

program : Write a c / C++ program to accept  
a number and display its square

```
#include <iostream.h>
#include <conio.h>
void main ()
{
    clrscr();
    int x,y;
    cout << "Enter a number.";
    cin >> x;
    y = x * x;
    cout << "The square of " << x << " is " << y;
    getch();
}
```

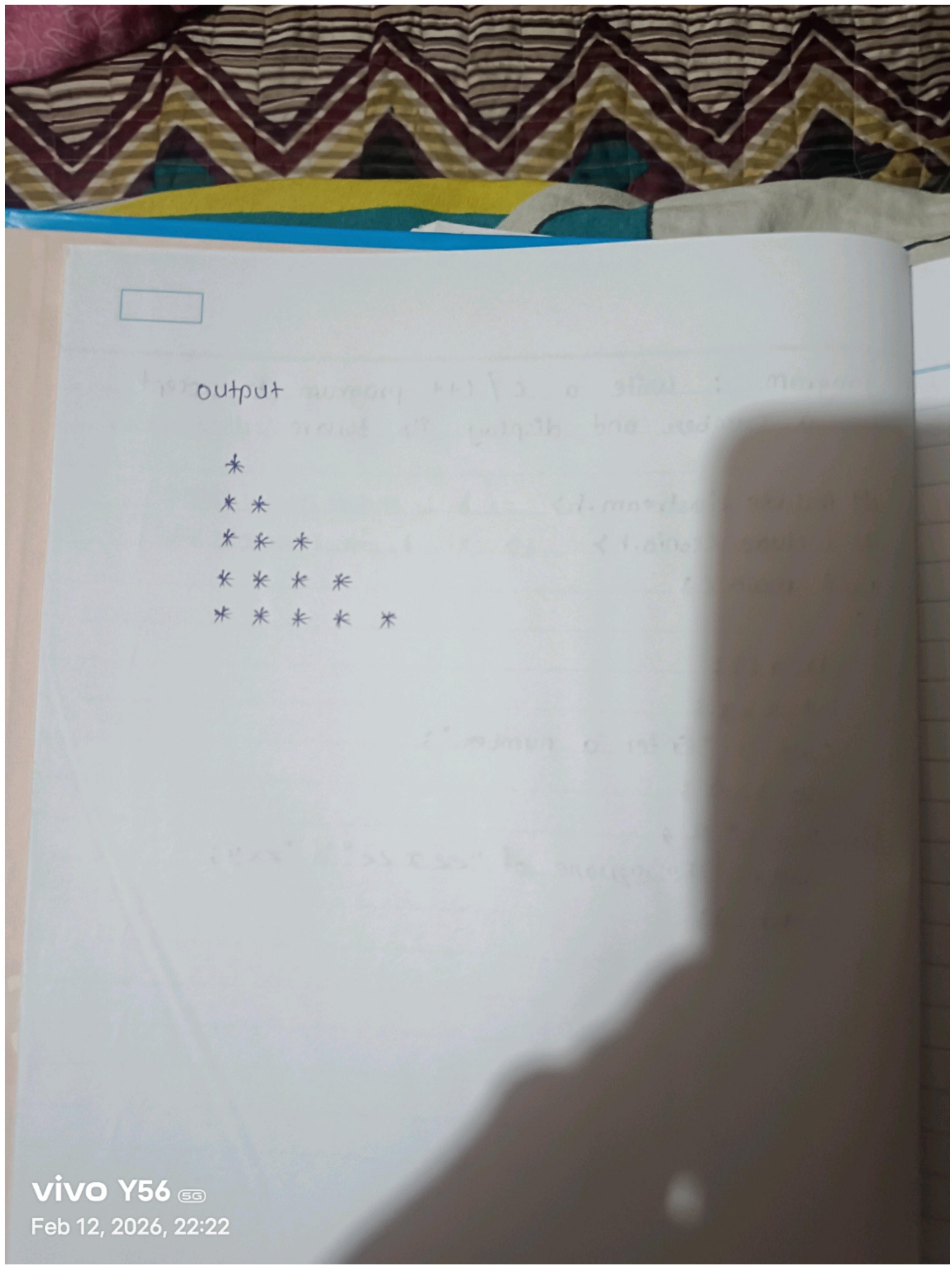
Output

Enter a number : 6

The square of 6 is 36

program : Write a program to display the following :

```
#include <iostream.h>
#include <conio.h>
void main ()
{
    clrscr();
    int i, j;
    for (i=1; i<=5; i++)
    {
        for (j=1; j<=i; j++)
        {
            cout << "*";
        }
        cout << endl;
    }
    getch();
}
```

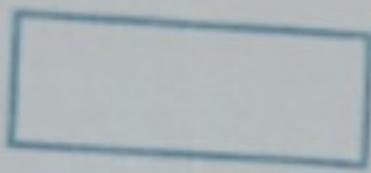


vivo Y56 5G

Feb 12, 2026, 22:22

program : Write a program to check if the entered number is prime number or not

```
#include <iostream.h>
#include <conio.h>
Void main ()
{
    clrscr();
    int i=2, n;
    cout << "Enter a number : ";
    cin >> n;
    while (n % i != 0)
    {
        i++;
    }
    if (n == i)
    {
        cout << "prime Number ";
    }
    else
    {
        cout << "Not a prime number ";
    }
    getch();
}
```



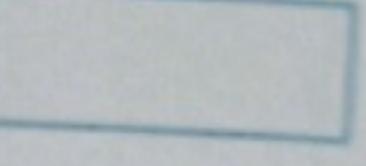
With regards to memory : memory  
Output

Enter a number : 17  
prime number

program : Write a program to display the user entered single digit number in word

```
#include <iostream.h>
#include <conio.h>
void main ()
{
    clrscr ();
    int n ;
    cout << "Enter a single digit number : ";
    cin >> n ;
