KONGU ENGINEERING COLLEGE
PERUNDURAI, ERODE - 638 052.

Aim & Algorithm (20) = 18	Name: Na.Poorrnaeubaehni
Program (40) = 40	ROLLNO: 15CSR133
Execution & Output (30) = 30	CLASS : CSE - C
Viva (10) = 8	SUBJECT: 14CSL32- Object Oriented Programming using C++ Laboratory
96	DATE: 4/11/16

96

1. Decrement the elements of a given matrix using unary operator overloading:

AIM: To write a C++ program to decrement the elements of a given matrix using unary operator overloading.

ALGORITHM:

Step1: Start.
Step2: Create a class named decre.
Step3: Within the class get the array elements and in display function inside class display the elements.
Step4: Then declare operator function as void operator --();
Step5: Define the operator function, by decrementing the matrix elements one by one.
Step6: In main, call the get function and display function.
Step7: Call the operator function by decrementing the object created for class (--d)
Step8: Finally display the matrix elements.
Step9: Stop.

PROGRAM:

```
#include<iostream>
using namespace std;
class decre
{
   int a[3][3],i,j;
   public:
   void get()
   {
   cout<<"\n Enter the
        matrix elements:";
for(i=0;i<3;i++)
{
for(j=0;j<3;j++)
{
 cin >>a[i][j];
 }
 }
 }
 void display()
 {
cout<<"\n The matrix elements
                are:";
for(i=0;i<3;i++)
{
for(j=0;j<3;j++)
{
cout<<a[i][j]<<"\t";
}
cout<<"\n";
}
 void operator --();
};
void decre :: operator --()
{
 for(i=0;i<3;i++)
 {
for(j=0;j<3;j++)
 {
  a[i][j] = --a[i][j];
 }
 }
}
main()
{
 decre d;
 d.get();
 cout<<"\n The matrix elements
 d.display();                \n";
cout<<"The decremented
matrix elements are:";
--d;
 d.display();
}
```

Output:

```
Enter the matrix elements: 1 2 3
                           4 5 6 7 8 9
the matrix elements are:
   1   2   3
   4   5   6
   7   8   9
```

1. Decrement the elements of a given matrix using unary operator overloading.
2. Write a C++ program to calculate area of a square, rectangle and triangle using Virtual function

The decremented matrix elements are:

0 1 2
3 4 5
6 7 8

2. Calculate area of a square, rectangle and triangle using Virtual function:

AIM: write a C++ program to
To calculate the area of square, rectangle and triangle using virtual function.

ALGORITHM:

Step 1: Start
Step 2: Create a base class shape and initialise the data members within get function and make the display function as virtual.
Step 3: In class square calculate the square area of square. and shape class is inherited as public.
Step 4: Similarly in class rectangle and triangle calculate its area and shape base class is inherited as public.
Step 5: In main create a pointer object and using array of objects get and display the respective functions.
Step 6: Stop.

Output:

```
enter the side of square: 2
the area of square is: 4
enter the length and breadth of rectangle: 5 6
the area of rectangle is: 30
enter the breadth and height of the triangle: 0.5 6
the area of triangle is: 1.5
```

```
#include<iostream>
using namespace std;
class shape
{
protected
double s,t;
Public:
void get()
{
cin>>s>>t;
}
virtual void disp()=0;
};
class Square: public shape
{int a1;
public:
void disp()
{
a1=s*s;
cout<<"The area of square is:"<<a1;
}
};
class rectangle: public shape
{ int a2;
public:
void disp()
{
a2=s*t;
cout<<"The area of rectangle is:"<<a2;
}
};
class triangle: public shape
{
float a3;
public:
void disp()
{
a3=0.5*s*t;
cout<<"The area of triangle is:"<<a3;
}
};
main()
{
Shape *a[3];
Square sq;
rectangle rec;
triangle tri;
a[0]=&sq;
a[1]=&rec;
a[2]=&tri;
cout<<"Enter the side of square:";
a[0]->get();
a[0]->disp();
cout<<"Enter the length and breadth of rectangle:";
a[1]->get();
a[1]->disp();
cout<<"Enter the breadth & height of triangle:";
a[2]->get();
a[2]->disp();
}
```