

## \*Experiment - 01\*

```
#include <iostream>
using namespace std;
class student
{
    int roll;
    string name;
public:
    void details()
}
cout << "Enter the name & roll number << endl;
cin <> name >> roll;
cout << "The name is << name;
cout << "The roll number is << roll;
}
int main()
{
    student s1;
    s1.details();
}
```

Output:

enter the name & roll number:

aarya

21

the student name is aarya

the roll number is 21.

2.

→ `#include <iostream>`

`using namespace std;`

`class times`

`{`

`Cout << "enter your times in hour, minutes & seconds";`

`Cin >> hour >>`

`int hour, min, sec;`

`public:`

`void details()`

`{`

`Cout << " enter your time in hour, minute & seconds";`

`Cin >> hour >> min >> sec;`

`}`

`Void Convert()`

`{`

`total += sec + (min * 60) + ((hour * 3600));`

`Cout << " your time in sec " << total;`

`}`

`};`

`int main()`

`{`

`times s1;`

`s1. details();`

`s1. Convert();`

`}`

Output : enter your time in hour, min & sec 3

46

11

your time in sec 13571

3]

→ #include < iostream >

using namespace std;

class book

{

int page, page2, price, price2; i;

string name, name2;

public:

void details()

{

Cout << "enter the name of 2 books, no of page & price"

Cout << "book 1";

Cin >> name >> page >> price

Cout << "book 2";

Cin >> name2 >> page2 >> price2;

{

void comp()

{

if (price > price2)

{

Cout << " " << name;

{

else

{

Cout << " " << name2;

{

{

int main()

{

Student s1;  
s1. details();

{ S1.comp(); }

Output;

enter the name of 2 books, no of pages & price

Book 1 name

123

220

Book 2 namee

125

250

namee.

Qn  
118



## \* Experiment - 02 \*

1.

→

```
#include <iostream>
```

```
using namespace std;
```

```
class city {
```

```
public:
```

```
string name[5];
```

```
int pop[5], max, i;
```

```
void accept ()
```

```
{
```

```
cout << "Enter the name of city & population:";
```

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

```
{ cin >> name[i] >> pop[i];
```

```
}
```

```
}
```

```
void display ()
```

```
{
```

~~max = pop[0];~~~~int k=0;~~~~for(i=0; i<5; i++)~~~~{~~~~if(pop[i]>max) {~~~~max = pop[i];~~~~k=i;~~~~{~~~~{~~

```
Cout << "City with greater population:" << name[k] << pop[k]
```

```
{
```

```
} c;
```

```
int main()
```

```
{  
    c.accept();  
    c.display();  
    return 0;  
}
```

## \*Experiment no:-2\*

2]

```
→ #include <iostream>
using namespace std;
class account
{
public:
    int acc_no;
    float bal;
    void accept ()
    {
        cout << "enter account number " << endl;
        cin >> acc_no;
        cout << "enter bal " << endl;
        cin >> bal;
    }
    void display ()
    {
        cout << "account number :" << acc_no << endl;
        cout << "account balance :" << bal << endl;
    }
};

int main ()
{
    account a[5];
    int i;
    for(i=0; i<5; i++)
    {
        cout << "enter details " << i+1 << endl;
        a[i].accept ();
    }
}
```

```
for (i=0; i<5; i++)  
if (a[i].bal > 5000)  
{
```

```
    a[i].bal = a[i].bal + (a[i].bal * 0.10);  
}
```

```
cout << "account with the updated balance:";
```

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

```
    a[i].display;  
}
```

#### \* Output :-

enter the balance 5000

enter the account details 1

enter the account number 7658

enter the bal : 4000

enter the account details 2

enter the account number 5876

enter the bal : 6000

enter the account details 3

enter the account number 8567

enter the bal : 4500

enter the account details 4

enter the account number : 5768

enter the bal : 7000

enter the account details 5

enter the account number : 8857

enter the bal : 8000

account with updated balance

account number 7658

- 11 -      bal : 4000

- 11 -      number : 5876

- 11 -      bal : 6000

- II - number : 8567
- II - balance : 4500
- II - number : 5168
- II - balance : 7700
- II - number : 8857
- II - balance : 8800

