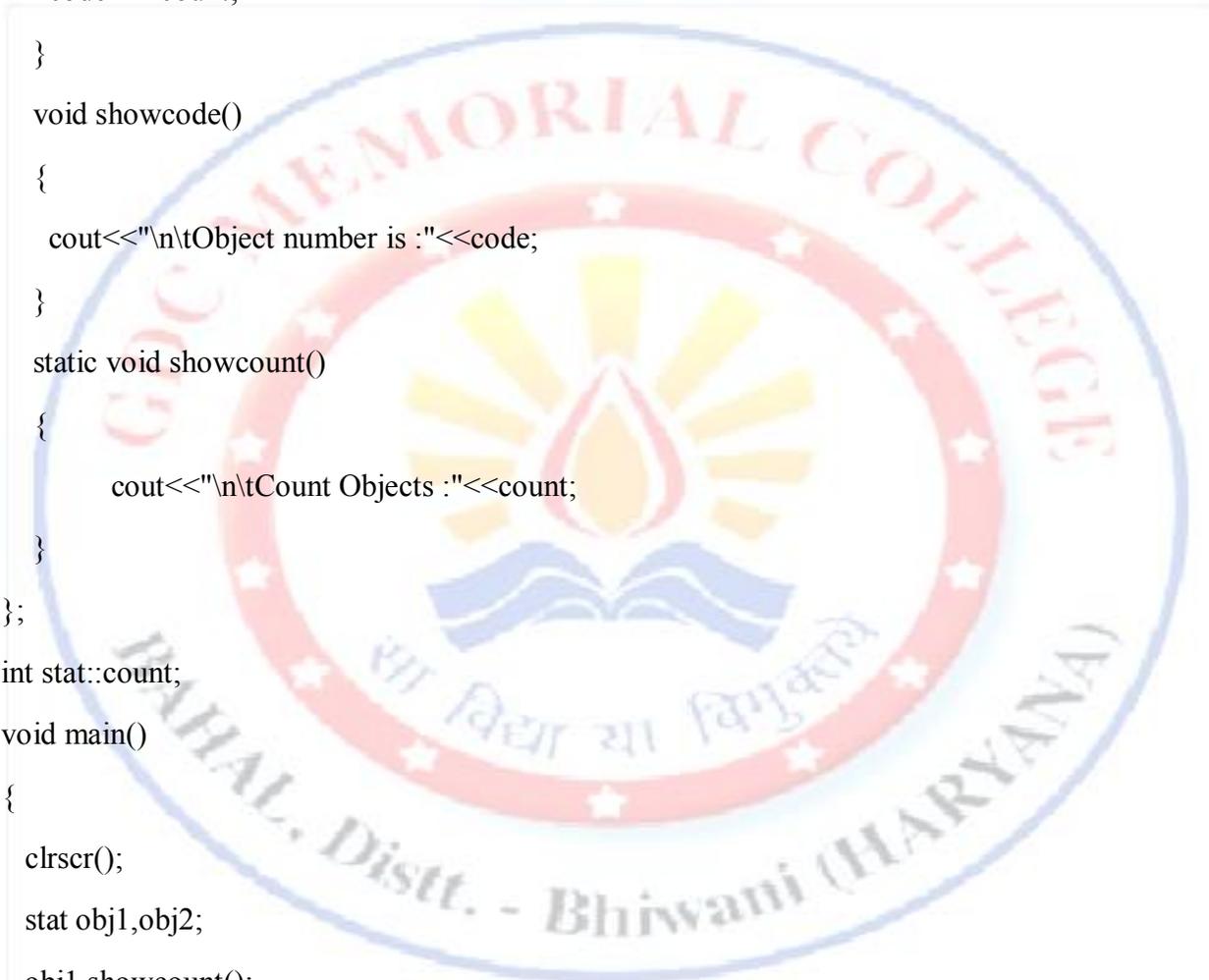
```

#include<iostream.h>
#include<conio.h>
class stat
{

```



```
int code;
static int count;
public:
stat()
{
code=++count;
}
void showcode()
{
cout<<"\n\tObject number is :"<<code;
}
static void showcount()
{
cout<<"\n\tCount Objects :"<<count;
}
};
int stat::count;
void main()
{
clrscr();
stat obj1,obj2;
obj1.showcount();
obj1.showcode();
obj2.showcount();
obj2.showcode();
getch();
}
```



/*PROGRAM 7: WAP TO ILLUSTRATE OBJECT AS FUNCTION ARGUMENT*/

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

```
class time
```

```
{
```

```
int hours;
```

```
int minutes;
```

```
public:
```

```
void gettime(int h,int m)
```

```
{
```

```
hours= h;
```

```
minutes= m;
```

```
}
```

```
void puttime(void)
```

```
{
```

```
cout <<hours<<"hours ";
```

```
cout<<minutes <<"minutes"<<"\n";
```

```
}
```

```
void sum(time,time);
```

```
};
```

```
void time::sum (time t1 ,time t2)
```

