

Exercise :

1. Write a C++ program to add 2 numbers
2. Arithmetic Operations ($+$, $-$, $*$) using switch
3. Check if even or odd
4. Print 1 to 10 no.s using for loop
5. Print 1 to 10 no.s using while loop
6. Print below code

a) * b) 1 c) 1
* * 1 2 2 2
* * * 1 2 3 3 3 3
 1 2 3 + 4 4 4 +
 1 2 3 4 5 5 5 5 5 5

I

```
#include <iostream>
using namespace std;
int main()
{
    int num1, num2;
```

```
cout << "Enter any 2 numbers : ";
cin >> num1 >> num2;
```

~~cout << "The sum of the numbers is : " <<~~
~~num1 + num2;~~

```
return 0;
}
```

* OUTPUT :

Enter any 2 numbers : +5

54

The sum of the numbers is: 99

II

```
#include <iostream>
using namespace std;
int main()
{
    int num1, num2;
```

```
cout << "Enter any 2 numbers : ";
cin >> num1 >> num2;
```

```
cout << "Enter operator : ";
cin >> o;
```

switch (o)

{

case '+':

```
cout << "Sum is : " << num1 + num2;
break;
```

case '-':

```
cout << "Difference is : " << num1 - num2;
break;
```

* OUTPUT

Enter

7

Enter

Prod

III

#inclu

using

int

g

int n

cout <

ari >

if Cn

g

cou

g

V

```
#include <iostream>
using namespace std;
int main()
{
```

```
    int i;
```

```
    while (i != 11)
    {
        cout << i << " ";
        i++;
    }
    return 0;
}
```

* OUTPUT :

```
1 2 3 4 5 6 7 8 9 10
```

VI

```
b] #include <iostream>
```

```
using namespace std;
int main()
```

```
{
```

```
    int i, j;
```

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

```
        for (j = 1; j <= i; j++)
        {
```

```
            cout << j; cout " "
```

```
}
```

else

{

cout << "The number is odd";

}

return 0;

{

* OUTPUT :

Enter any number : 5798972

The number is even.

V

#include <iostream>

using namespace std;

int main()

{

int i;

for (i=1; i<=10; i++)

{

cout << i << " ";

}

return 0;

}

* OUTPUT :

1 2 3 + 5 6 7 8 9 10

V

*

VI

6]

8]

11]

case '*':

cout << "Product is : " << num1 * num2 ;
break ;

case '/':

cout << "Division is : " << num1 / num2 ;

break

3

return 0;

3

* OUTPUT :

enter any 2 numbers : 6

7

Enter operator : *

Product is : +2

III #include <iostream>

using namespace std ;

int main()

3

int num;

cout << "Enter any number : " ;

cin >> num ;

2 ;

if (num % 2 == 0)

3

cout << "The number is even " ;

3

EXPERIMENT 1

1. Write a program having class student having data members as roll no and name. Accept and display data for 1 object
2. Write a program:
to declare a class book having data members as book name, price, no of pages. Accept this data for 2 objects and display the name of book having greater price.
3. Write a program:
to declare a class time. Accept the time in HH:MM:SS format and convert it into total seconds and display them.

II

I

```
#include <iostream>
using namespace std;
class student
{
    int roll_no;
    string name;
public:
    void details()
    {
        cout << "Enter the roll number of a student : ";
        cin >> roll_no;
        cout << "Enter name of the student : ";
        cin >> name;
    }
    void display()
    {
        cout << "Roll No. : " << roll_no << endl;
        cout << "Name : " << name << endl;
    }
}
```

1
 2 2
 3 3 3
 4 4 4 4
 5 5 5 5 5

a] #include <iostream>
 using namespace std;

int main ()

{

int i, j, k;

for (i=1; i<=3; i++)

{

for (j=1; j<=3-i; j++)

{

cout << " ";

}

for (k=1; k<=i; k++)

{

cout << " ";

~~if (k < i)~~

~~cout << " ";~~

~~{~~

cout << "\n";