3] #include <iostream>

#include <string.h>

using namespace std;

{ class staff

public :

char name[50], post[50];

void accept()

{

cout << "Enter staff name & post" << endl;

cin >> name >> post;

}

void display()

{

cout << "member having post HOD is " << name << endl;

}

int main()

{

staff a[5];

int i;

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

{

a[i].accept;

}

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

{

if (strcmp(a[i].post, "HOD") == 0)

{ a[i].display();

}

}

}

Output:-

enter the staff name: jim  
enter the post: bod  
enter the staff name: lij  
enter the post: pod  
enter the staff name: freak  
enter the post: HOD  
enter the staff name: ERP  
enter the post: pp  
enter the staff name: rgg  
enter the post: cod  
member having post HOD is freak.

Q1  
118

### \*Experiment no: 3\*

```
[1] # include <iostream>
using namespace std;
class book
{ public:
    int price;
    string book_title;
    string author_name;
    void accept()
    {
        cout << " Enter title,author & price" << endl;
        cin >> book_title >> author_name >> price;
    }
    void display()
    {
        cout << "title is : " << book_title << endl;
        cout << "Author is : " << author_name << endl;
        cout << "Price is : " << price << endl;
    }
};

int main()
{
    book b1;
    book * p = &b1;
    p->accept();
    p->display();
    return 0;
}
```

enter

Output :-

Enter title, Author & price: The winner's Mindset Share  
649 ;

Book Title is: The winner's mindset

Author

Name is: Share Watron

Price is: 649.

2]

```
#include <iostream>
```

```
using namespace std;
```

```
class student
```

```
{ float perc; public
```

```
int roll;
```

```
string name;
```

~~float public~~ float perc;

```
void accept()
```

{

```
cout << "Enter roll no:" << endl;
```

```
cin >> this->roll;
```

```
cout << "Enter name:" << endl;
```

```
cin >> this->name;
```

```
cout << "Enter percentage:" << endl;
```

```
cin >> this->perc;
```

{

```
void display()
```

{

~~this → accept()~~~~cout << "roll no of student" << this->roll << endl;~~~~cout << "name of student" << this->name << endl;~~~~cout << "percentage of student" this->perc << endl;~~

{

```
int main()
```

```
{ Student s;
```

```
s.display();
```

{

3]

→ #include <iostream>  
using namespace std;

class student

{  
public:  
int roll;  
string name;  
public: void accept ()

cout << "enter roll no & name:" ;  
cin >> roll >> name;  
}

class marks

{ public:  
int CPP\_marks;  
int C\_marks;  
float percentage;  
int add;  
float d;  
void accept ()

cout << "enter C & C++ marks : " ;  
cin >> C\_marks >> CPP\_marks ;

void display ()

cout << "In C mark are : " << C\_marks ;  
cout << "<sup>CPP</sup> marks are : " << CPP\_marks ;  
add = C\_marks + CPP\_marks ;  
d = (float) add / 200 ;

$\text{percentage} = d * 100;$

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

?;

}

int main() {

s1.accept(); student s1;

s1.accept();

student :: marks m1;

m1.accept();

m1.display();

return 0;

}.