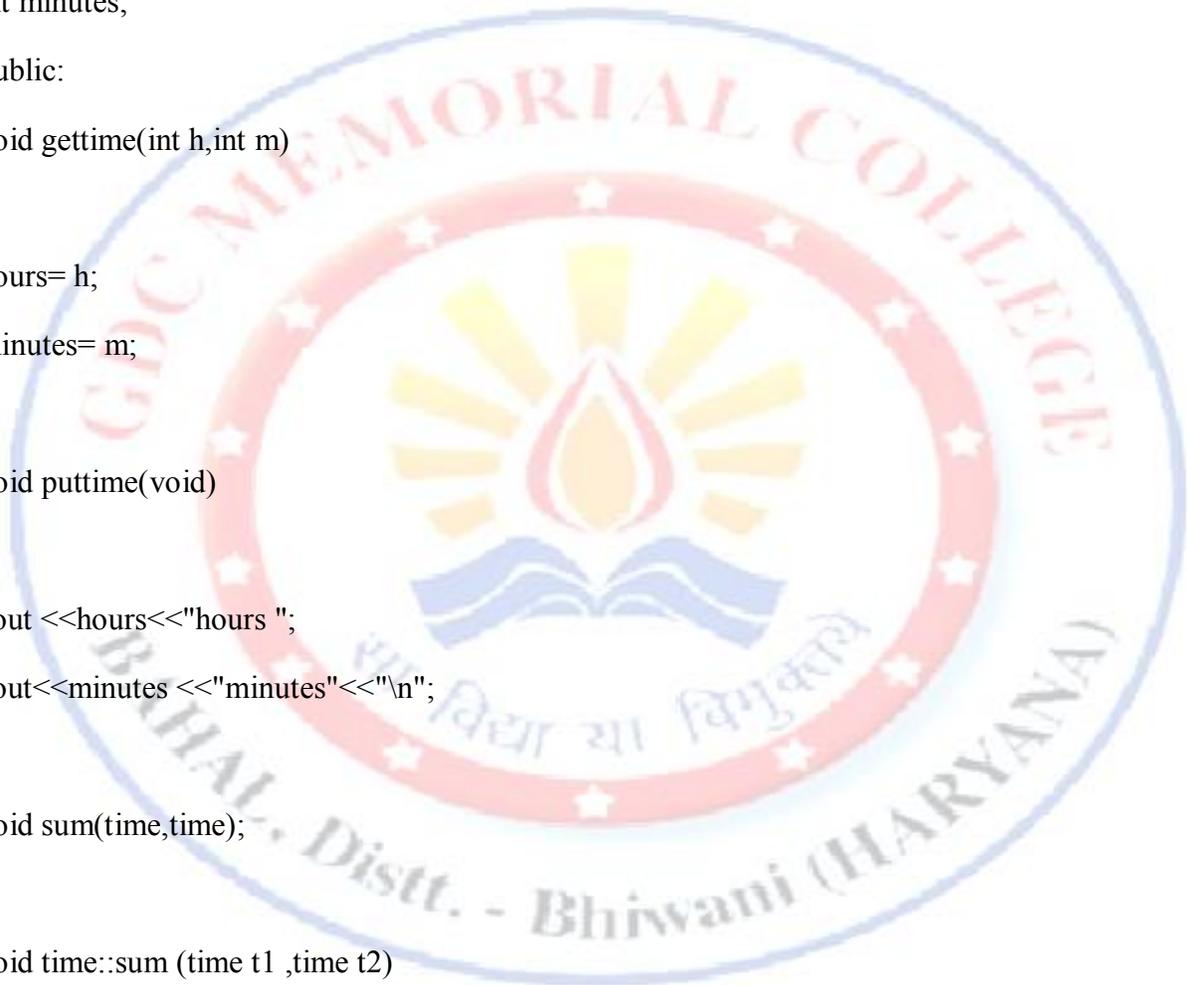
```
{minutes = t1.minutes+t2.minutes;
```

```
hours = minutes/60;
```

