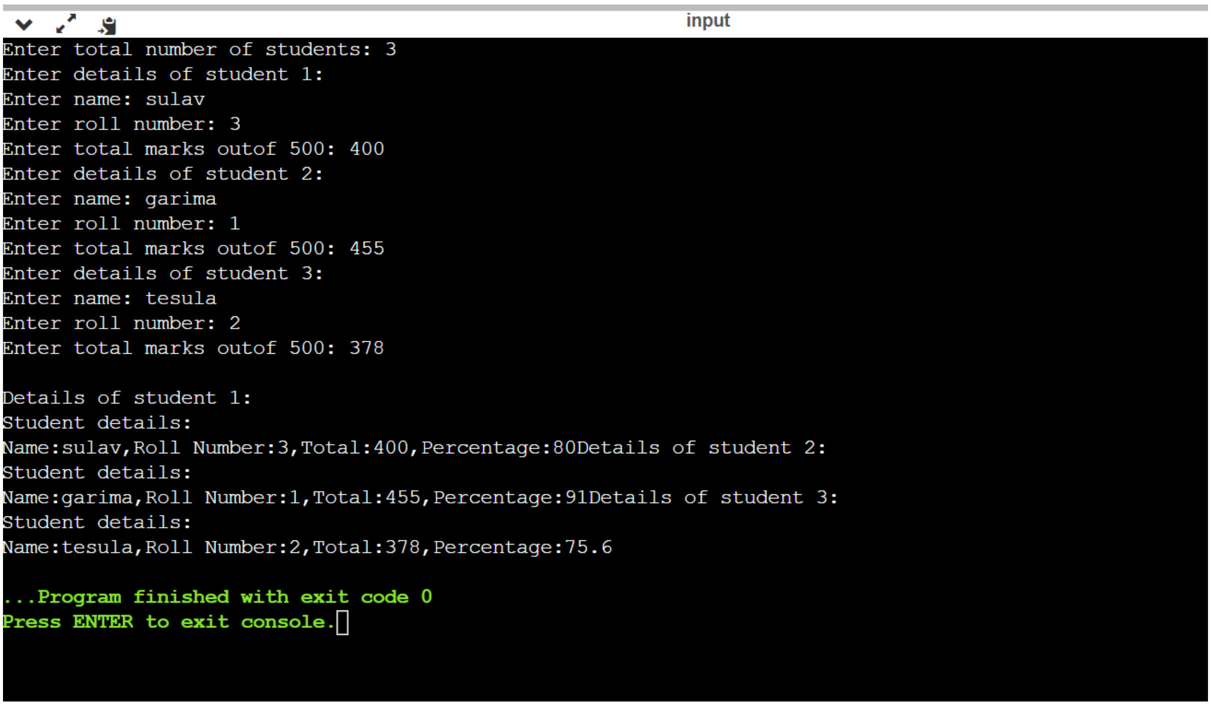
1. Write a C++ Program to display Names, Roll No., and grades of 3 students who have appeared in the examination. Declare the class of name, Roll No. and grade. Create an array of class objects. Read and display the contents of the array.

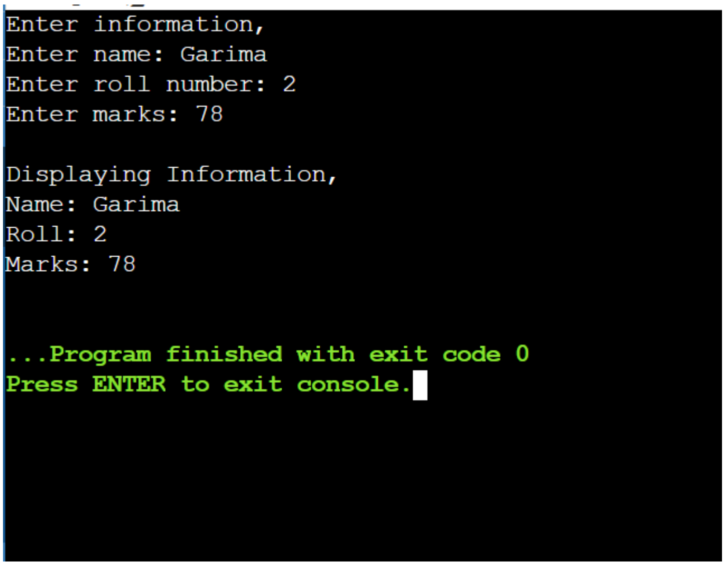
```
#include <iostream>
using namespace std;
#define MAX 10

class student  {
 private:
            char name[30];
            int   rollNo;
            int   total;
            float perc;
public:
             void getDetails(void);
            void putDetails(void);
};
void student:: getDetails(void) {
            cout << "Enter name: " ;
            cin >> name;
            cout << "Enter roll number: ";
            cin >> rollNo;
            cout << "Enter total marks out of 500: ";
            cin >> total;
            perc=(float)total/500*100;
 }
 void student:: putDetails(void) {
            cout << "Student details:\n";
            cout << "Name:"<< name << ",Roll Number:" << rollNo << ",Total:" << total <<
",Percentage:" << perc;
}

int main() {
            student std[MAX]; int n,loop;
            cout << "Enter total number of students: ";
            cin >> n;
            for (loop=0;loop< n; loop++) {
                        cout << "Enter details of student " << loop+1 << ":\n"; std[loop].getDetails();
            }
            cout << endl;
            for(loop=0;loop< n; loop++) {
                        cout << "Details of student " << (loop+1) << ":\n";
                        std[loop].putDetails();
             }
            return 0;
 }
```



```
Enter total number of students: 3
Enter details of student 1:
Enter name: sulav
Enter roll number: 3
Enter total marks outof 500: 400
Enter details of student 2:
Enter name: garima
Enter roll number: 1
Enter total marks outof 500: 455
Enter details of student 3:
Enter name: tesula
Enter roll number: 2
Enter total marks outof 500: 378

Details of student 1:
Student details:
Name:sulav,Roll Number:3,Total:400,Percentage:80Details of student 2:
Student details:
Name:garima,Roll Number:1,Total:455,Percentage:91Details of student 3:
Student details:
Name:tesula,Roll Number:2,Total:378,Percentage:75.6

...Program finished with exit code 0
Press ENTER to exit console.
```

