

## INDEX

Name : Jayraj Pawar

Std. : \_\_\_\_\_ Div. : \_\_\_\_\_ Roll No. : \_\_\_\_\_

Sub.: 5400P

School / College : SYAIRADS



Books for Success

## Experiment - 1

- a) Way to declare a class student having data members name, rollno, Accept and display for one student

```
#include <iostream>
#include <string>
using namespace std;
class student
{
    int r;
    string n;
public,
    void accept()
    {
        cout << "Enter name and roll number:" ;
        cin >> n >> r;
    }
    void display()
    {
        cout << "Name = " << n << endl;
        cout << "Roll no = " << r << endl;
    }
};

int main()
{
    student s1;
    s1.accept();
    s1.display();
    return 0;
}
```

Output

Enter name and roll no : Jayraj Pawar  
38

Name = Jayraj Pawar

Roll number = 38

b) WAP to declare a class book having data members as id, name, price, etc. Accept data for 2 books and display data of book having greater price.

→

```
#include <iostream>
#include <string>
using namespace std
class Book;
public:
    int b1, b2;
    string n1, n2;
    void accept();
    void display();
};

void Book::accept()
{
    cout << "Enter name and price:" >> n1 >> b1;
    cout << "Enter pages:" >> n2 >> b2;
}

void Book::display()
{
    cout << "Book name = " << n1 << endl;
    cout << "Book price = " << b1 << endl;
    cout << "Book pages = " << b2 << endl;
}

int main()
{
    Book b1, b2;
    b1.accept();
}
```

```
b2.accept();
if (b1.bprice > b2.bprice)
    cout << "Book with higher price";
b1.display();
}
else {
    cout << "Book with higher price";
    b1.display();
}
else {
    cout << "Book with higher price";
    b2.display();
}
return 0;
}
```

### Output

Enter name and price : ABC 50

Enter pages : 200

Enter name and price : XYZ 100

Enter pages 300

Book with higher price

BOOK name = XYZ, BOOK price = 100

BOOK price = 100

BOOK pages = 300

c) WAP to declare a class Time having data members as H, M and S. Accept data for one object and display total time in sec

→

```
#include <iostream>
#include <string>
using namespace std;
class Time
{
    int h,m,s,t;
public:
    void accept()
    {
        cout << "Enter hours:-";
        cin >> h;
        cout << "Enter minutes:-";
        cin >> m;
        cout << "Enter seconds:-";
        cin >> s;
    }
    void display()
    {
        t = (h * 3600) + (m * 60) + s;
        cout << "Total time = " << t;
    }
}
```

3

```
int main()
```

{

Time t1;

t1.accept();

t2.display();

return 0;

}

Output :-

Enter hours: 4

Enter minutes: 6

Enter seconds: 89

total time: 14899

~~Ques  
12/11~~

## Experiment - 2

- 1) Write a C++ program to demonstrate the use of class and object

MAP to declare a class 'city' having data members as name and population Accept this data for 25 cities and display name of city having highest population

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

```
class city
```

```
{
```

public:

```
string name;
int population
```

```
void accept()
```

```
{
```

```
<< "city name" << name << std::cin >> name>> population
```

```
}
```

~~void display()~~

```
{
```

```
cout << "cityname : " << name << endl;
cout << "Population : " << population << endl;
```

```
}
```

```
};
```

```
int main()
```

{

```
vector<city> cities(5);
```

```
for (int i = 0; i < 5; i++)
```

{

```
cout << "Enter data for city " << i + 1 << endl;
```

```
cities[i].accept();
```

}

```
city highestPopularityCity = cities[0];
```

```
for (int i = 1; i < 5; i++)
```

{

```
if (cities[i].population > highestPopulation  
    city.population)
```

{

```
highestPopulationCity = cities[i];
```

}

}

~~cout << "City with highest population " << endl;~~~~cout << "Name: " << highestPopulationCity.name << endl;~~~~cout << "Population: " << highestPopulationCity.population << endl;~~~~<< endl;~~~~return 0;~~

}

~~Output~~

```
Enter data for city 1:
```

```
Enter name and population of city : Delhi 2000
```

```
Enter data for city 2:
```

```
Enter name and population of city : Mumbai 2000
```