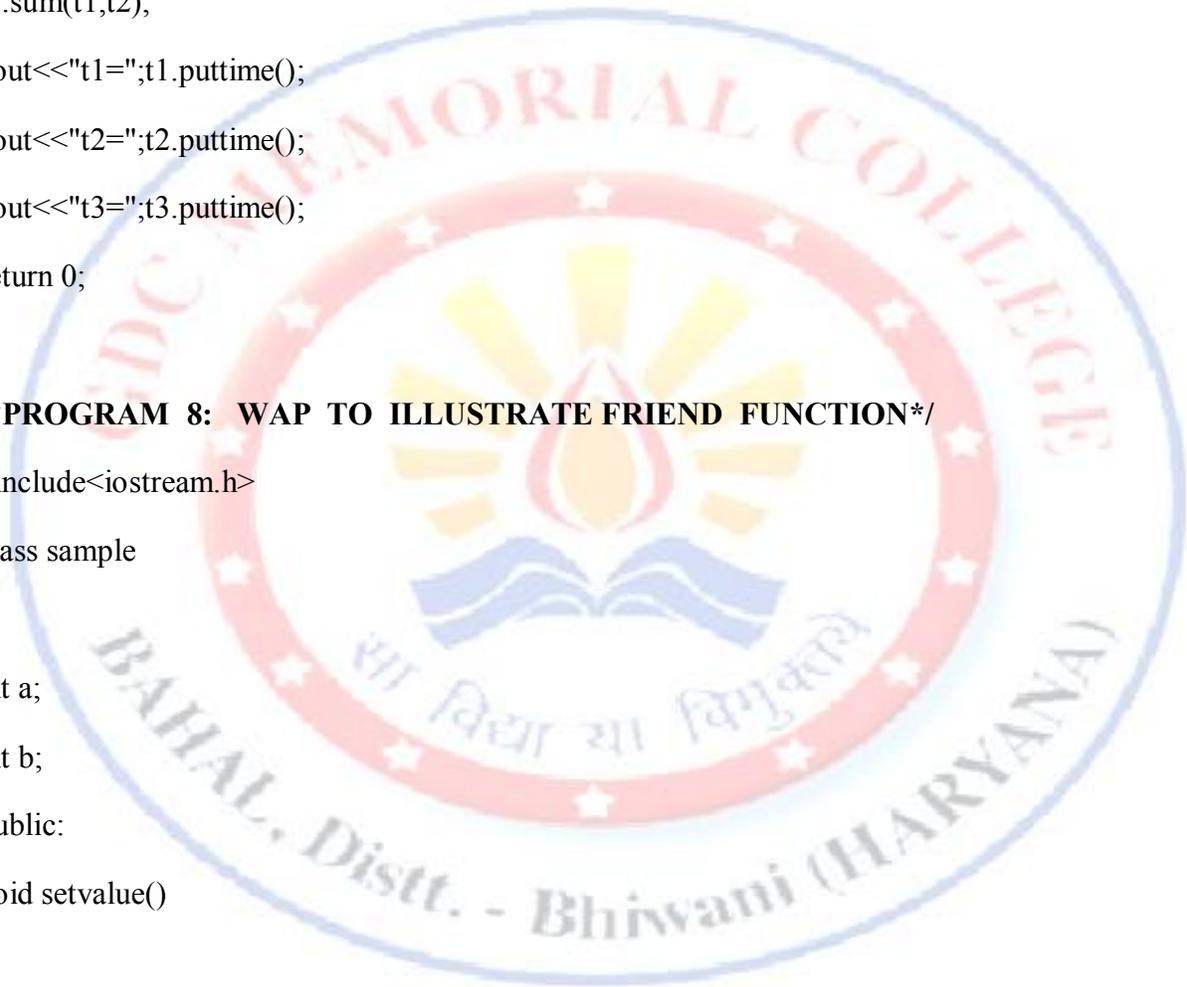
```
minutes = minutes%60;
```

```
hours+=t1.hours+t2.hours;
```

```
}
```



```
int main()
{
time t1,t2,t3;
t1.gettime(3,6.7);
t2.gettime(4,40 );
t3.sum(t1,t2);
cout<<"t1=";t1.puttime();
cout<<"t2=";t2.puttime();
cout<<"t3=";t3.puttime();
return 0;
}
/*PROGRAM 8: WAP TO ILLUSTRATE FRIEND FUNCTION*/
#include<iostream.h>
class sample
{
int a;
int b;
public:
void setvalue()
{
a=25;
b=67;
}
friend float mean(sample s);
};
```



```
float mean(sample s){
return float(s.a+s.b)/2.0;}

int main()
{sample x;
x.setvalue();
cout<<"\nMean Value ="<<mean(x)<<"\n";
return 0;
}
```

/* PROGRAM 9: WAP THAT IMPLEMENTS MULTIPLE CONSTRUCTORS IN A CLASS.*/

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

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

```
class MyClass
```

```
{
```

```
public:
```

```
int x;
```

```
int y;
```

```
MyClass()
```

```
{
```