2.Write a C++ program to declare Struct. Initialize and display contents of member variables.

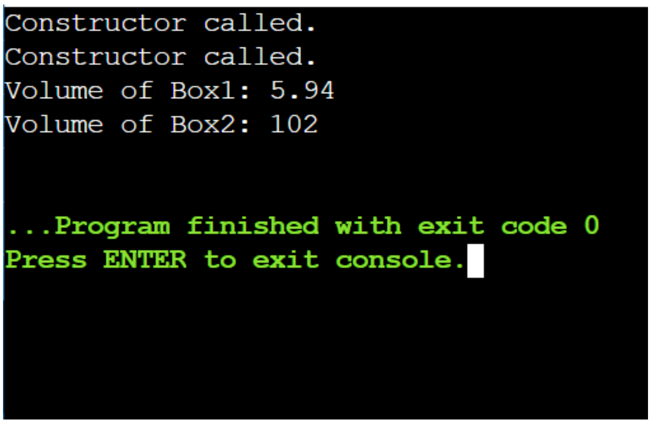
```
#include <iostream>
using namespace std;

struct student {
        char name[50];
        int roll;
        float marks;
};

int main()
{
        student s;
        cout << "Enter information," << endl;
        cout << "Enter name: ";
        cin >> s.name;
        cout << "Enter roll number: ";
        cin >> s.roll;
        cout << "Enter marks: ";
        cin >> s.marks;
        cout << "\nDisplaying Information," << endl;
        cout << "Name: " << s.name << endl;
        cout << "Roll: " << s.roll << endl;
        cout << "Marks: " << s.marks << endl;
        return 0;
}
```



```
Enter information,
Enter name: Garima
Enter roll number: 2
Enter marks: 78

Displaying Information,
Name: Garima
Roll: 2
Marks: 78


...Program finished with exit code 0
Press ENTER to exit console.
```

3. Write a C++ program to declare a class. Declare pointer to class. Initialize and display the contents of the class member.

```
#include <iostream>
using namespace std;

class Box {
  public:
        Box(double l = 2.0, double b = 2.0, double h = 2.0) {
                cout <<"Constructor called." << endl;
                length = l;
                breadth = b;
                height = h;
        }
        double Volume() {
                return length * breadth * height;
        }
        private:
                double length;
                double breadth;
                double height;
};

int main() {
  Box Box1(3.3, 1.2, 1.5);
  Box Box2(8.5, 6.0, 2.0);
  Box *ptrBox;
  ptrBox = &Box1;
  cout << "Volume of Box1: " << ptrBox->Volume() << endl; ptrBox = &Box2;
  cout << "Volume of Box2: " << ptrBox->Volume() << endl;
   return 0;
}
```