Enter data for city 3

Enter name and population of the city: Kolkato 900

Enter data for city 4

Enter name and population of city: chennai 1000

Enter data for city 5

Enter name and population of city Bangalore 1009

City with highest population

Name: chennai

Population: 10000

- b) Write to declare a class Account having data members as Account no. and balance. Accept this data for 10 accounts and give interest 10% where balance is equal or greater than 5000 - and display them

→

```
#include <iostream>
```

```
using namespace std;
```

```
class Account {
```

```
int acc_no;
```

```
float balance;
```

```
public:
```

```
void accept();
```

```
cout << "Enter account number:";
```

```
cin >> acc_no;
```

```
cout << "Enter balance:";
```

```
cin >> balance;
```

```

void addinterest () {
    if (balance >= 500) {
        balance += balance * 0.10;
    }
}

void display () {
    if (balance >= 5000) {
        cout << "account no: " << accno;
        cout << "balance with interest: " << balance;
    }
}

int main () {
    account acc[10];
    for (int j = 0; j < 10; j++) {
        cout << "Enter details for account " << (j + 1) << "\n";
        acc[j].accept();
        acc[j].addinterest();
    }

    cout << "In-accounts balance = $5000 after 3 yrs";
    cout << "\n";
    for (int i = 0; i < 10; i++) {
        acc[i].display();
    }
    return 0;
}

```

Output

Enter details for account 1

Enter account number: 17

Enter balance: 845

enter details for account 2:

enter account number: 56

enter balance: 876

enter details for account 3:

enter account number 45

enter balance: 897

enter details for account 4:

enter account number: 55

enter balance: 678

enter details for account 5:

enter account number: 669

enter balance: 7788

enter details for account 2:

enter account number 28

enter balance: 8888

enter details for account 8

enter account number 45

enter balance 7890

enter details for account 9

enter account number: 33

enter balance: 5644

Enter details for account '0  
 Enter account number: 33  
 Enter balance: 67865

Account with balance = 5000 adding  
 interest

account no: 666 balance with integer: 8552  
 account no: 78 balance with integer: 92776  
 account no: 45 balance with integer: 8679  
 account no: 23 balance with integer: 62086  
 account no: 33 balance with integer: 74851

- 3) Write a program to declare a class STAFF  
 having data members as name & post  
 Accept this data for 5 staff & display  
 names of staff who are 'HOD'

→

```
#include <iostream>
#include <string>
using namespace std
class staff {
    string name
    string post
public
void accept() {
    cout << "Enter name: ";
    cin >> name;
    cout << "Enter Post: ";
    cin >> post;
}
```

```

void display() {
    if (post == "Head") {
        cout << "Name" << name << endl;
    }
    int matches = 0;
    Staff S[5];
    for (int i = 0; i < 5; i++) {
        cout << "Enter staff with post Head : ";
        for (int j = 0; j < 5; j++) {
            S[j].display();
        }
    }
    return 0;
}

```

### Output:-

Enter the details of staff 1

Enter staff Name: Swapna

Enter staff Post: Head

Enter details for staff 2

Enter staff name: Maria

Enter staff Post: Head

Enter details for staff 3

Enter staff name : Jane

Enter staff Post : Head

Enter details for Staff #3

Enter staff name : Mark

Enter staff post : HOD

Enter details for staff #4

Enter staff name : Mark

Enter staff Post : HR

Enter details for staff 5

Enter staff name : ~~Jacan~~ JVB

Name of staff

Enter staff post : Head  
staff who are HOD:

Name of staff : Jacan

Name of : Jane

On  
12/11

PAGE NO.	
DATE	/ /

### Experiment - 3

(Q1)

```

→ #include <iostream>
using namespace std;
class book
{
public:
    string book_title
    int Price
    string author_name;
void accept()
{
    cout << " enter book name: ";
    cin >> book_title;
    cout << " enter the name of author: ";
    cin >> author_name;
    cout << " Price of the book: ";
    cin >> Price;
}
void display()
{
    cout << " bookname is: " << book_title;
    cout << " In author name: " << author_name;
    cout << " In Price is: " << Price;
}
}

```

```
int main()
```

```
{
```

```
book bl;
```

```
book * p;
```

```
bl . accept();
```

```
bl . display();
```

```
return 0;
```

```
}
```

```
{
```

