

Q3) #include <iostream>
using namespace std;

```

class Collage {
    int roll;
    string name;
    string course;
public:
    Collage(int r=00, string n="Unknown", string c="Computer Engineering") {
        roll = r;
        name = n;
        course = c;
    }
    void display() {
        cout << "Roll Number: " << roll << endl;
        cout << "Name: " << name << endl;
        cout << "Course: " << course << endl;
    }
};

int main() {
    Collage S1(39, "Ranvir");
    Collage S2(53, "Sahil");
    S1.display();
    S2.display();
    return 0;
}

```

Q4)

Experiment - 6

a) Include <iostream>
using namespace std;

```
class department {
protected:
    string dname;
};
```

```
class student : protected department {
protected:
    string Sname;
    int roll;
```

```
class marks : protected student {
int m1, m2, percentage; student
```

public:

```
void accept () {
```

```
cout << "Enter department: " << endl;
cin >> cname;
```

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

```
cout << "Enter marks 1: " << endl;
cin >> m1;
```

```
cout << "Enter marks 2: " << endl;
cin >> m2;
```

void calculate () {

```
int per = (m1 + m2) / 2
```

```
cout << "Department: " << cname << endl;
```

Output:
Total time = 25627

~~On
BT1~~

3 () Angle Unit

" : sum print <> time)

sum << (ii)

(←→)

introducing " >> time)

introducing (<< ii)

{ }

3 () min. val

: [2] > (ii)

O = xMax. val

3 (++i) : z>i : O = i[n]) val

" >> i >> " (i) n / " >> time)

D() Angle[i][i])

[value] < introducing [i]) (i

3 (maxAngle

4)

#include <iostream> #

using namespace std;

int main () {

cout << " Using for loop (1-10); "

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

cout << endl; cout << i << " ";

}

cout << endl; << endl;

{ } (++i, now => i - 1 = i - 1) not

{ } (cout = << " ") Using while loop (10-1);

; int > j = 10, now) {

while (j >= 1) {

; done > cout << j << " " ;

j -- ; }

cout << endl; }

return 0;

}

<< " "; }

nd;

(Q1) #include <iostream>
using namespace std;

```
class B;
class A {
    int numA;
```

public:

```
A (int n) { numA = n;
friend void swapNumbers(A&n, B&g);
void show () { cout << numA << " "; }
};
```

```
class B {
    int numB;
```

public:

```
B (int n) { numB = n; }
```

```
friend void swapNumbers(A&n, B&g);
void show () { cout << numB << " ";
};
```

int main ()
{
 student st; accept (&st);
 st.display ();
}

(input 20) student with
20 programmer prints

Output :

Enter name : Hanum Jaelha
Enter Roll no : 38

~~Student details:~~

Name : Hanum Jaelha
Roll No : 38

(J prints line)

i.e. : when we " >> true)

; (name, no) with <<

: when we " >> true)

; author << no)

(J prints line)

i.e. : when I Student? n / " >> true)

18 >> name >> " : name " >> true)

name >> author >> " : author " >> true)

Experiment 1:

- 1) WAP to declare a class Student having data members as name, Roll no. Accept & display data for one student

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

```
class Student
```

```
private:
```

```
String name;
```

```
int rollno;
```

```
public:
```

```
void accept();
```

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

```
getline (cin, name);
```

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

```
cin >> rollno;
```

```
}
```

```
void display()
```

```
{
```

```
cout << "In Student Data:\n";
```

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

```
cout << "Roll NO: " << rollno << endl;
```

```
}
```

```
};
```

Experiment - 12

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

stack <int> stack1;

void disp () {
    stack <int> temp = stack1;
    while (!temp.empty ()) {
        cout << temp.top () << " ";
        temp.pop ();
    }
    cout << endl;
}
```

```
int main () {
    int num;
    cout << "Enter a no : " << endl;
    cin >> num;

    for (int i = 0; i < num; i++) {
        int temp;
        cout << "Enter element at position " << i + 1 << endl;
        cin >> temp;
        stack1.push (temp);
    }
}
```

Void quo ()
 Count << "Quotient: " << n/y << endl;
 }

Void rem ()
 Count << "reminder << n%y << endl;"
 }

Void power ()
 Count << "X raised to Y: " << pow (x, y) << endl;
 }

Void min_num ()
 Count << "Min of number: " << fmin (n, y) << endl;

Void max_num ()
 Count << "Max of number: " << fmax (n, y) << endl;

Void sin_n ()
 Count << "Sin of n: " << sin (n) << endl;

Void cos_y ()

Count << "Cos of y: " << cos (y) << endl;

maxIndex = i; } // student[i] (i)
// maximum grade

cout << "In City with highest population
" << c[maxIndex].
name << endl;

} return 0; // main function

// read data from file "E:\C++\C++\Data.txt"

cin >> n >> m >> f;

// read data from file "E:\C++\C++\Data.txt"

cin >> name >> id >> g;

} // read data from file "E:\C++\C++\Data.txt"

if (code == 'r') { // read

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

{

{

if (code == 'w') { // write

cout << "NAME" >> name >> endl;

: grade >> endl;

: id >> endl;

{

{

Enter Experiment - 5

(Q2)

(Q1) #include <iostream>
using namespace std;