    switch (n)
    {
        case 0 : cout << "zero" ;
        break ;
        case 1 : cout << "One" ;
        break ;
        case 2 : cout << "Two" ;
        break ;
        case 3 : cout << "Three" ;
        break ;
        case 4 : cout << "Four" ;
        break ;
        case 5 : cout << "Five" ;
        break ;
        case 6 : cout << "Six" ;
        break ;
    }
}
```

```
case 7 : cout << "seven";  
break;  
case 8 : cout << "Eight";  
break;  
case 9 : cout << "Nine";  
break;  
}  
getch();  
}
```



3.05.9 output at mapping no stdio : 1 number

for to medium smdq to medium b934ns

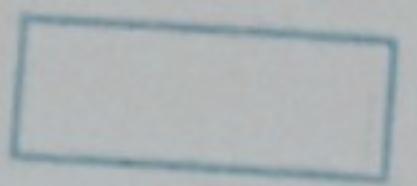
Enter a single digit number : 4

Four

program : write a program to add two numbers  
using a void function

```
#include <iostream.h>
#include <conio.h>
void main()
{
    clrscr();
    add();
    add(10);
    add(10, 20);
    getch();
}

void add (int a, int b)
{
    int c;
    c = a + b;
    cout << "sum = " << c << endl;
}
```



Output :

Sum = 11

Sum = 16

Sum = 30

program : Write a program to accept 'n' integers from user into an array and display them one in each line

```
#include <iostream.h>
#include <conio.h>
void main()
{
    clrscr();
    int n, i, a[100];
    cout << "Enter the number of elements:";
    cin >> n;
    for (i=0; i<=n-1; i++)
    {
        cout << "Enter a value:";
        cin >> a[i];
    }
    cout << "The numbers entered are " << endl;
    for (i=0; i<=n-1; i++)
    {
        cout << a[i] << endl;
    }
    getch();
}
```

[ ]