Output

```
Enter book name : Python
```

```
Enter name of author : Ruse
```

```
rice of the book : 999
```

```
BOOK name is : Python
```

```
Author name is : Ruse
```

```
Rice is : 999
```

2) Write a program to declare a class 'student' having data members rollno & %. Using 'this' pointer invoke member functions to accept & display this data for one object of the class

```
#include<iostream>
```

```
using namespace std;
```

```
class student
```

```
{
```

```
int rollno;
```

```
float percentage;
```

```
public
```

void accept()

{

cout << "Enter student roll-no.:";

cin >> this -> roll\_no

cout << "Enter student Percentage:";

cin >> this -> Percentage;

}

void display()

{

cout << "Roll number of student is:" << roll\_no;

cout << "In percentage is" << Percentage;

}

}

int main()

{

Student s1;

Student \* p;

p = &s1;

s1.accept();

s1.display();

return 0;

}

Output

Enter the student roll-number 39

Enter the student percentage = 39

Percentage = %.93

3) WAP to demonstrate use of nested class

→ ~~include <iostream>~~

Class namespace std;

Class students

{

Public:

int num;

Class operations

{

Public

int space(int n)

{

return n \* n;

}

}

Public:

Void accept()

{

cout << "Enter a number";

cin >> num

{

Void display()

{

Operations op:

cout << "square of: " << num;

cout << "cube of: " << num;

3  
8;  
int main()  
{

    number n;  
    n = accept();  
    n = display();  
    return 0;

8

### Output

Enter number : 2

Square of 2 : 4

~~Cube of 2 : 8~~

Ques  
(2/11)

## Experiment - 4

- i) Write to swap 2 numbers from same class object & as function argument write swap function as member function.

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

```
class number {
    int value;
```

```
public
```

```
// Function to display the value
```

```
void display () {
```

```
cout << "Value:" << value << endl;
```

```
void swap (Number & obj) {
```

```
int temp = value;
```

```
value = temp;
```

```
}
```

```
}}
```

```
int main () {
```

```
Number num1, num2;
```

```
// Set initial values
```

```
num1.set_value (10);
```

```
num2.set_value (20);
```

```
cout << "Before swap: " << endl;
```

```
cout << "Num1: " << num1.display ();
```

```
cout << "Num2: " << num2.display ();
```

`num1 · Swap (num2);`

`cout << "In · After Swap;" << endl;`

`cout << "num1:" num1 · display();`

`cout << "num2:" num2 · display();`

`return 0;`

`}`

`3`

Output:

Before swap:

`num1 · value: 10`

`num2 · value: 20`

After swap

`Num.1 · value: 20`

`Num.2 · value: 10`

2) WAP to swap two numbers from same class using the concept of a friend function.

~~#include <iostream>~~

~~using namespace std;~~

~~class number {~~

~~int value;~~

Public:

~~// function to get value~~

~~void set value (int v) {~~

~~value = v;~~

9

```
// function to display value
```

```
Void display() {
```

```
cout << "Value" << Value << endl;
```

9

```
// Declare friend function
```

```
friend void swap( Number&a, Number&b );
```

9:

```
// Friend function definition
```

```
void swap( Number &a, Number &b )
```

```
int temp = a.value,
```

```
a.value = b.value,
```

```
b.value = temp;
```

9

```
int main () {
```

```
Number num1, num2;
```

```
// set initial values
```

```
num1.setValues( 25 );
```

```
num2.setValues( 15 );
```

~~cout << "Before Swap:" << endl;~~~~cout << "num1, num1.display()";~~~~cout << "num2, num2.display()";~~

### Heat

```
// call the friend swap function
```

```
swap( num1, num2 );
```

`cout << "In After swap;" << endl;`  
`cout << "num1:" num1 displays`

? `return 0;`

Output

Before swap

`num1::value: 5`

`num2::value: 15`

After swap

`num1::value: 15`

`num2::value: 5`

Q WAP to swap two numbers from different class using friend function.