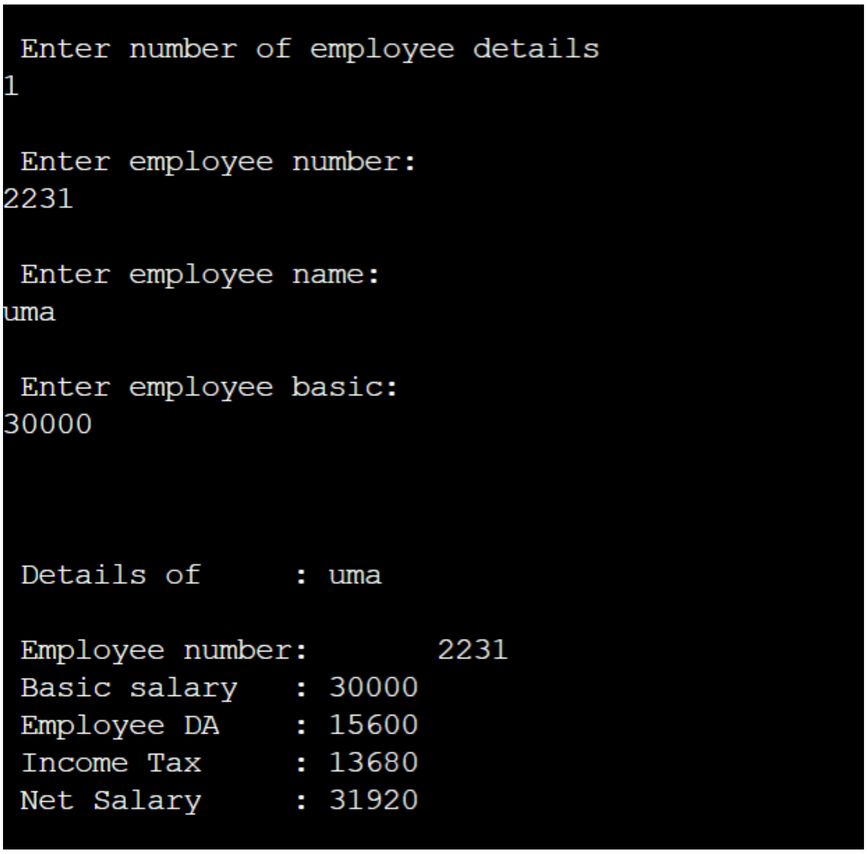


```
Constructor called.
Constructor called.
Volume of Box1: 5.94
Volume of Box2: 102


...Program finished with exit code 0
Press ENTER to exit console.
```

4.Given that an EMPLOYEE class contains following members: data members: Employee

```
#include<iostream>
using namespace std;

class employee {
        Int emp_num;
        char emp_name[20];
        float emp_basic;
        float sal;
        float emp_da;
        float net_sal;
        float emp_it;
public:
        void get_details();
        void find_net_sal();
        void show_emp_details();
};
void employee :: get_details() {
        cout<<"\n Enter employee number:\n";
        cin>>emp_num;
        cout<<"\n Enter employee name:\n";
        cin>>emp_name;
        cout<<"\n Enter employee basic:\n";
        cin>>emp_basic;
}
void employee :: find_net_sal() {
        emp_da=0.52*emp_basic;
        emp_it=0.30*(emp_basic+emp_da);
        net_sal=(emp_basic+emp_da)-emp_it;
}

void employee :: show_emp_details() {
        cout<<"\n\n\n Details of : "<<emp_name;
        cout<<"\n\n Employee number:        "<<emp_num;
        cout<<"\n Basic salary          : "<<emp_basic;
        cout<<"\n Employee DA           : "<<emp_da;
        cout<<"\n Income Tax : "<<emp_it;
        cout<<"\n Net Salary  : "<<net_sal;
}
int main() {
        employee emp[10];
        int i,num;
        cout<<"\n Enter number of employee details\n";
        cin>>num;
        for(i=0;i<num;i++)
                emp[i].get_details();
        for(i=0;i<num;i++)
```

```
                emp[i].find_net_sal();
        for(i=0;i<num;i++)
                emp[i].show_emp_details();
        return 0;
}
```