Output :-

Enter roll no & name of student : 17

parth

Enter C & CPP marks : 94

89

C marks : 94

CPP marks : 89

Percentage is: 92 %

Qn  
118

## \*Experiment no: 4 \*

1. WAP to find average of 2 results using friend function.

```
#include <iostream>
```

```
using namespace std;
```

```
class result2;
```

```
class result1;
```

```
{ public:
```

```
    int x;
```

```
    void accept(int a)
```

```
{
```

```
    x = a;
```

```
    friend void average(result1 r1, result2 r2);
```

```
}
```

```
class result2
```

```
{
```

```
public:
```

```
    int y;
```

```
    void accept(int a)
```

```
{
```

```
    y = a;
```

```
    friend void average(result1 r1, result2 r2);
```

```
}
```

```
void average(result1 r1, result2 r2)
```

```
{ float avg = (float)(r1.x + r2.y) / 2;
```

```
cout << "Average of 2 result is: " << avg;
```

```
int main()
{
    result1 r1;
    result2 r2;
    r1.accept(20);
    r2.accept(20);

    average(r1, r2);
    return 0;
}
```

Output:

Average of 2 results is : 20.

2. WAP to find greatest of 2 using friend function.

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

```
class A
{
```

```
public:
```

```
int x
```

```
{ void accept (int a)
```

```
    { x=a;
```

```
friend void greatest (A a1, B b1);
```

```
}
```

```
public:
```

```
int y;
```

```
{ void accept (int a)
```

```
    { y=a;
```

```
friend void greatest (A a1, B b1);
```

```
}
```

```
int g;
```

```
if(a1.x >= b1.y)
```

```
g = a1.x;
```

```
else
```

```
g = b1.y;
```

```
cout << "Greatest of the 2 numbers is : " << g;
```

```
{ int main()
```

```
{ A a1;
```

```
 B b1;
```

```
 a1 . accept (20);
```

```
 b1 . accept (15);
```

```
 return 0;
```

```
}
```

Output :-

Greatest of 2 numbers is : 20.

3.

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

```
{ public:
```

```
    int a, b;
```

```
    void accept (int x, int y)
```

```
{
```

```
    a = x;
```

```
    b = y;
```

```
}
```

```
friend void swap (Number n1);
```

```
{
```

```
    void swap (Number n1)
```

```
{
```

```
    int temp;
```

```
    temp = n1.a;
```

```
    n1.a = n1.b;
```

```
    n1.b = temp;
```

~~cout << "Numbers after swapping are: " << n1.a << n1.b;~~

~~"b" << n1.b;~~

```
{
```

```
int main()
```

```
{
```

```
    Number n1;
```

```
    n1.accept (2, 4);
```

```
    Swap (n1);
```

```
    return 0;
```

```
{
```

Output :-

Number after swapping are

$$a = 4$$

$$b = 2$$

4] WAP to swap numbers of different class using friend fun.

```
#include <iostream>
using namespace std;
class B;
class A
```

public:

```
int num1;
void accept(int a)
{
    num1 = a;
```

```
friend void swap(A a1, B b1);
};
```

```
class B
```

public:

```
int num2;
void accept(int a)
{
    num2 = a;
```

~~```
friend void swap(A a1, B b1);
};
```~~~~```
void swap(A a1, B b1)
```~~

~~num1~~ int temp;

~~temp = a1.num1,~~

~~a1.num1 = b1.num2;~~

~~b1.num2 = temp;~~

cout << "After swapping numbers are : a = " << a1.num1 <<  
b1.num2;

```
{  
    int main ()  
    {  
        A a1;  
        B b1;  
        a1.accept(4);  
        b1.accept(3);  
        swap(a1,b1);  
        return 0;  
    }  
}
```

\* Output :-

~~After swapping  
num1 is : 40  
num 2 is : 20.~~

Qn  
1118

5] Write a program for swapping number of same class.

→ #include <iostream>

using namespace std;

class A

{

public:

int a;

void accept(int x)

{

    a = x;

}

void swap(A a1, B b1)

{  
    }

    int temp;

    temp = a1.a;

    a1.a = a2.a;

    a2.a = temp;

cout << "Number after swapping are: a=" << a1.a << "b:"

        << a2.a;

}

};

int main()

{

    A a1;

    A a2;

    a1.accept(5);

    a2.accept(7);

    a1.swap(a1, a2);

    return 0;

}

\* Output :-

Values after swapping are ~~a=1~~<sup>a=9</sup> = 7 b = 5.