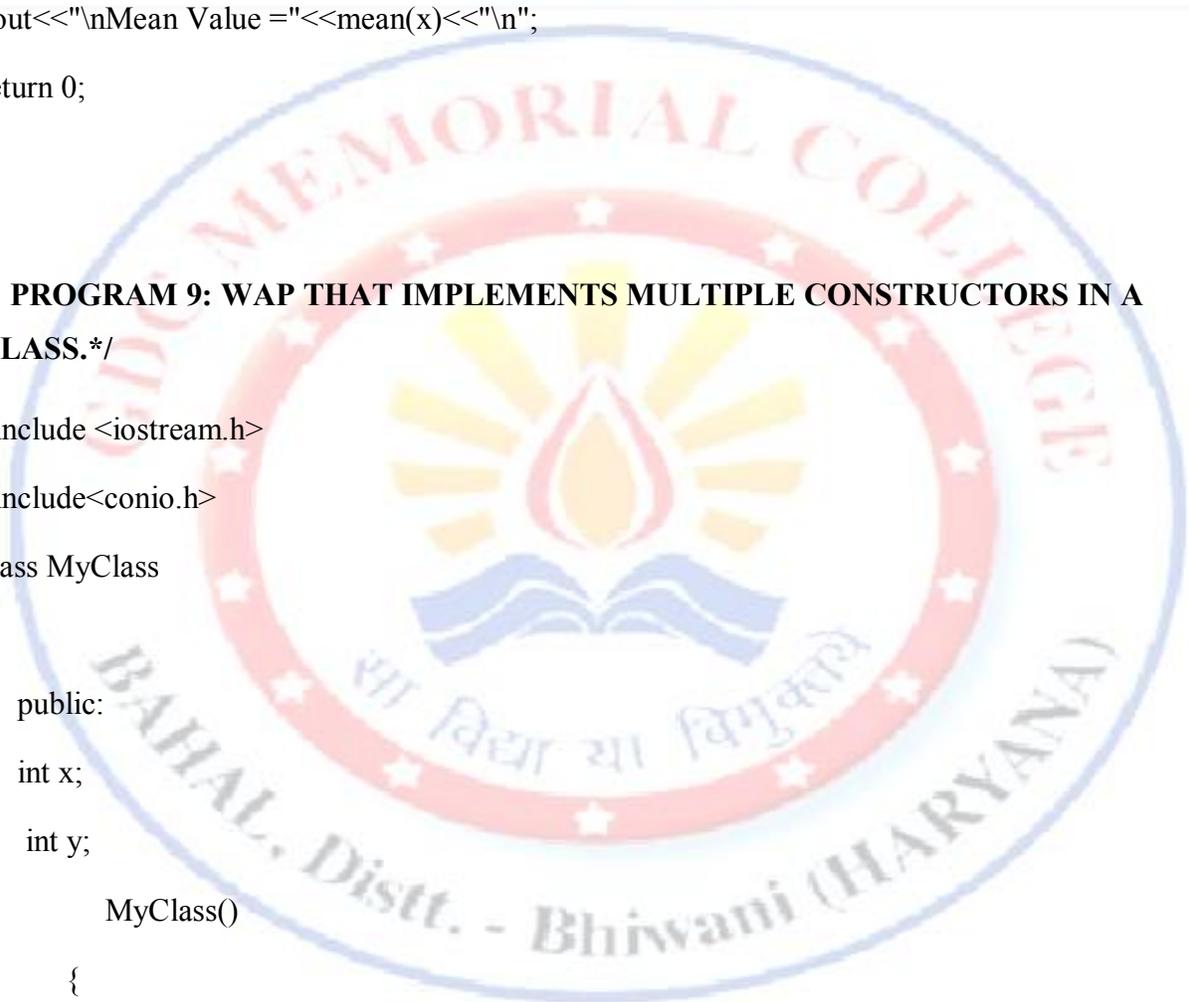
```
x = y = 0;
```

```
}
```

```
MyClass(int i)
```

```
{
```

```
x=y= i;
```



```

    }
    MyClass(int i, int j)
    {
        x=i;
        y=j;
    }
};

```

```
void main()
```

```

{
    clrscr();
    MyClass t;
    MyClass t1 (5);
    MyClass t2(9, 10);
    cout<< "t.x: " << t.x<< ", t.y: " <<t.y<< "\n";
    cout << "t1.x: " << t1.x << ", t1.y: " << t1.y << "\n";
    cout << "t2.x: " << t2.x << ", t2.y: " <<t2.y << "\n";
    getch();
}