}

return 0;

}

© 97/05

* OUTPUT :

*
* *
* * *

```
cout << "ln";
```

```
}
```

```
return 0;
```

```
}
```

* OUTPUT :

```
1
```

```
1 2
```

```
1 2 3
```

```
1 2 3 4
```

```
1 2 3 4 5
```

```
c] #include <iostream>
using namespace std;
int main()
```

```
{
```

```
int i,j;
```

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

```
{
```

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

```
{
```

~~```
cout << i; cout << " ";
```~~~~```
}
```~~~~```
cout << "ln";
```~~~~```
{
```~~

```
return 0;
```

```
}
```

* OUTPUT :

III

```
#include <iostream>
```

```
using namespace std;
```

```
class Time
```

```
{
```

```
public:
```

```
int H, M, S, total;
```

```
void details();
```

```
{
```

```
cout << "Enter no of hours : ";
```

```
cin >> H;
```

```
cout << "Enter no of minutes : ";
```

```
cin >> M;
```

```
cout << "Enter no of seconds : ";
```

```
cin >> S;
```

```
}
```

```
};
```

```
int main()
```

```
{
```

```
Time t1;
```

```
t1.details();
```

~~```
cout << "time is : " << t1.H << ":" << t1.M << ":"
```~~~~```
<< t1.S;
```~~

```
t1.total = t1.H * 3600 + t1.M * 60 + t1.S;
```

```
cout << "in total time in seconds is "
```

```
<< t1.total;
```

```
return 0;
```

```
{
```

int main()

{

 books b1;

 b1.details();

 book b2;

 b2.details();

if (b1.price > b2.price)

{

 cout << b1.name; cout << " is more
 expensive ");

}

else

{

 cout << b2.name; cout << " is more
 expensive ");

}

return 0;

}

*

OUTPUT

Enter name of the book : hello

Enter the price of the book : 199

Enter the no of pages of the book : 200

Enter name of the book : World

Enter the price of the book : 599

Enter the no of pages of the book : 600

World is more expensive.

III

{

cout << "Roll no : " << roll_no ;

cout << "\nName : " << name ;

381;

int main ()

{

st . details () ;

st . display () ;

return 0 ;

}

*** OUTPUT**

Enter roll no of a student : 1

Enter name of the student : ekta

Roll no : 1

Name : ekta.

II #include <iostream>

using namespace std ;

class books

{

public :

int price , pages ;

string name ;

void details ()

{

cout << "Enter name of the book : " ;

cin >> name ;

cout << "Enter the price of the book : " ;

cin >> price ;

cout << "Enter the no of pages of the
book : " ;

cin >> pages ;

{

{}

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

{

```
    if (c[i].population > maxpop)
```

{

```
        maxpop = i;
```

{

{

```
cout << "The city with the highest  
population is: " << c[maxpop].name;
```

```
return 0;
```

{

* OUTPUT :

Enter city name : Pune

Enter population of the city : 10000

Enter city name : Vatican

Enter population of the city : 100

Enter city name : Mumbai

Enter population of the city : 200000

Enter city name : Banaras

Enter population of the city : 3000

Enter city name : Leh

Enter population of the city : 40000

The city with the highest population is:
Leh

Experiment 2

1. WAPP to declare a class city having data members as name & population. Accept this data for 6 cities and display name of city having highest population.

```
#include <iostream>
using namespace std;
class city
{
public:
    string name;
    int population;
    void details()
    {
        cout << "Enter city name : ";
        cin >> name;
        cout << "Enter population of the city : ";
        cin >> population;
    }
};

int main()
{
    city c[5];
    int i, maxpop = c[0].population;
    for (i=0; i<5; i++)
    {
        c[i].details();
        if (c[i].population > maxpop)
            maxpop = c[i].population;
    }
    cout << "The city with maximum population is " << c[maxpop].name;
}
```