Output : ~~of displaying 6 rows & multiplying  
numbers block by block~~

Enter the number of elements : 5

Enter a value : 1 ~~(d. moorho) shilpi~~

Enter a value : 2 ~~(d. aino) shilpi~~

Enter a value : 3 ~~(d. nina) shilpi~~

Enter a value : 4 ~~(d. rani) shilpi~~

Enter a value : 5 ~~(d. anita) shilpi~~

The numbers entered are ' ~~(d. bho~~

1 ~~(d. lila) shilpi~~

2 ~~(d. ovi) shilpi~~

3 ~~(d. os) shilpi~~

4 ~~(d. gopi) shilpi~~

5 ~~(d. amita) shilpi~~

program : write a program to find area of circle  
using object oriented programming such that the class  
circle must have three member functions namely :

```
#include <iostream.h>
#include <conio.h>
class Circle
{
    float r,a;
public:
    void read()
    {
        cout << "Enter a radius : ";
        cin >> r;
    }
    void Compute()
    {
        a = 3.14 * r * r;
    }
    void display()
    {
        cout << "Area = " << a;
    }
};

void main()
{
    Circle();
}
```

circle c ;  
c. read ();  
c. compute ();  
c. display ();  
getch();

}



output

Enter radius = 5

Area = 78.5

Program : Write a program to find area of circle using object oriented programming such that the class Circle must have three inline functions namely

```
#include <iostream.h>
#include <conio.h>
class Circle
{
    float r, a;
public:
    void read();
    void compute();
    void display();
};

inline void Circle::read()
{
    cout << "Enter radius : ";
    cin >> r;
}

inline void Circle::compute()
{
    a = 3.14 * r * r;
}

inline void Circle::display()
{
    cout << "Area = " << a;
}
```

```
Void main ()  
{  
    clrscr();  
    circle c;  
    c.read ();  
    c.compute ();  
    c.display ();  
    getch ();  
}
```

Output

Enter radius : 5

Area = 78.5

program : Write a program to find area of circle using object Oriented programming. The value of rad accepted from the user in the constructor and the class circle must have two inline functions

```
#include <iostream.h>
#include <conio.h>
class circle
{
    float r, a;
public:
    Circle()
    {
        cout << "Enter the value of radius : ";
        cin >> r;
    }
    void compute();
    void display();
}
inline void circle::compute()
{
    a = 3.14 * r * r;
}
inline void circle::display()
{
    cout << "Area = " << a;
}
```

```
Void main ()  
{  
    clrscr ();  
    Circle c;  
    c. compute ();  
    c. display ();  
    getch ();
```

3

[Redacted]

Output

Enter the value of radius : 5

Area = 78.5

program : write a program to add two numbers using function overloading such that one function adds two integer second function adds two float numbers and the third function adds a float number with an integer

```
#include <iostream.h>
#include <conio.h>
float add (float a, int b)
{
    float c;
    c = a + b;
    return c;
}
int add (int a, int b)
{
    int c;
    c = a + b;
    return c;
}
float add (float a, float b)
{
    float c;
    c = a + b;
    return c;
}
void main ()
```

{

```
clrscr();  
int x, a=5, b=6;  
float y, p=3.5, q=6.6;  
x = add (a,b);  
cout << "Sum = " << x << endl;  
y = add (p,q);  
cout << "Sum = " << y << endl;  
Y = add (p,a);  
cout << "Sum = " << Y << endl;  
getch();
```

}

[ ]

Output :

Sum = 11

Sum = 10.1

Sum = 8.5

program : Write a program to negate (unary operator overloading) the values of two variables contained in an

```
#include <iostream.h>
#include <conio.h>
class Negate
{
    int x, y;
public:
    void read()
    {
        cout << "Enter two numbers";
        cin >> x >> y;
    }
    void Compute()
    {
        x = -x;
        y = -y;
    }
    void display()
    {
        cout << "x = " << x << endl << "y = " << y;
    }
}
void main()
```

clrscr();  
Negate n;  
n.read();  
n.compute();  
n.display();  
getch();