→ #include <iostream>

using namespace std;

class num1;

class num2 {

~~private~~

int a;

public:

void accept() {

`cout << "Enter the First number";`

~~for~~ (in ~~out~~ a)

}

friend void swap (num1 &, num2 &);

} ;

class num2 {

private:

int b;

public

void accept();

cout << "Enter the second number:";

(cin >> b);

}

friend void swap( int num1 & int num2 );

void swap( int &num1, int &num2 ) {

int temp;

temp = num1;

num1 = num2;

num2 = temp;

cout << "After Swapping:" << endl;

cout << "First number = " << num1 << endl;

cout << "Second number = " << num2 << endl;

}

int main()

{ int n1;

int n2;

n1.accept();

n2.accept();

swap(n1, n2);

return 0;

}

Output:-

Enter the First number = 5

Enter the second number = 6

After Swapping:

First Number = 6

Ques  
12(1)

## Experiment - 5

5b)  #include <iostream>  
using namespace std;

```
class student {
    string n;
    float p;
public:
    student(string a, float b) {
        n = a;
        p = b;
    }
}
```

```
void show() {
    cout << n << " " << p;
}
```

}

```
int main() {
    string n;
    float p;
    cin >> n >> p;
    student s(n, p);
    s.show();
}
```

~~return 0;~~

5a) #include <iostream>  
using namespace std;

class sum {

int n, s;

public

sum(int x) {

n = x;

s = 0;

for (int i = 1, i <= n; i++)

s += i;

cout << s;

}

};

int main() {

int n;

cin >> n;

sum a(n);

}

return 0;

};

5.) #include <iostream>  
using namespace std;

```
class college{
    int roll;
    string name, course;
public:
    college (int r, string n, string c = "Computer Engineering") {
        roll = r;
        name = n;
        course = c;
    }
}
```

```
void show() {
    cout << roll << " " << name << ""
    << course << endl;
}
```

3:

```
int main() {
college a(1,"A"), b(2,"B","IT");
a.show();
b.show();
}
```

5d) #include <iostream>  
using namespace std;

```
class Demo {  
    int x;  
public:  
    Demo() {  
        x = 0;  
        cout << x << endl;  
    }  
    Demo(int a) {  
        x = a;  
        cout << x << endl;  
    }  
};
```

```
int main() {  
    Demo a, b(5), c(b);
```

```
    return 0  
}
```

Ques  
(21)

## Experiment 6

A) #include <iostream>

using namespace std;

class A {

public:

void showA() { cout << "A"; }

}

class B : public A {

public:

void showB() { cout << "B"; }

}

class C : public B {

public

void showC() { cout << "C"; }

}

int main() {

C c;

c.showA();

c.showB();

c.showC();

2

return 0;

}

(6b) #include <iostream>  
using namespace std;

class A {

public:

void showA () { cout << "A"; }

}

class B {

public:

void showB () { cout << "B"; }

}

class C : public A, public B {

public:

void showC () { cout << "C"; }

}

int main ()

{

C c;

c.showA();

c.showB();

c.showC();

return 0;

6) #include <iostream>  
using namespace std;

class A {

public:

void showA () { cout << "A" ; }

class B : public A {

public:

void showB () { cout << "B" ; }

class C : public A {

public:

void showC () { cout << "C" ; }

}

int main () {

B b;

C c;

b.showA ();

c.showA ();

return 0;

}

6d) #include <iostream>  
using namespace std;

class A {

public:

void showA() { cout << "A"; }

}

class B : virtual public A {

public:

void showB() { cout << "B"; }

}

class C : virtual public A {

public:

void showC() { cout << "C"; }

}

class D : public B, public C {

public:

void showD() { cout << "D"; }

}

int main() {

D d;

d.showA();

d.showB();

d.showC();

d.showD();

}

return ej  
3

①  
read

## Experiment - 7

7a) `#include <iostream>`  
using namespace std;

`void area (int l, int b) {`  
`cout << l * b;`  
}

`void area (int s) {`  
`cout << s * s;`  
}

`int main () {`  
`int l, b;`  
`cin >> l >> b;`  
`area (l, b);`  
}

`return 0;`  
}

Q6) #include <iostream>  
using namespace std;

class sum {

public:

float add (float a, float b, float c, float d, float e);  
return a+b+c+d+e;

}

:

int add (int a1, int a2, int a3, int a4, int a5,  
int a6, int a7, int a8, int a9, int a10)

return a1+a2+a3+a4+a5+a6+a7+a8+a9+a10;

;

};

int main () {

sum s;

~~COW LC~~ "sum of 5 floats : " << s.add(1.1f, 2.2f,

3.3f, 4.4f, 5.5f) << endl;