OUTPUT

Enter no of hours : 06

Enter no of minutes : 45

Enter no of seconds : 24

time is : 6:45:24

~~total time in seconds is 24324~~

Qn
29/7/25

Enter account no & balance : 17

900

Enter account no & balance : 18

1000

Enter account no & balance : 19

1000

The accounts which received interest are :

5500.

```
cout << "The accounts which received  
interest are : ";
```

```
for (i=0; i<2; i++)  
{
```

```
if (a[i].balance >= 5000)
```

{

```
b = a[i].balance + 0.1 * a[i].balance;
```

```
cout << b << a[i].accountno;
```

{

}

```
return 0;
```

{

* OUTPUT:

Enter account no & balance : 10

300

Enter account no & balance : 11

400

Enter account no & balance : 12

500

Enter account no & balance : 13

600

Enter account no & balance : 14

5000

Enter account no & balance : 15

700

Enter account no & balance : 16

800

Enter acc

900

Enter acc

1000

Enter acc

1000

The acco

5500

2. WAP to declare a class 'Account' having data members as Account no. and balance accept this data for 10 accounts and give interest of 10% where balance is equal or greater than 5000 and display them.

```
#include<iostream>
using namespace std;
class account {
public:
    int acc_no;
    int balance;

    void accept() {
        cout << "Enter account no & balance: ";
        cin >> acc_no >> balance;
    }

    int main() {
        account a[10];
        int i, b;
        for (i = 0; i < 10; i++) {
            if (a[i].balance >= 5000)
                a[i].accept();
        }
    }
}
```

Experiment 3

1. WAP to declare a class 'Book' containing data members as book_title, author_name and price. Accept and display the info for one object using pointer to that object.

```
#include <iostream>
using namespace std;
class Book
{
    int price;
    string book_title;
    string author_name;
public:
    void accept()
    {
        cout << "Enter book title, author name
                and price of your book : ";
        cin >> book_title >> author_name >> price;
    }
    void display()
    {
        cout << "In Book title is : " << book_title;
        cout << " In Name is : " << author_name;
        cout << " In Price is : " << price;
    }
}
```

{

if(Cm == 0)

{

cout << "There is no HOD";

{

return 0;

{



OUTPUT

Enter your name & post : Scarlett
Manager

Enter your name & post : Ekta
Manager

Enter your name & post : Savali
Manager

Enter your name & post : Aanya
Manager

Enter your name & post : Denyani
HOD

Denyani is HOD

Pm

29/7/28

3. WAP to declare class staff having data members as name & post. accept details for 5 members and display persons with post HOD.

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

int main() {
    staff s[5];
    int i, m=0;
    for (i=0; i<5; i++) {
        s[i].accept();
    }
    for (i=0; i<5; i++) {
        if (s[i].post == "HOD") {
            cout << s[i].name << " is HOD" << endl;
            m = 1;
        }
    }
}
```

```
student s1;  
s1.accept C7, "devyani");  
s1.display C7);
```

```
return 0;  
}
```

* OUTPUT

Roll no is : 7

Name is : devyani

2. WAP to declare a class student having data members as roll_no and percentage using 'this' pointer invoke member function to accept the data and display this data for 1 object of a class.

```
#include <iostream>
using namespace std;
class student
{
    int roll_no;
    string name;
public:
```

```
void accept (int roll_no, string name)
```

```
{
```

```
    this -> roll_no = roll_no;
```

```
{
```

```
    this -> name = name;
```

```
}
```

```
void display()
```

```
{
```

```
    cout << "Roll no is : " << this -> roll_no;
```

```
{
```

```
    cout << "In Name is : " << this -> name;
```

```
}
```

```
};
```

```
int main()
```

```
{
```

not main()

{

Book b1;

Book *p = &b1;

p → accept();

p → display();

return 0;

{



OUTPUT

Enter book title, author name and price
of your book : harryPotter
JKRowling
450

~~Book title is : harryPotter~~

~~Name is JKRowling~~

~~Price is : 450~~

Experiment 4

classmate

Date _____

Page _____

1. WAP to swap 2 numbers from same class using object as function argument (swap function as member function)