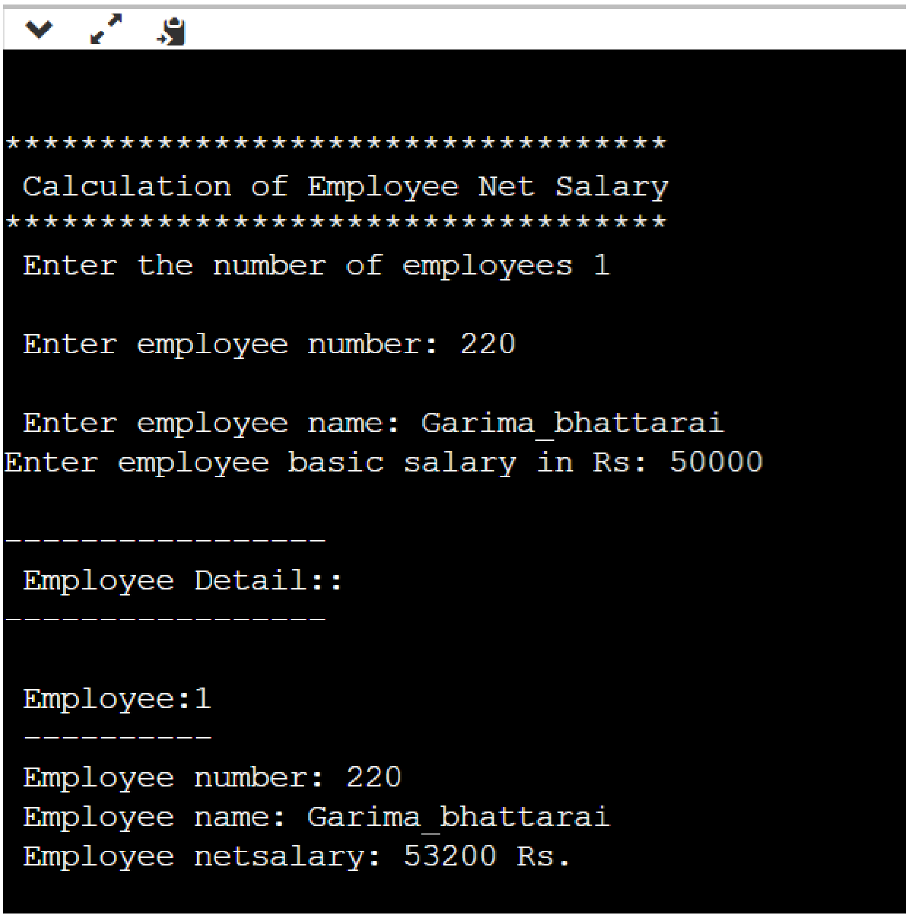


```
 Enter number of employee details
1

 Enter employee number:
2231

 Enter employee name:
uma

 Enter employee basic:
30000



 Details of     : uma

 Employee number:       2231
 Basic salary   : 30000
 Employee DA    : 15600
 Income Tax     : 13680
 Net Salary     : 31920
```

5. Write a C++ program to read the data of N employee and compute Net salary of each employee (DA=52% of Basic and Income Tax (IT) =30% of the gross salary).

```
#include<iostream>
using namespace std;
#define SIZE 5

class emp {
        float basic,da,it,netsal;
        char name[20],num[10];
    public:
        void getdata();
        void net_sal();
        void dispdata();
};
void emp::getdata() {
        cout<<"\n Enter employee number: " ;
        cin>>name;
        cout<<"\n Enter employee name: " ;
        cin>>num;
        cout<<"Enter employee basic salary in Rs: " ;
        cin>>basic;
}
void emp::net_sal( ){
        da=((0.52)*basic );
        float gsal=da+basic;
        it=((0.3)*gsal);
        netsal=gsal-it;
}
void emp::dispdata() {
        cout <<"\n Employee number: "<<name;
        cout <<"\n Employee name: "<<num;
        cout <<"\n Employee netsalary: "<<netsal<<" Rs.";
}
int main() {
        int i;
        emp ob[SIZE];
        int n;
        cout<<"\n\n***********************************"
        <<"\n Calculation of Employee Net Salary"
        <<"\n***********************************"
        <<"\n Enter the number of employees";
        cin>>n;
        for(int i=0;i<n;i++) {
                ob[i].getdata();
                ob[i].net_sal();
        }
        cout<<"\n-----------------"<<"\n Employee Detail::"<<"\n-----------------";
```

```
	for( i=0;i<n;i++) {
		cout<<"\n\n Employee:"<<i+1<<"\n ----------";
		ob[i].dispdata();
	}
}
```

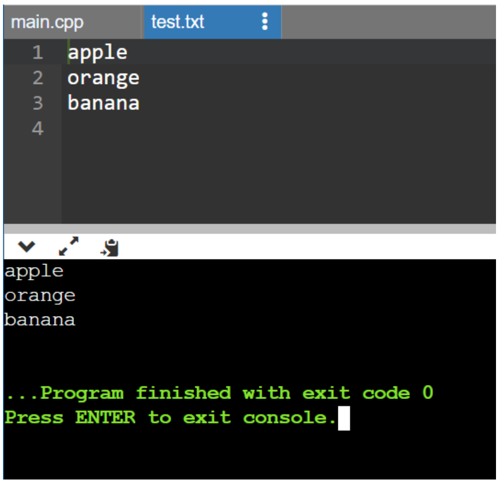


```


***********************************
 Calculation of Employee Net Salary
***********************************
 Enter the number of employees 1

 Enter employee number: 220

 Enter employee name: Garima_bhattarai
Enter employee basic salary in Rs: 50000

-----------------
 Employee Detail::
-----------------


 Employee:1
 ----------
 Employee number: 220
 Employee name: Garima_bhattarai
 Employee netsalary: 53200 Rs.
```

}

6. Write a C++ to illustrate the concepts of console I/O operations.

```
#include <iostream>
#include <fstream>
#include <cstdlib>
#include <string>
 using namespace std;
     int main()
{
    string filename = "test.txt";
     ofstream fout(filename.c_str()); // default mode is ios::out | ios::trunc
     if (!fout)
{
    cerr << "error: open file for output failed!" << endl; abort(); // in <cstdlib> header
}
    fout << "apple" << endl;
    fout << "orange" << endl;
    fout << "banana" << endl;
    fout.close();
    ifstream fin(filename.c_str()); // default mode ios::in
    If (!fin)
{
    cerr << "error: open file for input failed!" << endl;
    abort();
}
 char ch;
  while (fin.get(ch))
{
     // till end-of-file
  cout << ch;
}
    fin.close();
    return 0;
}
```