## \* Experiment - 5 \*

i] Default Constructor

```
#include <iostream>
using namespace std;
class Number
{
    int i, num, sum=0;
public:
    Number()
    {
        num = 10;
    }
    void display()
    {
        for(i=0; i<=num; i++)
        {
            sum = sum + i;
        }
        cout << "sum = " << sum;
    }
};

int main()
{
    Number n1;
    n1.display();
    return 0;
}
```

Output : sum : 55

## ii] Parameterized Constructor.

→ #include <iostream>  
using namespace std;  
class number  
{

int n, i, sum=0;

public:

number(int num)  
{

n=num;

}

void display()  
{

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

{

sum = sum + i;

}

cout << "sum = " << sum;

{

};

int main()

{

number n1(10);

n1.display();

return 0;

}

Output:- Sum: 55

iii] Copy Constructor.  
→

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

```
{ int i, n, sum=0;
public:
```

```
    number (int num)
```

```
    { n = num;
```

```
    number (number & NN)
```

```
    { n = NN.n;
```

```
    void display()
```

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

```
        sum = sum + i;
```

```
    cout << "sum = " << sum;
```

```
};
```

```
int main ()
```

```
    number n1(10);
```

```
    number n2(n1);
```

```
    n2.display();
```

```
    return 0;
```

```
}
```

Output:- Sum: 55

2.

i] Default Constructor.

```
#include <iostream>
using namespace std;
class student
{
```

```
    string name;
```

```
    int roll;
```

```
public:
```

```
    student()
```

```
{
```

```
    name = Yati;
```

```
    roll = 15;
```

```
}
```

```
void display
```

```
{
```

```
    cout << "name is: " << name << endl << "roll is: " << roll;
```

```
}
```

```
};
```

```
int main()
{
```

~~student s1;~~~~s1.display()~~

```
    return 0;
```

```
}
```

Output:

name is: Yati

roll is: 15

## iii] Parameterised Constructor

```
→ #include <iostream>
using namespace std;
class student
{
```

```
    string name;
    int roll;
public:
    student (int n, int r)
```

```
    name = n;
    roll = r;
```

```
void display()
```

```
    cout << "name is: " << name << endl << "roll is: " <<
```

```
}
```

```
int main()
```

```
student s1 (Yati, 15);
s1.display();
return 0;
```

Output:-

Name is: Yati  
Roll is: 15

## ii] Copy Constructor

```
#include <iostream>
using namespace std;
class student
{
```

```
    string name;
    int roll;
public:
    student ()
```

```
{
```

```
    name = 'Yati';
    roll = 15;
```

```
}
```

```
student (student &st)
{
```

```
    name = st.name;
    roll = st.roll;
```

```
}
```

```
void display ()
```

```
{
```

```
cout << "name is: " << name << endl << "roll is: " << roll;
```

```
int main ()
```

```
{
```

```
    student s1;
```

```
    student s2 (s1);
```

```
    s2.display ();
```

```
    return 0;
```

```
}
```

Output: Name is: Yati  
roll is: 15

3.

i Default Constructor

→ #include <iostream>  
using namespace std;  
class College

{ string name, course;

int roll;

public:

{ college & (int n, string c, int r)

name = "Yati";

name = n;

course = "CSE";

course = c;

roll = 15;

roll = r;

}

void display()

{

cout << "Name is: " << name << "Course is: " << course << roll;

}

};

int main()

{

College C1 ("Yati", "CSE", 15);

College C2 ("Aayush", "CSE", 60);

C1.display();

C2.display();

return 0;