```
#include <iostream>
using namespace std;
class A
{
    int a;
public:
    void accept (int a)
    {
        a = x;
    }
    void swap (A a1, A a2)
    {
        int temp;
        temp = a1.a;
        a1.a = a2.a;
        a2.a = temp;
    }
    cout << "Values after swapping
    are : num1 = " << a1.a <<
    "In num2 = " << a2.a;
}
int main()
```

Void display C)

{

cout << "In C marks are : " << cmarks ;

cout << "In C++ marks are : " << cpp_marks ;

add = c_marks + cpp_marks ;

d = (float)add / 200 ;

percentage = d * 100 ;

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

{

};

};

int main C)

{

student s1 ;

s1.accept C) ;

student :: marks m1 ;

m1.accept C) ;

m1.display C) ;

return 0;

*

OUTPUT

Enter roll no and name of student : 7

devyani

Enter C and C++ marks : 97

87

C marks are : 97

C++ marks are : 87

Percentage is : 92 %

Ques
30/7/25

8. Nested class program

```

#include <iostream>
using namespace std;
class student
{
    int roll_no;
    string name;
public:
    void accept()
    {
        cout << "enter roll no and name of student";
        cin >> roll_no >> name;
    }
    class marks
    {
        int cpp_marks;
        int c_marks;
        float percentage;
        int add;
        float d;
    public:
        void accept()
        {
            cout << "enter C and C++ marks : ";
            cin >> c_marks >> cpp_marks;
        }
    }
}

```

```
A a1;  
A a2;  
a1 accept(5);  
a2. accept (4);  
swap (a1, a2);
```

```
return 0;  
}
```

* OUTPUT

Values after swapping are : num1 = 4
num2 = 5

2. WAP to swap 2 numbers from same class using friend function

```
#include <iostream>
using namespace std;
class A
{
    int a;
public:
    void accept (int x)
    {
        a = x;
    }
    friend void swap (A a1, A a2);
    void swap (A a1, A a2)
    {
        int temp;
        temp = a1.a;
        a1.a = a2.a;
        a2.a = temp;
    }
    cout << "Values after swapping
    are : num1 = " << a1.a
    << "Inum2 = " << a2.a;
}
int main()
{}
```

{

A a1;

A a2;

a1 accept C5);

a2 accept C4);

a1 swap C a1, a2);

return 0;

{

X

OUTPUT

Values after swapping are : num1 = 4
num2 = 5

4 WAP to create 2 classes Result1 and Result2 which stores the marks of the students. Read the value of a mark for both the class objects and compute the average of the results

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

class result2;

```
class result1;
```

```
{
```

```
int x;
```

```
public:
```

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

```
{
```

```
x = a;
```

```
}
```

```
friend void average (result1 r1, result2
```

```
r2);
```

```
{
```

~~class result2~~

```
{
```

```
int y;
```

```
public:
```

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

```
{
```

```
y = a;
```

```
}
```

function
m
function

```
int temp;
temp = a1.num1;
a1.num1 = b1.num2;
b1.num2 = temp;
```

```
cout << "After swapping num1 is : " << a1.num1
      << " And num2 is : " << b1.num2;
}
```

```
int main()
{
```

```
    A a1;
```

```
    B b1;
```

```
    a1.accept(C20);
```

```
    b1.accept(C40);
```

```
    swap(a1, b1);
}
```

```
return 0;
}
```

*
~~OUTPUT~~

After swapping
num1 is : 10

And num2 is : 20

3. WAP to swap 2 numbers from different class using friend function

```
#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)
{
}
```

{

```
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 (C2);
```

```
b1.accept (C4);
```

```
greatest (a1, b1);
```

{

* OUTPUT

Greatest of the 2 numbers is : 4

Q
1118125

5. WAP to find the greatest number among 2 numbers from 2 different classes using friend function

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

```
class A
```

```
{
```

```
    int x;
```

```
public:
```

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

```
{
```

```
    x = a;
```

```
}
```

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

```
class B
```

```
{
```

```
    int y;
```

```
public:
```

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

```
{
```

```
    y = a;
```

```
}
```

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

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