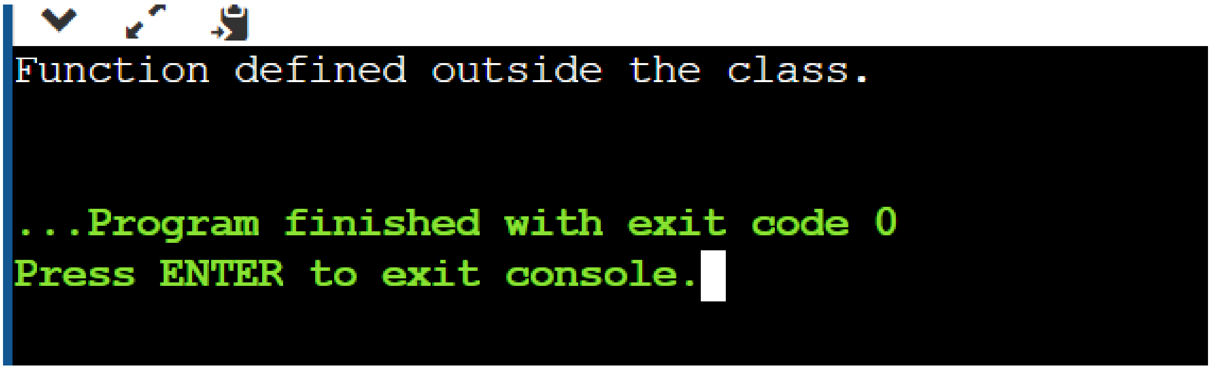


```
apple
orange
banana
```

```
apple
orange
banana


...Program finished with exit code 0
Press ENTER to exit console.
```

7. Write a C++ program to use scope resolution operator. Display the various values of the same variables declared at different scope levels.

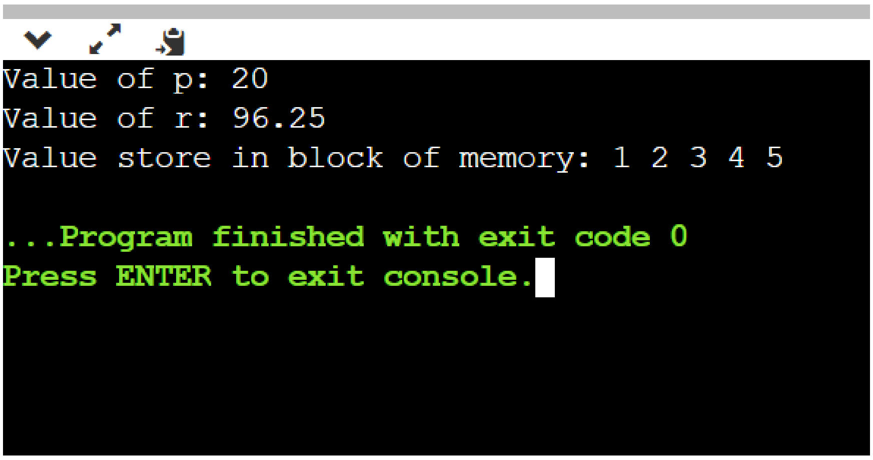
```
#include <iostream>
using namespace std;
class programming
{
  public: void output();  //function declaration
};

void programming::output()
{
   cout << "Function defined outside the class.\n";
}

 int main()
{
   programming x;
    x.output(); return 0;
}
```



```
Function defined outside the class.


...Program finished with exit code 0
Press ENTER to exit console.
```