3

output

Enter two numbers : 4

6

$x = -4$

$y = -6$

program : Write a program to overload unary operators ++ (increment) and -- (decrement)

```
#include <iostream.h>
#include <conio.h>
class IncDec
{
    int x, y;
public:
    Void read ()
    {
        cout << "Enter two numbers ";
        cin >> x >> y;
    }
    Void operator --()
    {
        x--;
        y--;
    }
    Void operator ++()
    {
        x++;
        y++;
    }
    Void display ()
    {
        cout << "x = " << x << endl << "y = " << y << endl;
    }
}
```

3;

void main ()  
{

clrscr();

IncDec n;

n.read();

-- n;

cout << "After decrementing the object one time \n";

n.display();

++n;

++n;

cout << "After incrementing the object twice \n";

n.display();

getch();

3

Output

Enter two numbers 4

5

After decrementing the object one time

x = 3

y = 4

After incrementing the object twice

x = 5

y = 6

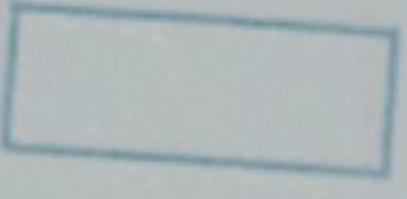
program : Write a program to find how many objects of a class has been created using static member function

```
#include <iostream.h>
#include <conio.h>
class Count
{
private :
    static int counter;
public :
    Count()
    {
        Counter++;
    }
    static int display()
    {
        return Counter;
    }
};

int Count::Counter = 0;
void main()
{
    clrscr();
    Count c1;
    cout << "Number of objects:" << c1.display() << endl;
```

```
Count c2;  
Count c3;  
cout << "Number of objects :" << c1.display() << endl;  
getch();
```

3



output

Number of object : 1

Number of object : 3

program :

```
#include <iostream.h>
#include <conio.h>
class Data
{
protected:
    int a, b;
public:
    void read()
    {
        cout << "Enter two numbers";
        cin >> a >> b;
    }
};

class Sum : public Data
{
private:
    int sum;
public:
    void add()
    {
        sum = a + b;
    }
    void display()

```

```
{  
    cout << "The sum is " << sum;  
}  
};  
void main()  
{  
    clrscr();  
    sum s;  
    s.read();  
    s.add();  
    s.display();  
    getch();  
}
```

Output

Enter two numbers 4

5

The sum is 9

Program : Write a program to calculate percentage of a student using multi level inheritance. The base class

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

```
class Data
```

```
{
```

```
protected :
```

```
int p, c, m;
```

```
public :
```

```
void read ()
```

```
{
```

```
cout << "Enter the marks obtained in physics,  
chemistry and Maths ";
```

```
cin >> p >> c >> m;
```

```
}
```

```
};
```

```
class Sum : public Data
```

```
{
```

```
protected :
```

```
int total;
```

```
public :
```

```
void sum ()  
{  
    total = p + c + m ;  
}  
};  
class percent : public sum  
{  
private :  
float percent ;  
public :  
void calculate ()  
{  
    percent = total / 300.0 * 100 ;  
}  
void display ()  
{  
    cout << "The percentage is " << percent ;  
}  
};  
void main ()  
{  
clrscr ();  
percent a ;  
a.read ();  
a.sum ();  
a.calculate ();  
a.display ();  
getch ();
```

program : Write a program to define the following relationship using multiple inheritance

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

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

```
class polygon
```

```
{
```

```
protected :
```

```
int height, width ;
```

```
public :
```

```
void read (int a, int b)
```

```
{
```

```
height = a ;
```

```
width = b ;
```

```
}
```

```
};
```

```
class Output
```

```
{
```

```
public :
```

```
void output (int x)
```

```
{
```

```
cout << "Area is " << x ;
```

```
}
```

```
};
```