```
friend void average (Cresult r1, result r2);  
{  
    void average (Cresult r1, result r2)  
    {  
        float avg = (float)(r1.x + r2.y)/2;  
        cout << "Average of the 2 results is : "  
        << avg;  
    }  
    int main ()  
    {  
        result r1;  
        result r2;  
        r1.accept (C2D);  
        r2.accept (C2D);  
  
        average (r1, r2);  
        return 0;  
    }  
}
```

* OUTPUT

Average of the 2 results is : 20

3 #include <iostream>
 using namespace std;
 class Number {
 int a, b, sum;
 public:
 Number() {
 a = 10;
 b = 20;
 }
 }

Number C const Number &n1) {

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

void display() {

sum = a + b;

cout << "Sum = " << sum;
 }

}

~~int main()~~

Number n1;

n1.display();

Number n2(n1);

n2.display();

return 0;

4 #include <iostream>
 using namespace std;
 class Number {
 int a, b, sum;
 public:
 Number() {
 a = 10;
 b = 20;
 }
 }

★

OUTP

Name _____

Percentage _____

★

OUTPUT

Sum = 30

Sum = 30

```
2 #include <iostream>
using namespace std;
class Number {
    int a, b, sum;
public:
    Number(int x, int y) {
        a = x;
        b = y;
    }
    void display() {
        sum = a + b;
        cout << "Sum = " << sum;
    }
};

int main()
{
    Number n1(100, 200);
    n1.display();
    return 0;
}
```

* OUTPUT

Sum = 300

Experiment 5

```
1 #include <iostream>
using namespace std;
class Number {
public:
    Number() {
        a=10;
        b=20;
    }
    void display() {
        sum = a+b;
        cout << "Sum = " << sum;
    }
};

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

* OUTPUT

Sum = 30

```
2 #include <iostream>
using namespace std;
class Number {
public:
    Number() {
        a=10;
        b=20;
    }
    void display() {
        cout << "Sum = " << sum;
    }
};

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

* OUTPUT

Sum

```
6 #include <iostream>
using namespace std;
class Student {
    float percentage;
    string name;
public:
    Student(string n, float p) {
        name = n;
        percentage = p;
    }
    Student(Student &s) {
        name = s.name;
        percentage = s.percentage;
    }
    void display() {
        cout << "Name = " << name << endl
            Percentage = " << percentage;
    }
};

int main() {
    Student s1("Devyani", 98);
    s1.display();
    Student s2(s1);
    s2.display();
    return 0;
}
```

* OUTPUT

Name = Devyani
Percentage = 98
Name = Devyani
Percentage = 98

5 #include <iostream>

using namespace std;

class Student {

float percentage;

string name;

public:

student C string n, float p) {

name = n;

percentage = p;

void display() {

cout << "Name = " << name << "

Percentage = " << percentage;

}

int main()

{

Student s1 ("Devyani", 98);

s1.display();

return 0;

}

*

OUTPUT

Name = Devyani

Percentage = 98

6 #include <iostream>

using namespace std;

class Student {

float percentage;

string name;

public:

student C string n, float p) {

name = n;

percentage = p;

void display() {

cout << "Name = " << name << "

Percentage = " << percentage;

}

int main()

{

Student s1 ("Devyani", 98);

s1.display();

return 0;

}

*

OUTPUT

Name = De

Percentage = 98

Name = De

Percentage = 98

```
4 #include <iostream>
using namespace std;
class Student {
    float percentage;
    string name;
public:
    Student() {
        name = "Devyani";
        percentage = 98;
    }
    void display() {
        cout << "Name = " << name << "In"
        Percentage = " << percentage;
    }
};

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

* OUTPUT

Name = Devyani
Percentage = 98

9 #include <iostream>
using namespace std;
class Num {
 int c, num; f=1;
public:
 Num() {
 num = 5;
 }