~~COW LC~~ "sum of 10 integers : " << s.add (1, 2, 3, 4, 5,

6, 7, 8, 9, 10) << endl;

return 0;

;

Q) #include <iostream>  
using namespace std;

class Number {  
 int n;

public  
 number (int a) {  
 n = a;  
 }

void display () {  
 cout << "value" << n << endl;  
}

void operator++ () {  
 n++;  
}

int main () {  
 number n(5);  
 n.display();  
 ++n;  
 n.display();  
 n++;  
 n.display();  
 return 0;  
}

Qd) #include <iostream>  
using namespace std

```
class Number {
    int n;
public:
    Number(int a) {
        n = a;
    }
}
```

```
void display() {
    cout << value << endl;
}
```

```
void operator -( ) {
    n = -n;
}
```

```
int main() {
    Number n(9);
    n.display();
    -n;
    n.display();
    return 0;
}
```

Qd  
11

# Experiment 8

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

class concat  
string s;

public:

concat(string a = "") {s = a; }  
concat operator + (concat b) {  
 return concat(s + b.s); }

};

void show() {cout << s; }

};

int main()

concat a("xyz"), b("pqr"), c;

c = a + b

c.show();

return 0;

}

8b

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

```
class Login {
protected:
    string name, pwd;
public:
    virtual void accept() {
        cin >> name >> pwd;
    }
};
```

```
Class EmailLogin : public login
```

```
public
    void accept() {
        cin >> name >> pwd;
    }
};
```

```
int main() {
    EmailLogin e;
    MemberLogin m;
    e.accept();
    m.accept();
    return 0;
}
```

Q  
12/11

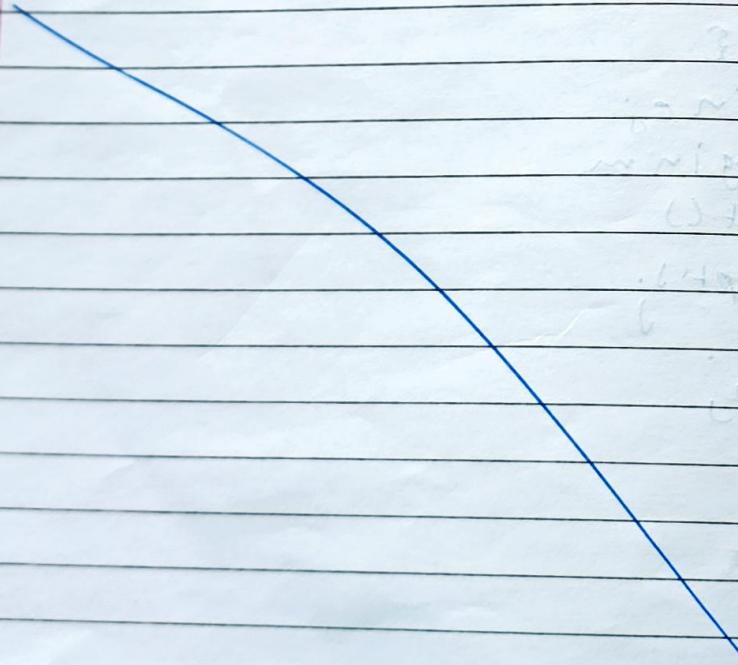
## Experiment - 7

- a) ~~#include <iostream>~~  
~~#include <fstream>~~  
using namespace std

```
int main () {  
    istream p1 (first.txt);  
    ofstream p2 (second.txt);  
    char ch;  
    while (p1.get(ch))  
        p2.put(ch);  
    p2.putcn
```

Return obj:

}



Qb) ~~#include <iostream>~~  
~~#include <fstream>~~  
 using namespace std;

int main() {

ifstream p1(first.txt)  
 ofstream p2(second.txt  
 char ch  
 while (p1.get(ch))  
 p2.put(ch);

}

return 0;

}

Q) ~~#include <iostream>~~  
~~#include <fstream>~~  
~~#include <string>~~  
 using namespace std;

int main()

fstream('file.txt')

string w

int count = 0

while (ws > w)

count++

cout << count

3

return 0;

Open  
File

## Experiment 10

Q1

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

```
template <class T>
T myfunction(T a[], int n) {
    T sum = 0;
    for (int i = 0; i < n; i++) {
        sum = sum + a[i];
    }
    return sum;
}
```