```
class number {
    int num;
public:
    number () {
        cout << "Enter a number : " << endl;
        cin >> num;
        int sum = 0;
        for (int i = 1; i <= num; i++) {
            sum = sum + i;
        }
        cout << "Sum of number upto " << num << " is "
             : " " << sum << endl;
    }
};
```

```
int main () {
    number n;
    return 0;
}
```

→ Output
 Enter a number : 3
 The sum of number upto
 3 is 6

→ 0

Experiment 8:

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

class Combine {
public:
    Combine & startingStr;
    string str = S;
};

Combine operator+(Combine &obj) {
    return Combine(str + obj.str);
}

void display() {
    cout << str << endl;
}

int main() {
    string s1("xyz"), s2("pqr"), s3 = s1 + s2;
    cout << "Concatenated string : ";
    s3.display();
}
```

(Q3) #include <iostream>
using namespace std;

class Teacher {

int experience;

public

Teacher (int e) {

experience = e;

}

void display () {

cout << "Experience : " << experience << endl;

 }

Varil operator - () {

 experience = - experience;

 }

int main () {

Teacher t(10);

t.display ();

+1;

cout << "After negation : " ;

t.display ();

return 0;

}

PAGE NO. / / /
DATE / / /

PAGE NO. / / /
DATE / / /

id << endl;
<< subject
<< endl;

(e4)

#include <iostream>
using namespace std;

class College {
protected:
string emp_name;
int id;
};

class Staff : public Employee {
string name;
int dept_id;

public:

void accept Emp () {

cout << "Enter Mgr. name: " << endl;
cin >> name;

cout << "Enter Emp. name: " << endl;

cin >> emp_name;

cout << "Enter ID: " << endl;

cin << Enter staff name: " << endl;
cin >> name;

cout << "Enter dept id: " << endl;

cin >> dept_id;

}

void display Emp () {

cout << "College: " << name << endl;

cout << "Emp name: " << emp_name << endl;

cout << "ID: " << id << endl;

cout << "Staff Name: " << name << endl;

cout << "Reportment ID: " << dept_id << endl;

```
int main () {  
    Complex a (2, 3);  
    Complex b (4, 5);  
  
    Complex c = add (a, b);  
  
    cout << "Sum of complex no: ";  
    c.show();  
    return 0;  
}
```

) Output:

Sum of Complex numbers : 6+8i

```
int main () {  
    Account a[10];  
  
    for (int i = 0; i < 10; i++) {  
        cout << "In Account " << (i + 1) << ":"  
        << endl;  
        a[i].input();  
        a[i].giveInterest();  
    }  
}
```

Credit << "In Account with balance = 50
000 after interest" >>

```
for (int i = 0; i < 10; i++) {  
    if (a[i].balance >= 5000)  
        a[i].display();  
}  
}
```

((main + n)

:5272 main

{(++i, 2 > i, 0 = i & n) ->

>> "Value of interest in account " >> now
>> bus >> i + j

5)

#include <iostream> with
using namespace std;

int main () {

>> cout << "Enter the first number" << endl;

cin >> num;

for (int i = 1; i <= num; i++) {

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

cout << i << endl;

}

} (i = < j) <|> i

>> i > cout << endl;

}

(-- i)

{

} (num >> num)

; O output

{

6)

→ Output

Roll number: 39

Name: Kanan

Course: Computer Engineering

Roll no: 53

Name: Sathan

Course: Computer Engineering

Q4)

#include <stdio.h>
using namespace std;

class student {

int roll;

string name;

public:

student () {

name = "Unknown";

roll = 0;

}

~~student (string n) {~~

name = n;

roll = 0;

}

student (string int) {

name = k;

roll = r;

void display () {

cout << "Name: " << name << endl;

```
int main () {  
    calc <int int> n (100, 200);
```

- n. sum ();
- n. diff ();
- n. pro ();
- n. quo ();
- n. rem ();
- n. pancer ();
- n. min_num();
- n. max_num();
- n. sin-n ();
- n. cos-y ();

Qn
11/11

PAGE NO. / /
DATE / /

a3) #include <iostream>
#include <iostream>
#include <string>
using namespace std;

int main() {
 ifstream file("First.txt");
 if (!file) {
 cout << "Error" << endl;
 return 1;
 }
 string word;
 int count = 0;
 while (file >> word)
 count++;
 cout << "Total words : " << count << endl;
 file.close();
 return 0;
}