8. Write a C++ program to allocate memory using new operator.

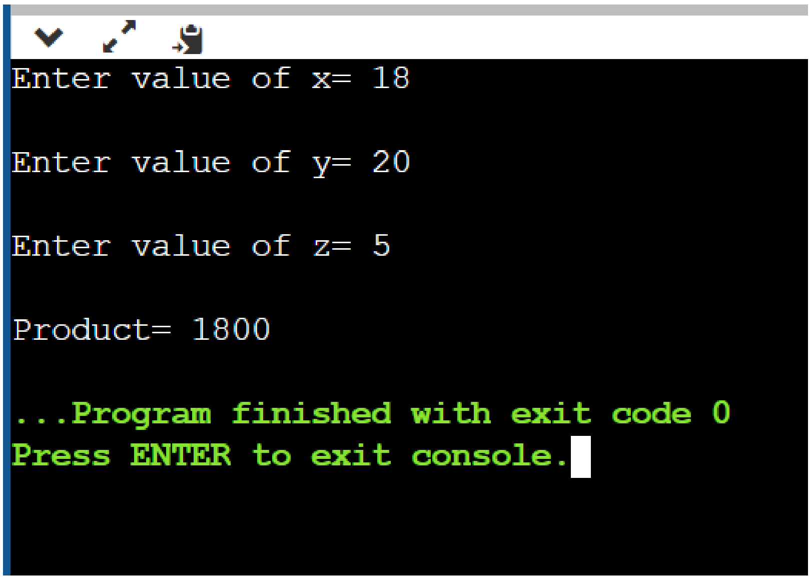
```
#include <iostream>
using namespace std;

int main ()
{
        int* p = NULL;
        p = new(nothrow) int;
        if (!p)
        cout << "allocation of memory failed\n";
        else
        {
                *p = 20;
                cout << "Value of p: " << *p << endl;
        }
        float *r = new float(96.25);
        cout << "Value of r: " << *r << endl;
        int n = 5;
        int *q = new(nothrow) int[n];
        if (!q)
                cout << "allocation of memory failed\n";
        else
        {
        for (int i = 0; i < n; i++)
        q[i] = i+1;
        cout << "Value store in block of memory: ";
        for (int i = 0; i < n; i++)
        cout << q[i] << " ";
        }
        delete p;
        delete r;
        delete[] q;
        return 0;
}
```



```
Value of p: 20
Value of r: 96.25
Value store in block of memory: 1 2 3 4 5

...Program finished with exit code 0
Press ENTER to exit console.
```

9.	Write a C++ program to create multilevel inheritance. (Hint: Classes A1, A2, A3)

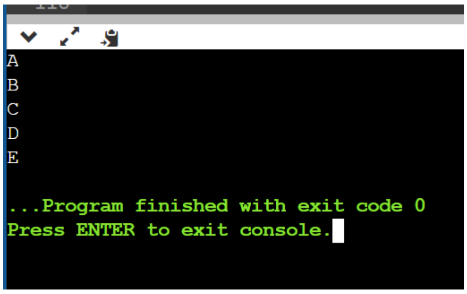
```
#include <iostream>
using namespace std;

class base   {
        public: int x;
        void getdata() {
                cout << "Enter value of x= "; cin >> x;
        } };
class derive1 : public base {
        public: int y;
        void readdata() {
                cout << "\nEnter value of y= "; cin >> y;
} };
class derive2 : public derive1 {
        private: int z;
        public:
                void indata() {
                        cout << "\nEnter value of z= "; cin >> z;
                        }
                void product() {
                        cout << "\nProduct= " << x * y * z;
                        }
        };
int main() {
        derive2 a;
        a.getdata();
        a.readdata();
        a.indata();
        a.product();
        return 0;
}
```



```
Enter value of x= 18

Enter value of y= 20

Enter value of z= 5

Product= 1800

...Program finished with exit code 0
Press ENTER to exit console.
```

10. Write a C++ program to create an array of pointers. Invoke functions using array objects.

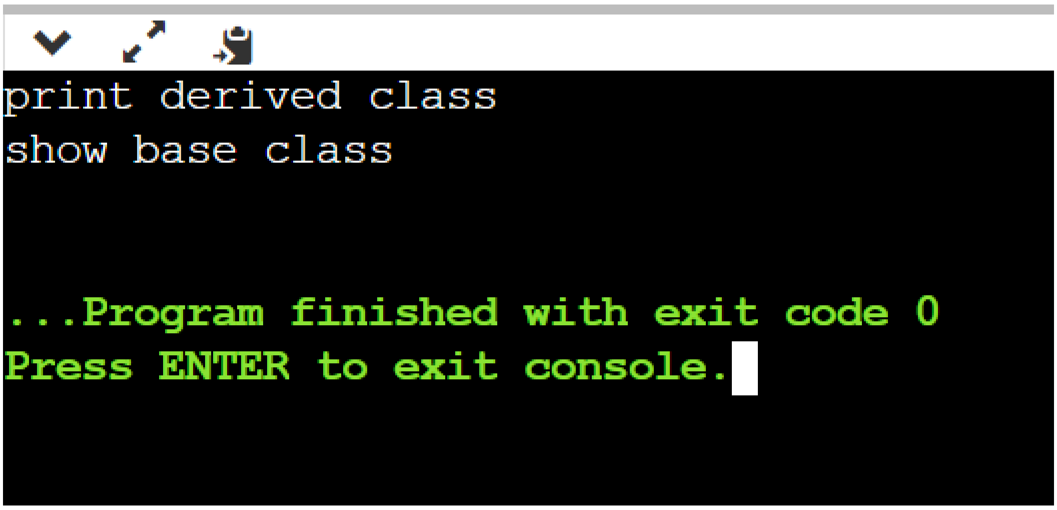
```
#include<iostream>
using namespace std;
class A {
        public: virtual void show() {
                cout<<"A\n";
        }
};
class B : public A {
        public:
                void show() {
                        cout<<"B\n";
        }

};
class C : public A {
        Public:
                void show() {
                        cout<<"C\n";
} };

class D : public A {
        public:
                void show() {
                        cout<<"D\n";
} };
class E : public A {
        Public:
                void show() {
                        cout<<"E";
                }
};
int main() {
        A a; B b; C c; D d; E e;
        A *pa[]={&a,&b,&c,&d,&e};
        for ( int j=0;j<5;j++) pa[j]->show();
        return 0;
}
```



```
A
B
C
D
E

...Program finished with exit code 0
Press ENTER to exit console.
```