Num (int n) {
 num = n;
}

Num (Num &a) {
 num = a.num;
}

void display () {
 for (i=1; i<=num; i++) {
 f = f * i;
 }
 cout << "In fact = " << f;
}

int main () {
 Num n1;
 n1.display();
 Num n2 (7);
 n2.display();
 Num n3 (n2);
 n3.display();

```
8 #include <iostream>
using namespace std;
class College {
    string name, course;
    int roll;
public:
    College (int r, string n, string c = "Computer
        Science") {
        roll = r;
        name = n;
        course = c;
    }
    void display () {
        cout << "name: " << name << " | roll: "
        << roll << " | course: " <<
    }
    int main () {
        College c1(1, "Sara");
        College c2(c1);
        c2.display();
        College c3(c2, "Tara");
        College c4(c3);
    }
}
```

```

7. #include <iostream>
using namespace std;
class College {
    string name, course;
    int roll;
public:
    College(int r, string n, string c = "computer science");
    void display();
};

void College::display() {
    cout << "name : " << name << "\nroll : " << roll
        << "\ncourse : " << course;
}

int main() {
    Student s1(6, "Sita");
    s1.display();
    Student s2(7, "Gita");
    s2.display();
}

```

OUTPUT

name : Sita
 roll : 6
 course : computer science

name : Gita
 roll : 7
 course : Computer Science

* OUTPUT

Enter name, age and roll no : devyani

17

7

Name is: devyani

Age is: 17

Roll no is: 7

and

Experiment 6

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

class Student : protected Person {
    int roll_no;
public:
    void accept() {
        cout << "Enter name, age and
        roll no : ";
        cin >> name >> age >> roll_no;
    }

    void display() {
        cout << "Name is : " << name;
        cout << "Age is : " << age;
        cout << "Roll no is : " << roll_no;
    }
};

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

* OUTPUT
Enter
17
7
Name
Age
Roll

OUTPUT

fact: 120

fact: 6040

fact: 5040

P
1719

```
3 #include <iostream>
using namespace std;
class Vehicle {
protected:
    string brand, model;
};

class Car : protected Vehicle {
protected:
    string type;
};

class ElectricCar : protected Car {
int capacity;
public:
    void accept();
};

void accept() {
    cout << "Enter brand, model,
            type and capacity : ";
    cin >> brand >> model >> type >>
        capacity;
}

void display() {
    cout << "Brand is " << brand;
    cout << " Model is " << model;
    cout << " Type is " << type;
    cout << " capacity is " << capacity;
}

ElectricCar e1;
e1.accept();
e1.display();
return 0;
```

OUTPUT

Enter marks and sports score : 100
100

Total : 200

```
2 #include <iostream>  
using namespace std;  
class Academics {  
protected:  
    int marks;
```

```
};  
class Sports {  
protected:  
    int score;
```

```
};  
class Result : protected Academics,  
protected Sports {  
public:
```

```
    int total;  
    void accept() {
```

```
        cout << "Enter marks and score  
        score : ";  
        cin >> marks >> score;
```

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

```
        total = marks + score;
```

```
        cout << "Total : " << total;
```

```
    }
```

```
int main() {
```

```
{
```

```
    Result r1;
```

```
    r1.accept();
```

```
    r1.display();
```

```
    return 0;
```

```
}
```

```
int main()
{
    Manager m1;
    m1.accept();
    m1.display();
    Developer d1;
    d1.accept();
    d1.display();
    return 0;
}
```

~~Output~~ Enter name empID and department
devyani

07

CSE

Name is devyani

empid is 7

department is CSE

Programming language

Python

Programming language is Python

+ #include <iostream>
using namespace std;
class Employee {
protected:
 string name;
 int empID;
};
class Manager : protected Employee {
public:
 string department;
 void accept() {
 cout << "enter name, empID &
 department";
 cin >> name >> empID >> department;
 }
 void display() {
 cout << "Name is " << name;
 cout << "Empid is " << empID;
 cout << "Department is " << department;
 }
};
class Developer : protected Employee {
public:
 string p_l;
 void accept() {
 cout << "Programming language";
 cin >> p_l;
 }
 void display() {
 cout << "Programming language is "
 << p_l;
 }
};