```

/*PROGRAM 10: WAP THAT OVERLOAD UNARY OPERATOR USING FRIEND FUNCTION */

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

```
class xyz{
```

```
    int x;
```

```
    int y;
```

```
    int z;
```

```

public:
    void get(){
        cout<<"XYZ \n";
        cin>>x>>y>>z;
    }
    void disp(){
        cout<<x<<y<<z<<endl;
    }
    void friend operator-(xyz &S);
}s1;

void operator-(xyz &S){
S.x=-S.x;
S.y=-S.y;
S.z=-S.z;
}

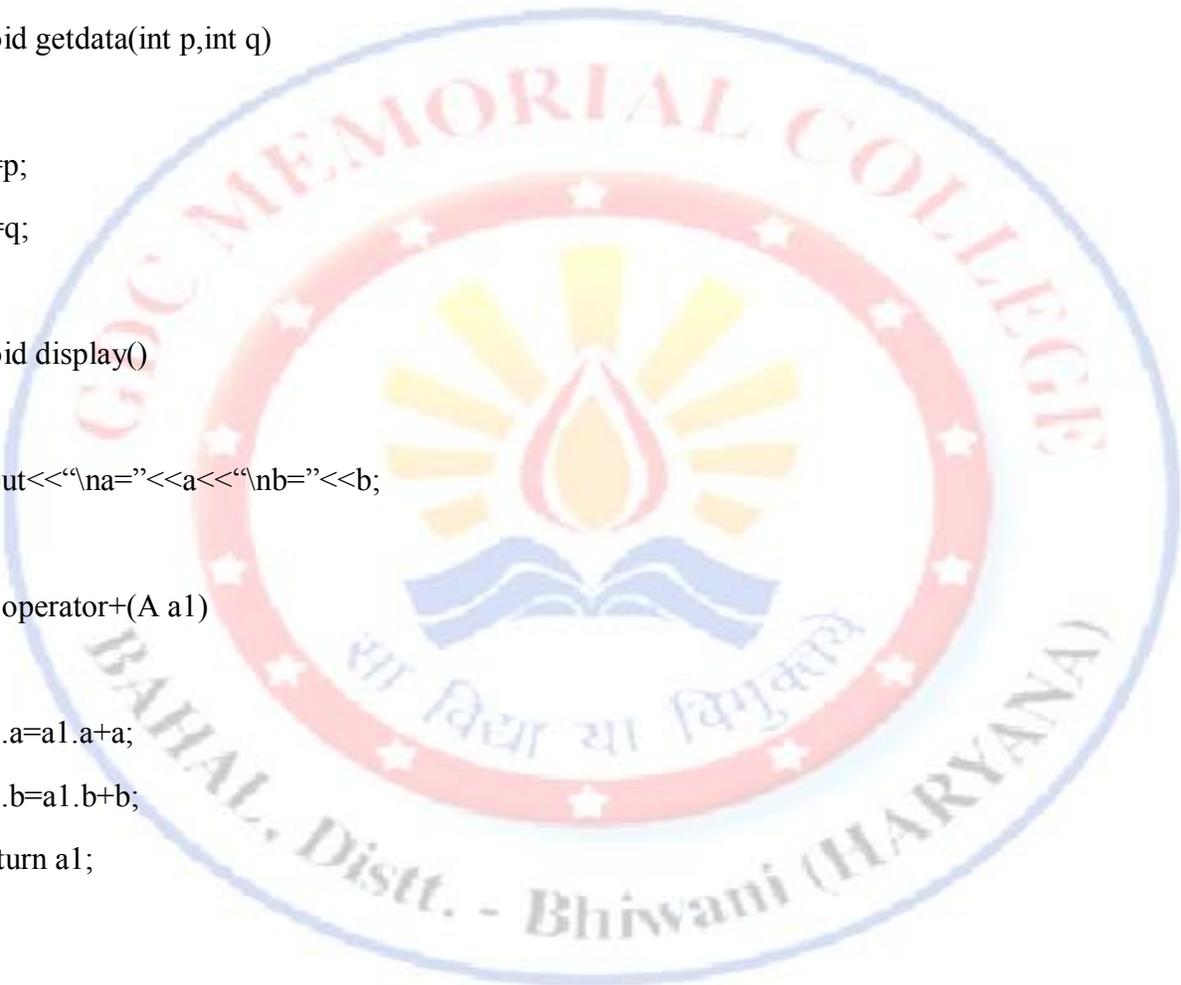
int main(){
    s1.get();
    cout<<"BEFORE OVERLOADING\n";
s1.disp();
    cout<<"AFTER OVERLOADING \n";
    -s1;

s1.disp();
return 0;
}

/*PROGRAM 11: WAP THAT OVERLOAD BINARY OPERATOR USING MEMBER
FUNCTION */