11. Write a C++ program to use pointer for both base and derived classes and call the member.

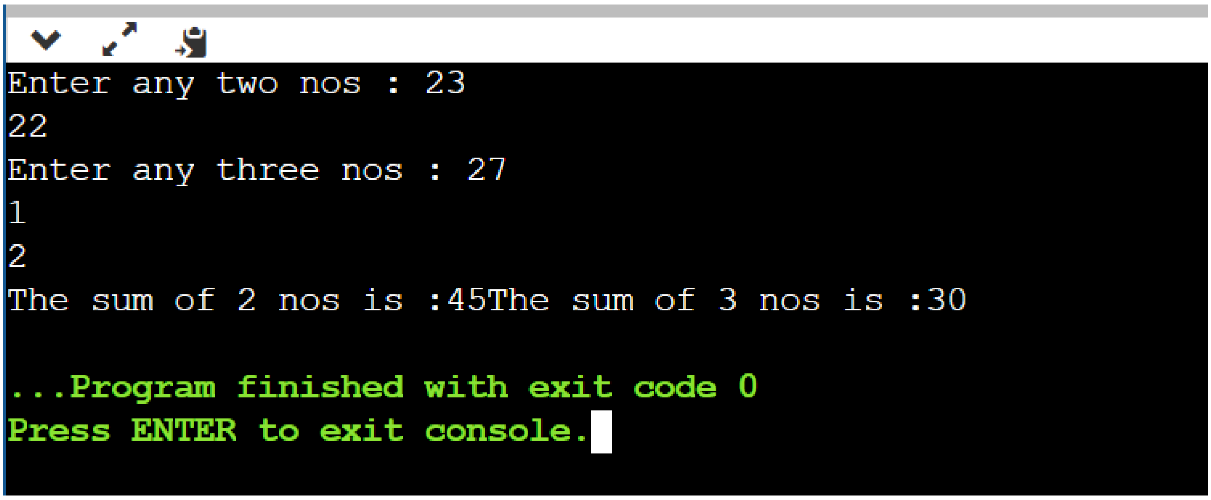
```
#include<iostream>
using namespace std;
class base {
    public:
        virtual void print () {
            cout<< "print base class" <<endl;
        }
    void show () {
        cout<< "show base class" <<endl;
        }
    };
class derived: public base {
    public:
        void print () {
            cout<< "print derived class" <<endl;
        }
        void show () {
            cout<< "show derived class" <<endl;
        }
};
int main() {
    base *bptr; derived d;
    bptr = &d;
    bptr->print();
    bptr->show();
}
```



```
print derived class
show base class


...Program finished with exit code 0
Press ENTER to exit console.
```

12. Write a C++ program to use function overloading.

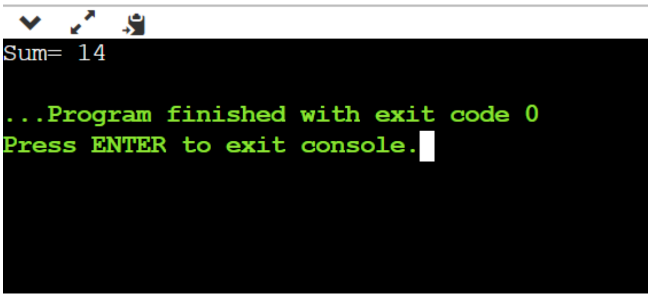
```
#include<iostream>
using namespace std;
int sum (int a, int b){
    return a+b;
 }
 int sum (int a, int b, int c){
    return a+b+c;
 }

 int main(){
   int a,b,c,d,e;
   cout<<"Enter any two nos : ";
   cin>>a>>b;
   cout<<"Enter any three nos : ";
   cin>>c>>d>>e;
   int s1= sum(a,b);
   int s2= sum(c,d,e);
   cout<<"The sum of 2 nos is :"<<s1;
    cout<<"The sum of 3 nos is :"<<s2;
    return 0;
 }
```



```
Enter any two nos : 23
22
Enter any three nos : 27
1
2
The sum of 2 nos is :45The sum of 3 nos is :30

...Program finished with exit code 0
Press ENTER to exit console.
```