}

Output: Name is : Yati

Course is : CSE

Roll is : 15

Name : Aayush

Course : CSE

Roll is : 60

#4]

```
→ #include <iostream>
using namespace std;
class Num
```

{

```
int i, num, f = 1;
```

```
public:
```

```
Num()
```

{

```
num = 5;
```

}

```
Num(int n)
```

{

```
num = n;
```

}

```
Num(Num &nn)
```

{

```
num = nn.num;
```

}

```
void display()
```

{

```
for (i = 1; i <= num; i++)
```

```
f = f * i;
```

{

```
cout << "fact: " << f;
```

}

};

```
int main()
```

{

```
num n1;
```

```
n1.display();
```

```
Num n2(n1);
```

Qn  
17/9/25

```
n2.display();  
num n3(n2);  
n3.display();  
return 0;  
}
```

Output: fact: 120 fact: 5040 fact: 5040

## \*Experiment: 06\*

```
→ #include <iostream>
using namespace std;
class person {
protected:
    string name;
    int age;
};

class student : protected person {
public:
    void accept()
    {
        cout << "Enter name & age";
        cin >> name >> age;
        cout << "Enter the roll number : ";
        cin >> roll;
    }

    void display()
    {
        cout << "Name is :" << name << endl << "Age is :" << age << endl << "Roll is :" << roll;
    }
};

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

Output:-

enter name & age : Yati

17

Enter ~~roll~~ the roll number : 15

name : Yati

Age : 17

roll is : 15

2.]

```
→ #include <iostream>
using namespace std;
class Academic
```

{

protected:

int scores;

{;

class sports

{

protected:

int performance;

{;

class result: protected Academic, sports

{

int r;

public:

void accept()

{

cout &lt;&lt; "enter academic Score";

cin &gt;&gt; scores;

cout &lt;&lt; " enter sports performance";

cin &gt;&gt; performance;

{

void display()

{

r = score + performance;

cout &lt;&lt; " result is: " &lt;&lt; r;

{;

{;

```
int main()
```

```
{  
    result fl;  
    fl.accept();  
    fl.display();  
    return 0;  
}
```

3.]

```
→ #include <iostream>
using namespace std;
class vehicle
{
```

protected:

```
    string type;
};
```

```
class Electric_car : protected Car
{
```

int battery;

public:

```
    void accept()
{
```

```
        cout << "enter brand & number" model ;
```

```
        cin >> brand >> model;
```

```
        cout << "enter type:";
```

```
        cin >> type;
```

```
        cout << "enter battery size";
```

```
        cin >> battery;
```

```
}
```

void display()

```
    cout << "brand is:" << brand << endl << "model is:" <<
```

```
    model << endl;
```

```
    cout << "type is:" << type << endl << "battery is:"
```

```
    << battery;
```

```
}
```

```
};
```

int main()

```
{
```

Electric\_car el;

ei.accept();  
ei.display();  
return 0;  
}

Output:-

enter brand & model dodge  
challenger  
enter type : muscle CAR

4

```
→ #include <iostream>
using namespace std
class employee
{
```

protected:

```
String name;
int empID;
```

{;

```
class manager : protected employee
{
```

protected:

```
string department;
```

public:

```
void accept()
```

{

```
cout << "enter department";
```

```
cin >> department;
```

{

```
void display()
```

{

```
cout << "department is: " << department;
```

{;

```
class developer : protected employee
{
```

```
String programmingLang;
```

public:

```
void accept()
```

{

```
cout << "enter name & empID";
```

```
cin >> name >> empID;
```

```
cout << "enter the programming language:";  
cin >> programmingLang;  
}
```

```
void display()  
{
```

```
cout << "name is: " << name << endl << "empId" empId  
"language is: " << programmingLang;  
}  
};
```

```
int main()  
{
```

```
manager m  
developer d;  
m.accept();  
m.display();  
d.accept();  
d.display();  
return 0;  
}
```

5] #include <iostream>

using namespace std;

class Person

{

protected:

string name;

int age;

};

class student : protected person

{

};

class sports : protected student

{

protected:

int sportsScore;

};

class Academic

{

protected:

int marks;

};

class Results : public sports, & Academic

{

public:

void accept()

{

cout << "enter name:";

cin >> name.

cout << "enter age:";

cin >> age;

cout << "enter marks & sports performance"

cin >> marks >> sportscore;