~~g~~ OUTPUT

Enter brand, model, type and capacity :
 ~~escada~~

mercedis

c-class

sedan

o

Brand is mercedes

Model is c-classes

Type is sedan

capacity is 0

7 8

add,
"j
pe >>

nd
det
spec
less capacity

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 student ID: ";

cin >> student_id;

cout << "Enter college code: ";

cin >> college_code;

cout << "Enter Percentage: ";

cin >> percentage;

cout << "Enter Grade: ";

cin >> grade;

{

void display () {

cout << "Student ID: " << student_id;

cout << "College code: " << college_code;

```
classmate  
Date _____  
Page _____
```

```
classmate  
Date _____  
Page _____
```

```
void display() {  
    cout << "Name : " << name << "Age : "  
        << age;  
    cout << "Marks : " << marks;  
    cout << "Sports Score : " << sportsScore;  
    cout << "\nTotal : " << marks + sportsScore;  
}  
int main() {  
    Result r;  
    r.accept();  
    r.display();  
    return 0;  
}
```

* OUTPUT

```
Enter name: Devyani  
Enter age: 17  
Enter marks: 50  
Enter sports score: 50;
```

```
Name: Devyani  
Age: 17  
Marks: 50  
Sports Score: 50  
Total: 100
```

5 #include <iostream>
using namespace std;

class Person {

protected:

string name;

int age;

{}

class Student : protected Person {

{}

class Sports : public Student {

protected:

int sportsScore;

{}

class Academics : public Student {

protected:

int marks;

{}

class Result : public Sports, public Academics {

public:

void accept() {

cout << "Enter name : ";

cin > name;

cout << "Enter age : ";

cin > age;

cout << "Enter marks : ";

cin > marks;

cout << "Enter sports score : ";

cin > sportsScore;

{}

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

public:
void sum(float a[5])
{

int i;
float s=0;
for (i=0; i<5; i++)
{
s = s + a[i];
}

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

void sum(int b[10])
{

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);

Experiment 7

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

```
class Area
```

```
{
```

```
public:
```

```
void areaC(int a)
```

```
{
```

```
int r;
```

```
r = a * a,
```

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

```
}
```

```
void areaC(int l, int b)
```

```
{
```

```
int r;
```

```
r = l * b;
```

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

```
}
```

~~```
3 } main()
```~~~~```
{
```~~

```
Area a;
```

```
a.areaC(9).
```

```
a.areaC(9, 9);
```

```
return 0; }
```

X

OUTPUT

Area of the class is : 81

Area of the lab is : 81

```
cout << "Percentage : " << percentage  
cout << "Sports Grade : " << grade;  
33;  
int main() {  
    Result r;  
    r.accept();  
    r.display();  
    return 0;  
}
```

Qn
26/9/25

+ `#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`

~~Qn
16/10~~

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

public:
int a; };

read operator - () {
a = a; };

void display () {
cout << "a: " << a; };

};

int main () {

Num n;

-n;

n.display ();

};

* OUTPUT

a: -5

return 0;

}

*

OUTPUT

sum is : 15
sum is : 55

3

```
public :
```

```
void accept ()
```

```
{
```

```
cout << " \n Enter Membership Name : " ;
```

```
cin > name ;
```

```
cout << " \n Enter Membership Password : " ;
```

```
cin > password ;
```

```
}
```

```
void display ()
```

```
{
```

```
cout << " \n Membership login Details : " ;
```

```
cout << " Name : " << name << " \n Password : "
```

```
" << password << endl ;
```

```
}
```

```
};
```

```
int main ()
```

```
{
```

~~```
Login *login ;
```~~~~```
EmailLogin e ;
```~~~~```
MembershipLogin m ;
```~~