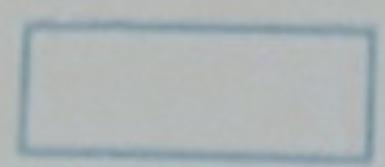
```
class Rectangle : public Polygon, public Output
```

```
{
```

```
public :
```

```
int area ()  
{  
    return (height * width);  
}  
};  
  
class Triangle : public polygon, public Output  
{  
public:  
    int area ()  
    {  
        return (height * width / 2);  
    }  
};  
  
void main ()  
{  
    clrscr();  
    int h, w, choice, a;  
    cout << "1. Area of Rectangle \n 2. Area of Triangle \n";  
    Enter your choice: "  
    cin >> choice;  
    cout << "Enter height and width: "  
    cin >> h >> w;  
    switch (choice)  
    {  
        Case 1:  
        Rectangle r;  
        r.read(h, w);
```

```
a = t.area();  
t.output(a);  
break;  
case 2:  
Triangle t;  
t.read(h,w);  
a = t.area();  
t.output(a);  
break;  
default: cout << "Invalid Choice";  
}  
getch();
```



output

I. Area of Rectangle

II. Area of Triangle

Enter your choice : 2

Enter height and width : 5

4

Area is 10

program : Write a program to add two complex numbers using operator overload by a friend function

```
#include <iostream.h>
#include <conio.h>
class Complex
{
    int x, y;
public:
    void read()
    {
        cout << "Enter the real and imaginary parts of a
complex number: ";
        cin >> x >> y;
    }
    friend Complex operator + (Complex c1, Complex c2);
    void display()
    {
        if (y < 0)
            cout << x << y << "i";
        else
            cout << x << "+i" << y;
    }
};
```

Complex operator + (Complex c1, Complex c2)

```
Complex c;
```

```
c.x = c1.x + c2.x;
```

```
c.y = c1.y + c2.y;
```

```
return c;
```

```
}
```

```
void main()
```

```
{
```

```
clrscr();
```

```
Complex c1;
```

```
Complex c2;
```

```
Complex c3;
```

```
c1.read();
```

```
c2.read();
```

```
c3 = c1 + c2;
```

```
c3.display();
```

```
getch();
```

```
}
```

output  
Enter the real and imaginary parts of a  
Complex number : 2

3  
Enter the real and imaginary parts of a  
Complex number : 4

5  
 $6 + 1i$

program  
numbe  
functio

# inclu  
# inclu  
class Co  
E  
int \*  
publ  
void  
P

Complex

program : Write a program to demonstrate dynamic binding using virtual function

```
#include <iostream.h>
#include <conio.h>
class Base
{
protected:
    int a, b;
public:
    virtual void read()
    {
        cout << "Enter two values : ";
        cin >> a >> b;
    }
    virtual void display()
    {
        cout << "\nThe values are :" << a << "\n" << b <<
    }
};

class Sub : public Base
{
protected:
    int c, d;
public:
    virtual void read()
```

{

```
cout << "Enter 4 values : ";
cin >> a >> b >> c >> d;
```

}

```
virtual void display ()
```

{

```
    cout << "\n The values are : " << a << "\n " << b
    << "\n " << c << "\n " << d << endl;
```

}

};

```
void main ()
```

{

```
clrscr ();
```

```
Base * ptr;
```

```
Base b;
```

```
Sub s;
```

```
ptr = &b;
```

```
ptr -> read ();
```

```
ptr -> display ();
```

```
ptr = &s;
```

```
ptr -> read ();
```

```
ptr -> display ();
```

```
getch ();
```

}

[ ]

output

Enter 2 values : 1

2

The values are : 1

2

Enter 4 values : 1

2

3

4

The values are : 1

2

3

4

prog

# i

<# i

class

E

end l ;

3;

class S

E

pr

in

pi

al