```
void display()
```

```
{  
    cout << "Name is: " << name << endl << "Age is: " << age  
    cout << "Marks are: " << marks << endl << "Sport score is: " <<  
    << endl;  
    cout << "Total is: " << marks + sportscore;  
}  
};
```

```
int main()
```

```
{  
    Result r;  
    r.accept();  
    r.display();  
    return 0;  
}
```

6. #include <iostream>  
using namespace std;  
class collegestudent  
{

protected:

int student\_id;  
int college\_code;  
};

class Test : virtual protected collegestudent  
{

protected:

float percentage;  
};

class sports : virtual protected collegestudent  
{

protected:

char grade;  
};

class Result : protected Test, protected sports  
{

public:

void accept()  
{

cout << "enter id & college code : ";  
cin >> student\_id >> college\_code;  
cout << "Enter percentage & Grade : ";  
cin >> percentage >> grade;

void display{

cout << "student\_id = " << student\_id << "college\_code is. " <<  
college\_code << "Percentage is. " << percentage << "Sport  
Grade : " << grade ; }

```
int main()
```

```
{  
    Result r;  
    r.accept();  
    r.display();  
    return 0;  
}
```

~~Q1~~  
26/9/25

## \*Experiment NO: 07 \*

→ `#include <iostream>`  
`using namespace std;`  
`class Area`  
{

`public:`

`void area(int a)`  
{

`int r;`

`r = a * a;`

`cout << "Area of class is: " << r;`

}

`void area(int l, int b)`  
{

`int r;`

`r = l * b;`

`cout << "In Area of the lab is: " << r;`

}

`};`

`int main()`  
{

`Area a;`

`a.area(9);`

`a.area(9, 9);`

`return 0;`

}

**Output :-**

Area of the class is: 81

Area of the lab is: 81

2.

→ #include <iostream>  
using namespace std;  
class Num

{  
public:  
void sum(float a[])  
{  
int i;  
float s=0;  
for(i=0; i<5; i++)  
{  
s=s+a[i];  
}  
cout << "sum is: " << s;  
}

void sum(int b[])

{  
int i, s=0;  
for(i=0; i<10; i++)  
{  
s=s+b[i];  
}

cout << "\n Sum is: " << s;  
};

int main()

{  
Num n;  
float arr1[5]={1.0, 2.0, 3.0, 4.0, 5.0};  
int arr2[10]={1, 2, 3, 4, 5, 6, 7, 8, 9, 10};  
n.sum(arr1);

```
n.sum(arr2);  
return 0;  
}
```

Sum is : 15

Sum is : 55

3  
→ #include <iostream>  
using namespace std;  
class Num  
{  
public:  
 int a = 6;  
 void operator - () {  
 a = -a;  
 }  
 void display()  
 {  
 cout << "a:" << a;  
 }  
};  
int main()  
{  
 Num n;  
 -n;  
 n.display();  
}

\* Output :-  
a:- 5

4.

→ #include <iostream>  
using namespace std;  
class Num  
{

public:

int a = 5;

{ void operator ++ ()

a = ++a;

}

{ void display ()

cout << "a:" << a;

}

}

int main () {

Num n;

++n;

n.display();

}

Output:-

a: 6.

Q  
511

## \*Experiment : 08 \*

```
1. #include <iostream>
#include <string>
using namespace std;
```

```
class Concat {
    string str;
public:
```

```
    Concat(string s)
```

```
    str = s;
```

```
    void operator + (Concat obj)
```

```
    cout << "Concatenated string: " << str + obj.str;
```

```
}
```

```
int main()
```

```
{
```

```
    Concat s1 ("xyz");
```

```
    Concat s2("pqr");
```

```
    s1 + s2;
```

```
    return 0;
}
```

\* Output:

Concatenated string: xyzpqr

2. #include <iostream>  
#include <string>  
using namespace std;