```

```
#include<conio.h>
#include<iostream.h>
class A
{
int a,b;
public:
void getdata(int p,int q)
{
a=p;
b=q;
}
void display()
{
cout<<"a="<<a<<"\nb="<<b;
}
A operator+(A a1)
{
a1.a=a1.a+a;
a1.b=a1.b+b;
return a1;
}
};
int main()
{
A a1,a2,a3;
int f,g,h,i;
cout<<"\nEnter 2 numbers\n";
```



```

cin>>f>>g;
a1.getdata(f,g);
a1.display();
cout<<"\nEnter 2 numbers\n";
cin>>h>>i;
a2.getdata(h,i);
a3.display();
a3=a2+a1;
getch();
return 0;
}

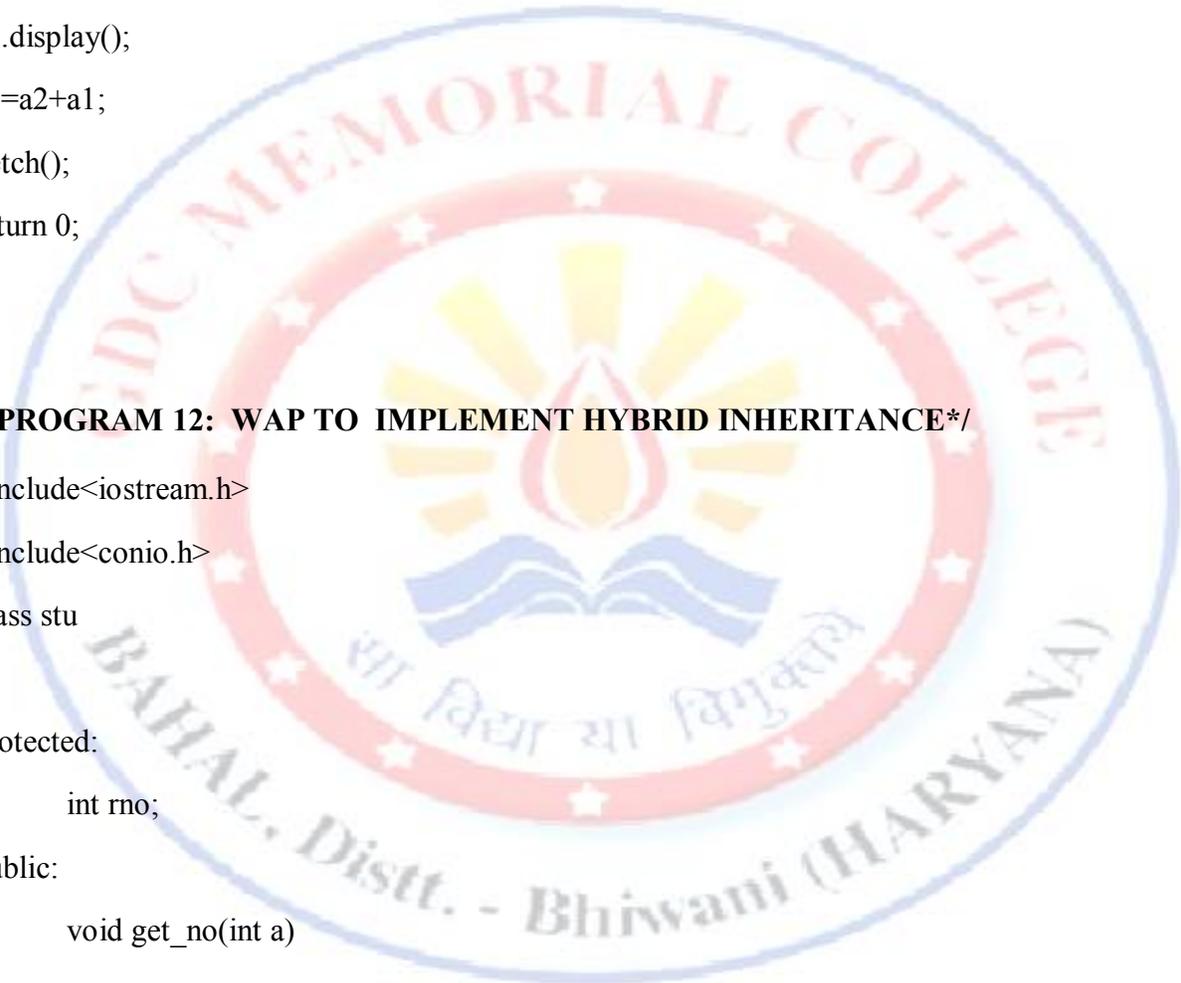
```

/*PROGRAM 12: WAP TO IMPLEMENT HYBRID INHERITANCE*/

```

#include<iostream.h>
#include<conio.h>
class stu
{
protected:
    int rno;
public:
    void get_no(int a)
    {
        rno=a;
    }
    void put_no(void)
    {
        out<<"Roll no"<<rno<<"\n";

```



```
    }  
};  
class test:public stu  
{  
    protected:  
    float part1,part2;
```

```
public:
```

```
void get_mark(float x,float y)
```

```
{  
    part1=x;  
    part2=y;  
}
```

```
void put_marks()
```

```
{  
    cout<<"Marks obtained:"<<"part1="<<part1<<"\n"<<"part2="<<part2<<"\n";  
}
```

```
};  
class sports
```

