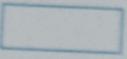


program : Write a c / c++ program to accept
a number and display its square

```
#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();
}
```

output summary for a study:
would be helpful here

*

**

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();
}
```

and
date
in

With regards of memory & efficiency : a major
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();  
}
```

9.68 In output of mapping is stored 2 integer
for randomizing single digit number

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 : ~~i = 92 > limit = 50~~

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 : at memory address 3 mb/4th
notable block is given

Enter the number of elements : 5

Enter a value : 1 (d, m, o, l) ; cout <<

Enter a value : 2 (d, m, o, l) ; cout <<

Enter a value : 3 (d, m, o, l) ; cout <<

Enter a value : 4 (d, m, o, l) ; cout <<

Enter a value : 5 (d, m, o, l) ; cout <<

The numbers entered are : (d, m, o, l)

1 (d, m, o, l)

2 (d, m, o, l)

3 (d, m, o, l)

4 (d, m, o, l)

5 (d, m, o, l)

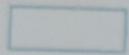
program : Write a program to find area of circle using object oriented programming such that the class circle must have three member functions namely :

```
#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()
{
    clrscr();
}
```

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

}



output as

Enter radius = 5

Area = 78.5

pr

1

U
E

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 ();

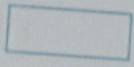
circle c ;

c. compute ();

c. display ();

getch();

3



Output

Enter the value of radius : 5

Area = 78.5

program : write a program to add two number 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 ()
```

2

```
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();
```

3

Output :

Sum = 11

Sum = 10.1

Sum = 8.5

pro
usi
two
and
an

In
In
float
{

P
C=

Y
int ad
{

int
C=

ret

B
float ad
{

float
C=a

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()
{
}
```

clscr();
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
ope

inclu

inclu

class

{

in-

pu

ve

{

3

ve

{

3

ve

{

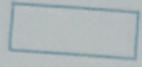
program : Write a program to find how many objects of a class has been created using static member function

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

3



output

Number of objects : 1

Number of objects : 3

• Monta con bordo sottopavimento refba 100x60

i C) 100x60

i C) 100x60

• Monta con bordo sottopavimento refba 100x60

i C) 100x60

i C) 100x60

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

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 :
```

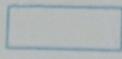
```
int area ()  
{  
    return (height * width);  
}  
};  
  
class Triangle : public polygon, public output  
{  
public:  
    int area ()  
    {  
        return (height * width / 2);  
    }  
};  
  
void main ()  
{  
    clrscr();  
    int h, w, choice, a;  
    cout << "1. Area of Rectangle \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.rod(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

pro

if p

if i

clas

E

3;

class

E

p

v

e

3;

class

E

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

3
frie
void
E

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 <<
    endl;
}
};

class Sub : public Base
{
protected:
    int c, d;
public:
    virtual void read()
    {
        cout << "Enter two values : ";
        cin >> c >> d;
    }
    virtual void display()
    {
        cout << "\nValues are : " << c << "\n" << d <<
    endl;
}
```

{

```
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

n

class

E

endl;

3;

class S

E

pr

in

P

v