13. Wap to show Hybrid inheritance.

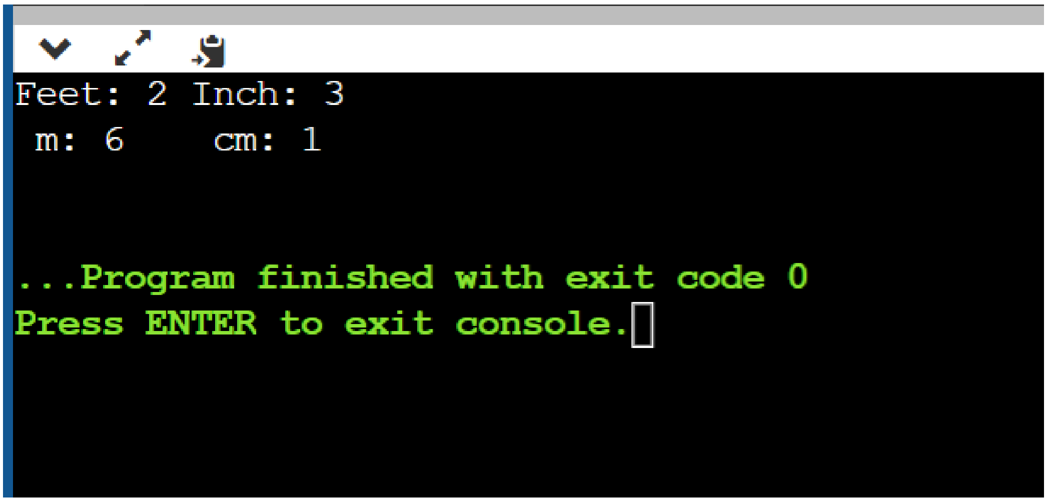
```
#include <iostream>
using namespace std;
class A {
    public:  int x;
};
class B : public A {
    public:
        B() {
            x = 10;
        }
};
class C {
    public:
        int y;
        C() {
            y = 4;
        }
    };
class D : public B, public C {
    public:
        void sum() {
            cout << "Sum= " << x + y;
        }
};

int main()
{
   D obj1;       //object of derived class D
    obj1.sum();
    return 0;
}
```



```
Sum= 14

...Program finished with exit code 0
Press ENTER to exit console.
```

14. Write a C++ program to demonstrate operator overloading.

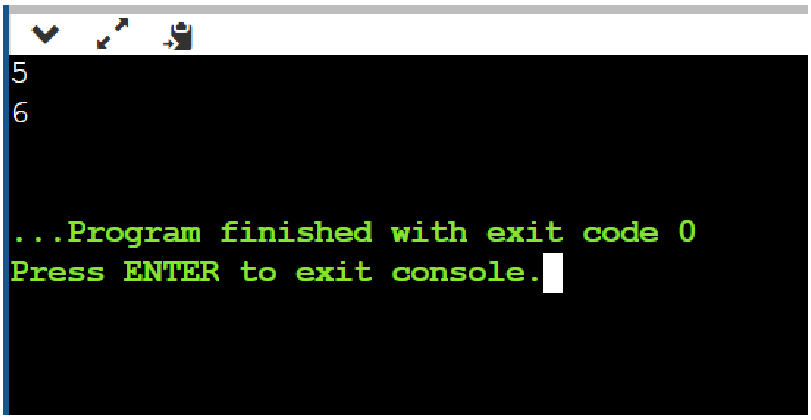
```
#include<iostream>
using namespace std;
class Distance {
   public:
            int feet, inch;
            Distance(int feet, int inch) {
                  this->feet = feet;
                  this->inch = inch;
           }
          void display() {
                  cout<<"Feet: "<<feet<<"\tInch: "<<inch<<endl;
         }
};
class SecDistance {
   public:
            int m,cm;
            void operator = (Distance a) {
                  m = a.feet * 3.3;
                  cm = a.inch * 0.4;
            }
           void display() {
                  cout<<" m: "<<m<<"\t cm: "<<cm<<endl;
                  }
 };
int main() {
   Distance distance = Distance(2,3);
   distance.display();
   SecDistance secDistance;
   secDistance = distance;
   secDistance.display();
   return 0;
}
```



```
Feet: 2 Inch: 3
 m: 6     cm: 1


...Program finished with exit code 0
Press ENTER to exit console.
```

15. Write a C++ program to use class template.

```
#include<iostream>
using namespace std;

template <class T> class DateTime {
        private:
                T time;
        public:
                DateTime(T time) {
                        this->time = time;
                }
                void tick() {
                        time++;
                }
                T getTime(){
                        return time;
                }
};

int main() {
   DateTime<int> *dt = new DateTime<int>(5);
   cout<<dt->getTime()<<endl;
   dt->tick();
   cout<<dt->getTime()<<endl;
   return 0;
}
```



```
5
6


...Program finished with exit code 0
Press ENTER to exit console.
```