```
class Logon {  
protected:  
    string name;  
    string password;  
public:  
    virtual void accept() {}  
    virtual void display() {}  
};
```

```
class Emaillogin : public Logon
```

```
{  
public:  
    void accept()  
}
```

```
    cout << "Enter Email: ";  
    cin >> name;  
    cout << "Enter Email Password: ";  
    cin >> password;  
}
```

```
{  
void display()  
}
```

```
    cout << "\n Email Login Details: \n";  
    cout << "\n Name: " << name << "\n Password: " << password  
    << endl;  
}
```

```
class membershiplogin : public Logon
```

```
{  
public:
```

void accept ()  
{

cout << "\n Enter membership Name :";  
cin >> name;  
cout << " Enter membership password ";  
cin >> password;  
}

void display ()  
{

cout << "\n Membership login Details : \n";  
cout << " name is : " << name << "\n Password  
password << endl;

}

int main ()  
{

Login \* Login ;  
Email/Login e ;  
MembershipLogin m ;

Login = & e ;

login → accept ();

login → display ();

login = & m ;

login → accept ();

login → display ();

return 0 ;

}

Q  
/ \

## \*Experiment: 09 \*

```
1. #include <iostream>
#include <fstream>
using namespace std;
int main()
{
    ifstream fin;
    ofstream fout;

    fin.open("source_file.txt", ios::in);
    if (!fin)
    {
        cout << "No such file" << endl;
        return 1;
    }
    else
    {
        fout.open("destination_file.txt", ios::out);
        if (!fout)
        {
            cout << "Unable to create destination file"
            endl;
            return 1;
        }
        char ch;
        while (fin.get(ch))
        {
            fout.put(ch);
        }
        cout << "File Copied successfully" << endl;
    }
}
```

fin.close();

fout.close();

}

return 0;

```
2 #include <iostream>
#include <fstream>
#include <string>
using namespace std;
int main()
{
    ifstream fin;
    ofstream fout;
    fin.open("source_file.txt", ios::in);
    if(!fin)
    {
        cout << "Could not open file" << endl;
        return 1;
    }
    fout.open("destinationfile.txt", ios::out);
    if(!fout)
    {
        cout << "Could not create file" << endl;
        fin.close();
        return 1;
    }
    string word;
    int word_count = 0;
    while(fin >> word)
    {
        word_count++;
    }
    cout << "Word Count is: " << word_count << endl;
    fin.close();
    fout.close();
    return 0;
}
```

3 & 4. #include <iostream>  
#include <fstream>  
#include <string>  
using namespace std;

int main ()  
ifstream fin ;  
ofstream fout ;  
string filenameIn = "Input.txt" ;  
string filenameOut = "Output.txt" ;  
fin.open(filenameIn.c\_str());  
fout.open(filenameOut.c\_str());

if (!fin)  
{  
 cout << "Error opening input file" ;  
 return 1 ;  
}

String word, searchWord ;

while (fin >> word)  
{

wordCount ++  
if (word == searchWord)

occ ++;

fout << "Total words:" << wordCount << endl ;  
fout << "Occurrences of " << searchWord << " " << endl ;  
fin.close();  
fout.close();  
cout << "Result written to" << file nameOut << endl

```
    return 0;
```

## \* Output

Total Words : 3

Occurrences of "Hello": 2.

Qn  
511

\*Experiment no: 10\*

1.]