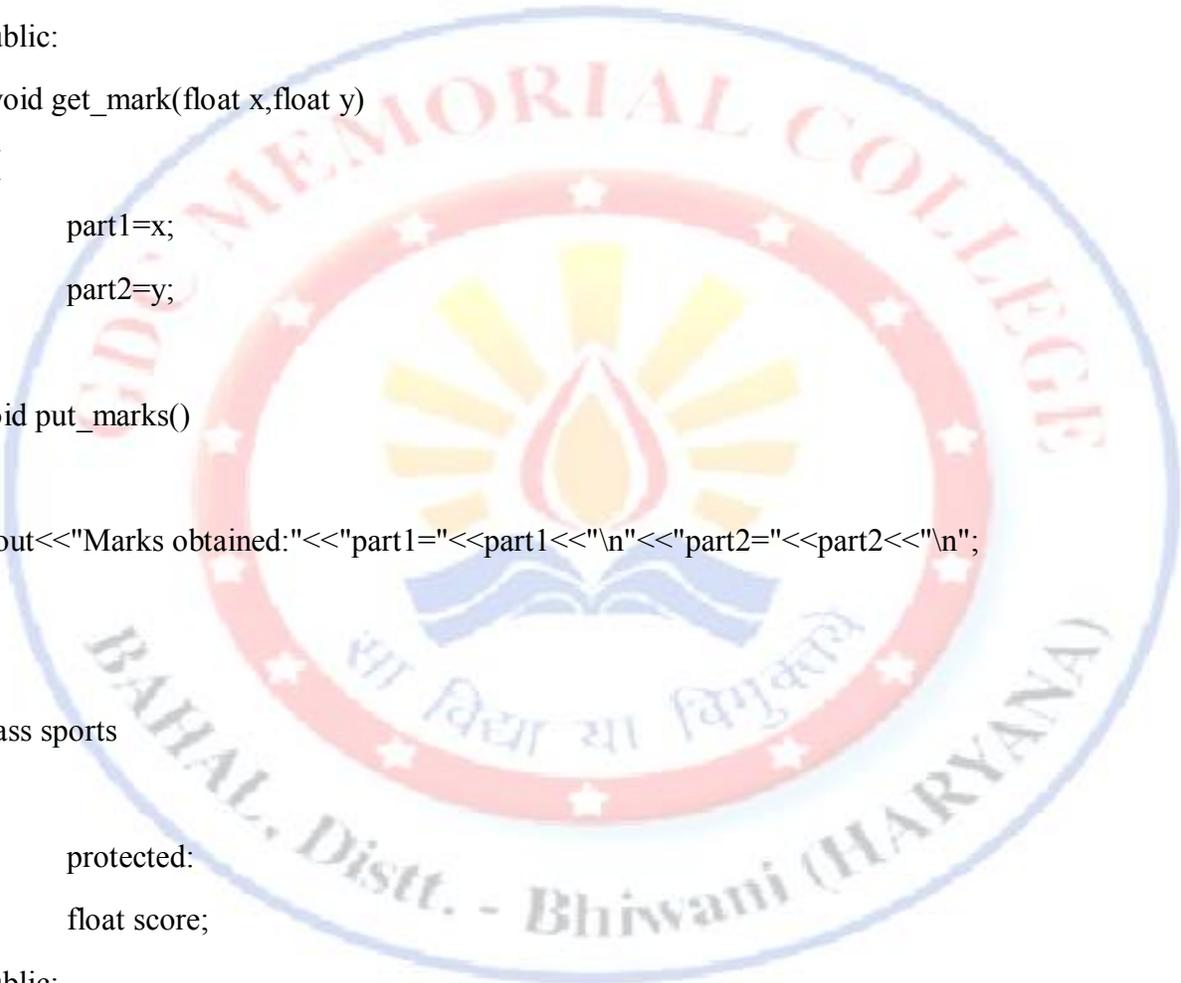
```
{  
    protected:  
    float score;
```

```
public:
```

```
void getscore(float s)
```

```
{  
    score=s;  
}
```

```
void putscore(void)
```



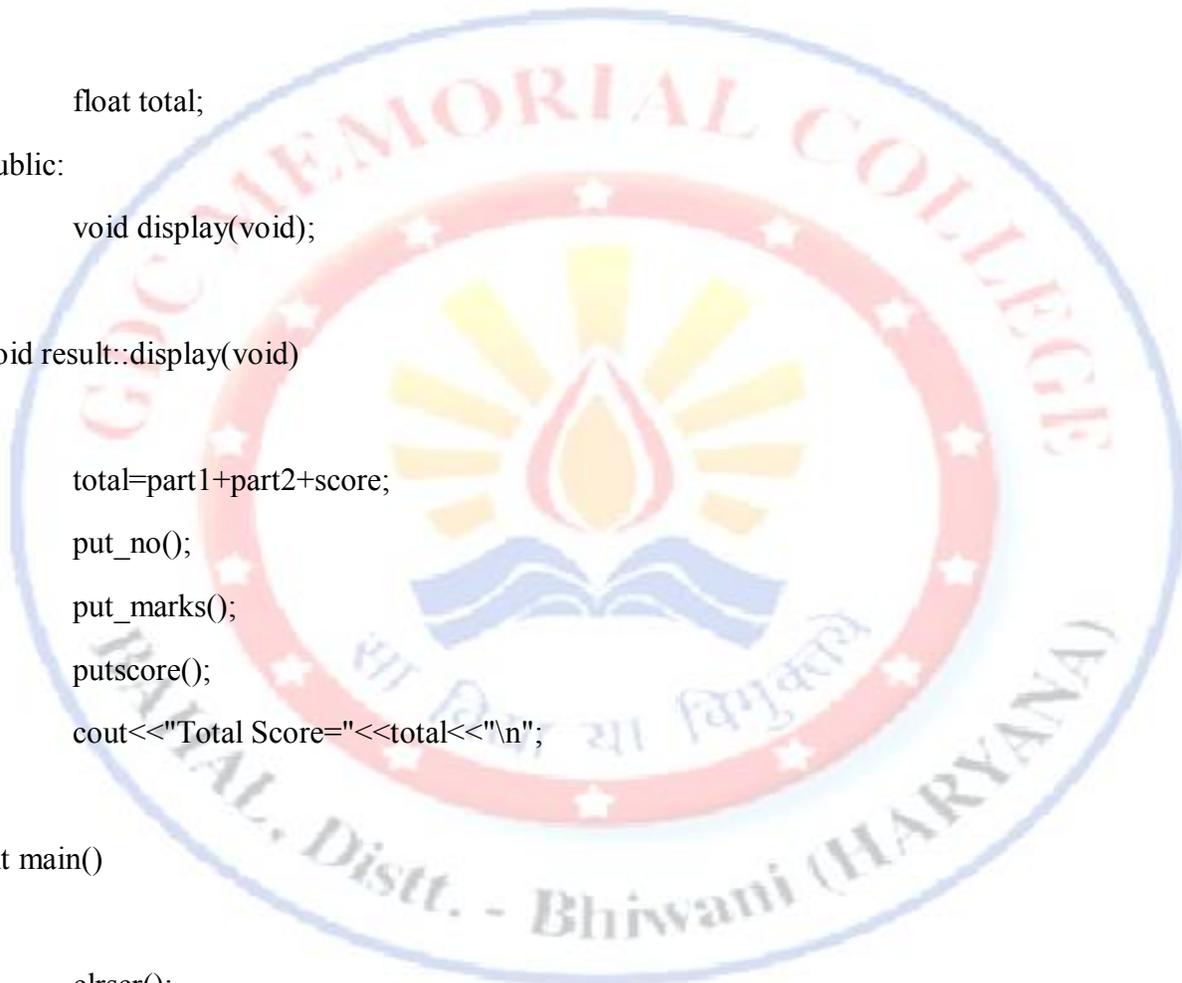
```
{  
    cout<<"sports:"<<score<<"\n";  
}  
};
```

```
class result: public test, public sports
```

```
{  
    float total;  
public:  
    void display(void);  
};  
void result::display(void)  
{  
    total=part1+part2+score;  
    put_no();  
    put_marks();  
    putscore();  
    cout<<"Total Score="<<total<<"\n";  
}
```

```
int main()
```

```
{  
    clrscr();  
    result stu;  
    stu.get_no(123);  
    stu.get_mark(27.5,33.0);  
    stu.getscore(6.0);  
    stu.display();  
}
```



```
    return 0;
}
```