3

```
int main () {
```

```
    int a1[3] = {2, 0, 3};
```

```
    float a2[3] = {1.2, 1.3, 2.4};
```

```
    double a3[3] = {10.5, 20.5, 30.5}
```

```
    cout << "sum of int array is : " << myfunction(a1, 3) << endl;
```

```
    cout << "sum of float array is : " << myfunction(a2, 3) << endl;
```

```
    cout << "sum of double array : " << myfunction(a3, 3) << endl;
```

```
    return 0;
}
```

Q2 #include <iostream>  
using namespace std;

```
template <class T>
T square(T num) {
    return num * num;
}
```

templated

```
String square (string str) {
    return str + str;
}
```

```
int main() {
    int i = 5;
    String s = "Hello";
```

cout << "square of integer" << i << "=" << square(i)

<< endl;

cout << "square of string" << s << "!" <<

~~Square(s)~~ << endl;

return 0;

}

(c) include <iostream>  
using namespace std

template <class T>

class calculator

T add(T x, T y)

public;

calculator (T x, T y) {

a = x;

b = y;

}

T add() { return a + b; }

T sub() { return a - b; }

T mul() { return a \* b; }

T div() {

if (b != 0)

return a / b;

else {

cout = "Error! Division by zero!" << endl;

return 0;

}

int main()

calculator<int> c(10, 5);

cout << Addition = << c.add();

cout << Division = c.div();

return 0;

3

d) #include <iostream>  
using namespace std;

```
template (class T)
class stack {
    T s[10];
    int top;
public:
    stack() { top = -1; }
    void push (T x) {
        s[++top] = x;
    }
    void pop () { if (top >= 0) top--; }
    void display () { for (int i = 0; i < top; i++)
        cout << s[i]; }
}
```

```
int main () {
    stack <int> s
    s.push(1);
    s.push(2);
    s.push(3);
    s.pop();
    s.display()
}
return 0;
```

Qn  
T2/11

## Experiment II

11(a) ~~#include <iostream>~~  
~~#include <vector>~~  
using namespace std;

template <class T>

class member {

vector<T> v;

public:

void createvector (int n) {

T val;

for (int i = 0; i < n; i++) {

cin >> val;

v.push\_back (val);

}

void modifyvalue (int index) {

if (index == 0 || index > v.size())

v[0] = modify;

void multiply\_byscalar (T scalar) {

for (int i = 0; i < v.size(); i++)

v[i] \*= scalar;

}

void display () {

cout << "(";

for (int i = 0; i < v.size(); i++) {

cout << v[i];

if (i == v.size() - 1)

cout << ",";

}

cout << ")" << endl;

{}

```
int main() {  
    myvector<int> obj;  
    int n;  
    cin >> n;  
    obj.createvector(n);  
    obj.display();  
    obj.modifyValue(1, 50);  
    obj.display();  
    obj.multiplyByScalar(2);  
    obj.display();  
    return 0;  
}
```

Qn  
2/1

# Experiment 12

PAGE NO.  
DATE

12(a)

```
#include <iostream>
#ifndef include <stack>
using namespace std;
int main() {
    stack <int> s;
    s.push(10);
    s.push(20);
    s.push(30);
    cout << "Top: " s.top() << endl;
    s.pop();
    cout << "Top after pop: " << s.top();
    return 0;
}
```

12(b)

```
#include <iostream>
#include <queue>
using namespace std;
int main() {
    queue <int> q;
    q.push(10);
    q.push(20);
    q.push(30);
    cout << "front: " << q.front() << endl;
    q.pop();
    cout << "front after pop: " << q.front() << endl;
    return 0;
}
```

Qn  
Ans

```

11(b) #include <iostream>
#include <vector>
using namespace std
int main()
{
    vector<int> v = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
    cout << "Initial vector : " << endl;
    for (vector<int>::iterator it = v.begin(); it != v.end(); ++it)
        cout << *it << " ";
    cout << endl;
    cout << "multiply by 10" << endl;
    for (vector<int>::iterator it = v.begin(); it != v.end(); ++it)
        *it *= 10;
    cout << endl;
    cout << "New vector : " << endl;
    for (vector<int>::iterator it = v.begin(); it != v.end(); ++it)
        cout << *it << " ";
    cout << endl;
    cout << "return" << endl;
}

```