```
#include <iostream>
using namespace std;
template <class T> T sum(T arr[], int n)
{
    int i;
    T s = 0;
    for (i = 0; i < n; i++)
    {
        s = s + arr[i];
    }
    return s;
}

int main()
{
    int b = 5;
    int a[] = {1, 2, 3, 4, 5};
    cout << "Sum = " << sum(a, b);
    return 0;
}
```

2. #include <iostream>

using namespace std;

template <class T> T

Square (T a)

{

T s = a \* a;

return s;

}

template <> string square

<string>(string a)

{

return (a + a)

}

int main()

{

int x = 5;

string y = "abc";

cout << "square = " << square(x);

cout << endl << "square = " << square(y);

return 0;

3.

3. #include <iostream>  
using namespace std;  
template <class T1, class T2> class calc  
T1 a;  
T2 b;  
public:  
calc(T1 x, T2 y)  
{  
a = x;  
b = y;  
}  
void func()  
{  
cout << "add, sub, Mul, div, exit";  
int ch;  
while (1)  
{  
cout << "In enter your choice:";  
cin >> ch;  
switch (ch)  
{  
case 1: cout << "In sum:" << a + b;  
break;  
case 2: cout << "In diff:" << a - b;  
break;  
case 3: cout << "In Mul :" << a \* b;  
break;  
case 4: cout << "In div :" << a / b;  
break;  
case 5:  
return;  
default: cout << "In wrong choice";  
}

3  
2  
3;  
};

int main()  
{

calc < int, int > c1(2, 3);  
c1.func();

calc < int, float > c2(2, 4.3);  
c2.func();

calc < float, float > c3(2.1, 3.4);  
c3.func();

Q  
still

## \*Experiment : 11 \*

1.

```
→ #include <iostream>
#include <vector>
using namespace std;
int main()
{
    vector<int> v = {1, 2, 3, 4, 5};
    int index = 2;
    int newvalue = 10;
    v[index] = newvalue;
    int scalar = 3;
    for(int i = 0; i < v.size(); i++)
    {
        v[i] *= scalar;
    }
    for(int i = 0; i < v.size(); i++)
    {
        cout << v[i];
        if(i != v.size() - 1) cout << ", ";
    }
    cout << endl;
    return 0;
}
```

## \*Experiment : 12 \*

```
1. #include <iostream>
#include <stack>
#include <ctype>
using namespace std;
int main()
{
    stack<int> v;
    v.push(1);
    v.push(2);
    v.push(3);
    v.push(4);
    v.push(5);

    if(v.empty())
    {
        cout << "In stack is empty";
    }
    else
    {
        cout << "In stack is not empty";
    }

    cout << "In size :" << v.size();
    cout << "In topmost :" << v.top();

    cout << "In stack:";

    while(!v.empty())
    {
        cout << v.top() << " ";
        v.pop();
    }

    cout << "In size after popping) :" << v.size();
}
```

2. #include <iostream>  
#include <queue>  
#include <type>  
using namespace std;  
int main()  
{

queue <int> v;

v.push(11);

v.push(22);

v.push(33);

v.push(44);

v.push(55);

if(v.empty())

{ cout << "In queue is empty";

}

else {

cout << "In queue is empty";

}

else {

cout << "In queue is not empty";

cout << "In size:" << v.size();

cout << "In front:" << v.front();

cout << "In back:" << v.back();

cout << "In queue";

while(!v.empty())

{

cout << v.front() << " ";

{

v.pop();

cout << "In size(after popping):" << v.size();

## \*Experiment no: 11\*

(2)

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

```
int main()
{ vector<int> v = { 1, 2, 3, 4, 5 };
```

```
int index = 2;
int newValue = 10;
vector<int> :: iterator it_modify = v.begin();
it_modify = index;
* it_modify = newValue;
```

```
int scalar = 3;
for (vector<int> :: iterator it = v.begin();
     it != v.end(); it++)
{
    * it = scalar;
```

```
for (vector<int> :: iterator it = v.begin();
     it != v.end(); it++)
{
```

```
cout << * it;
if (it != v.end() - 1)
```

```
{ cout << "1" ; }
```

```
cout << endl;
return 0; }
```

Ques  
11/11