```
login = &e
```

```
login -> accept () ;
```

```
login -> display () ;
```

```
login = &m
```

```
login -> accept () ;
```

```
login -> display () ;
```

```
return 0 ;
```

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

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

class Emaillogin : public Ilogin {

public:  
 void accept ()  
{

cout << " Enter Email Name : " ;  
cin >> name;

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

void display ()  
{

cout << "\n Email login details : \n " ;  
cout << " Name : " << name << "\n "  
password : " << password << endl ;  
}

};

class Membershiplogin : public Ilogin  
{

# Experiment 8

```
1 #include <iostream>
#include <strings>
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

fin close();  
tout close();  
}  
return 0;  
}

# Experiment 9

```
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;
```

## \* OUTPUT

Enter Email Name : Scarlett@gmail.com  
Enter Email Password : hello123

Email login Details :

Name : Scarlett@gmail.com

Password : hello

Enter Membership Name : hehe

Enter Membership Password : 123

Membership login Details :

Name : hehe

Password : 123

Qn

16) ID

```
28+ #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;
int wordCount = 0, occ = 0;
```~~

```
cout << "Enter word to find
occurrence ";
cin >> searchWord;
```

```
while (fin >> word) {
    wordCount++;
    if (word == searchWord)
        occ++;
}
```

track();
putchar();

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 ("destination_file.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;
```

2. #include <iostream>
using namespace std;
template < class T > T square(T a)
{
 Ts = a * a;
 return s;
}
template < > string square < string >
(string a)
{
 return cata;
}
int main()
{
 int x = 5;
 string y = "abc";
 cout << "square = " << square(x);
 cout << "\nsquare = " << square(y);
 return 0;
}



OUTPUT

square = 25
square = abcabc

Experiment 10

```
1. #include <iostream>
using namespace std;
template< class T> T sum( T arr[], int no)
{
    int s=0,
        for ( i=0; i<no; 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;
}
```

* OUTPUT

sum = 15

```
    cout << "Total words : " << wordCount  
    << endl;  
    cout << "Occurrences of " << searchWord  
    " : " << occ << endl;  
    fri.close();  
    fout.close();  
    cout << "Result written to " <<  
    filenameOut << endl;  
    return 0;  
}
```

* OUTPUT

Total words : 3

Occurrences of "hello" : 2

On
3/11

Experiment 11

classmate

Date _____
Page _____

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;
}`

* OUTPUT

3, 6, 30, 12, 15

```
7;  
int main () {  
    calc < int, int > c1(c2, 3);  
    c1. func ();  
    calc < int, float > c2(c2, 4.8);  
    c2. func ();  
    calc < float, float > c3(c2.1, 3.4);  
    c3. func ();  
}
```

* OUTPUT

add, sub, mult, div, exit
enter your choice : 1

sum = 5

enter your choice : 5

add, sub, mult, div, exit

enter your choice : 2

diff = 2.3

enter your choice : 5

add, sub, mult, div, exit

enter your choice : 3

mul : 714

enter your choice :-

Qn
511

① #include
#include
using

int
vec

int
vect
vec

int
for

3

for

3

com

retu

3

* OUTPUT

3, 6,

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

Experiment 11

with iteration

(2)

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

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

```
int index = 2;
int newValue = 10;
```

~~✓~~ ~~index = 7 -~~

```
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;
}
```

~~(Output)~~

```
for (vector <int> :: iterator it = v.begin();
     it != v.end(); it++) {
    cout << *it;
    if (it != v.end() - 1) {
        cout << ", ";
    }
}
```

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

OUTPUT

3, 6, 30, 12, 15

2 #include <iostream>
include <queue>
include < ctype >
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 12

classmate

Date _____
Page _____

```
1 #include <iostream>
# include <stack>
# include <cctype>
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 << "\n size : " << v.size();
        cout << "\n topmost : " << v.top();
        cout << "\n stack : ";
        while(v !empty())
        {
            cout << v.top() << " ";
            v.pop();
        }
        cout << "\n size after popping : " << v.size();
    }
}
```