/*PROGRAM 13: WAP TO SHOW THE USE OF VIRTUAL BASE CLASS */

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

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

```
class student
```

```
{
```

```
    int rno;
```

```
public:
```

```
void getnumber()
```

```
{
```

```
    cout<<"Enter Roll No:";
```

```
    cin>>rno;
```

```
}
```

```
void putnumber()
```

```
{
```

```
    cout<<"\n\n\tRoll No:"<<rno<<"\n";
```

```
}
```

```
};
```

```
class test:virtual public student
```

```
{
```

```
public:
```

```
int part1,part2;
```

```
void getmarks()
```

```
{
```

```
    cout<<"Enter Marks\n";
```

```

        cout<<"Part1:";
        cin>>part1;
        cout<<"Part2:";
        cin>>part2;
    }
void putmarks()
{
    cout<<"\tMarks Obtained\n";
    cout<<"\n\tPart1:"<<part1;
    cout<<"\n\tPart2:"<<part2;
}
};
class sports:public virtual student
{
public:
    int score;
    void getscore()
    {
        cout<<"Enter Sports Score:";
        cin>>score;
    }
    void putscore()
    {
        cout<<"\n\tSports Score is:"<<score;
    }
};
class result:public test,public sports

```



```

{
    int total;
public:
    void display()
    {
        total=part1+part2+score;

        putnumber();
        putmarks();
        putscore();
        cout<<"\n\tTotal Score:"<<total;
    }
};
void main()
{
    result obj;
    clrscr();
    obj.getnumber();
    obj.getmarks();
    obj.getscore();
    obj.display();
    getch();
}

```

/*PROGRAM 14: WAP TO ILLUSTRATE THE CONCEPT OF 'THIS' POINTER. */

```

#include <iostream.h>
class MyClass {
    int data;
public:

```



```

    MyClass() {data=100;};

    void Print1();

    void Print2();

};

void MyClass::Print1() {
    cout << data << endl;
}

void MyClass::Print2() {
    cout << "My address = " << this << endl;
    cout << this->data << endl;
}

int main()
{
    MyClass a;
    a.Print1();
    a.Print2();
    cout << sizeof(a) << endl;
}

```

/*PROGRAM 15: WAP TO ILLUSTRATE THE CONCEPT OF VIRTUAL FUNCTION*/

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

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

```
class base
```

```
{
```

```
    public:
```

```
        virtual void show()
```

```
{
    cout<<"\n Base class show:";
}
void display()
{
    cout<<"\n Base class display:" ;
}
};

class drive:public base
{
public:
    void display()
    {
        cout<<"\n Drive class display:";
    }
    void show()
    {
        cout<<"\n Drive class show:";
    }
};

void main()
{
    clrscr();
    base obj1;
    base *p;
```



```
cout<<"\n\t P points to base:\n" ;
```

```
p=&obj1;
```

```
p->display();
```

```
p->show();
```

```
cout<<"\n\n\t P points to drive:\n";
```

```
drive obj2;
```

```
p=&obj2;
```

```
p->display();
```

```
p->show();
```

```
getch();
```

```
}
```