}

class Gramma {

private :

 int e;

public :

 Grammer (int Val) {
 e = Val

}

 friend Val total sum (Alpha, Beta
 Gramma);

}

Val total sum (Alpha x, Beta y, Gramma
z);

int sum = x.a + y.b + z.c;

cout << "sum of all of value : " << sum << endl;

}

Alpha Obj1(10);

Beta Obj2(20);

Gramma Obj3(30);

total sum (Obj1, Obj2, Obj3);

return 0;

}

→

→ Output

sum of all values : 60.

cout << "Roll Number: " < roll < endl;

} ;
int main () {

Student S1;

student S2 ("12 anam", 39);

student S3 ("Sohan", 53);

S1. display();

S2. display();

S3. display();

return 0;

}

→ Output:

Name : Utkarsh

Roll no : 0

Name : Ishaan

Roll no :

Roll no : 0

Name : Sohan

Roll no : 53

Q
1111

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

#include <iostream>
using namespace std;

class Book

int id;

String name;

float price;

public:

void input()

cout << "Enter book ID : ";

cin >> id;

cout << "Enter book name : ";

cin >> name;

cin.ignore();

cout << "Enter book name : ";

getline(cin, name);

cout << "Enter book price : ";

cin >> price;

}

void display() const

cout << "Book ID : << id << endl;

cout << "Book name : << name << endl;

cout << "Book Price : << price << endl;

DE

(a) Class Result 29
 Recent marks;

```

public {
    void read() {
        cout << "Enter marks for Result 1: ";
        cin >> marks1;
    }
    friend void average(Result1, Result2);
};

void average(Result1, Result2) {
    float avg = (Result1.marks + Result2.marks) / 2;
    cout << "Average marks = " << avg << endl;
}

int main() {
    Result obj1;
    Result obj2;

    obj1.read();
    obj2.read();

    average(obj1, obj2);

    return 0;
}
    
```

→ Output:

Enter marks for Result 1: 100
 Enter marks for Result 2: 99
 Average marks = 99.5

Experiment 10

(Q1) #include <iostream>
 using namespace std;
 template <class T>

```
T sumArray(T arr[], int n) {
    T sum = 0;
    for (int i = 0; i < n) {
        sum = sum + arr[i];
    }
    return sum;
}
```

int main ()
 int iArr[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
 float fArr[10] = {1.1, 2.2, 3.3, 4.4, 5.5, 1.1, 2.2, 3.3,
 4.4, 5.5};
 double dArr[10] = {0.5, 1.5, 2.5, 3.5, 4.5, 0.5,
 1.5, 2.5, 3.5, 4.5};

cout << "Sum of Integer array: " << sumArray(iArr,
 10) << endl;

cout << "Sum of float array: " << sumArray(fArr,
 10) << endl;

cout << "Sum of Double arr: " << sumArray(dArr,
 10) << endl;

return 0;

}

Q7) #include <iostream>
 #include <cmath>
 using namespace std;

```
class Point {
private:
```

```
int x, y;
```

public:

```
Point (int xVal, int yVal) {
```

```
x = xVal;
```

```
y = yVal;
```

}

friend double calculateDistance (Point,
 Point);

}

double calculateDistance (Point p1, Point p2);

```
int dx = p2.x - p1.x;
```

```
int dy = p2.y - p1.y;
```

```
return sqrt (dx * dx + dy * dy);
```

int main () {

```
Point p1 (3, 4);
```

```
Point p2 (7, 17);
```

double distance = calculateDistance (p1, p2);

```

a2) #include <iostream>
#include <fstream>
using namespace std;

int main () {
    ifstream file ("First - test");
    if (file.is_open ()) {
        cout << "Error opening file" << endl;
        return 1;
    }

    char ch;
    int digit = 0, space = 0;
    while (file.get (ch)) {
        if (is digit (ch))
            digit++;
        else if (is space (ch))
            space++;
    }

    cout << "Digits : " << digits << endl;
    cout << "space : " << space << endl;

    file.close ();
    return 0;
}

```

3
 class student : protected College &
 string stu-name;
 int rall;

Public -

```
Void accept stud () {  

    cout << "Enter College name: " endl;  

    cin >> College;  

    cout << "Enter name: " endl;  

    cin >> stu-name;  

    cout << "Enter Roll number: " endl;  

    cin >> rall;
```

}

Void display stud () {

```
cout << "College :" << College << endl;  

cout << "Student name :" << stu-name <<  

endl;  

cout << "Roll Number :" << rall << endl;
```

}

8 int main () {

```
staff staff1;  

staff1. accept Emp();  

staff1. display Emp();
```

student stud1;

stud1. accept stud();

stud1. display stud();

} return 0;

Count << "Distance between points" >>
distance << endl;

return 0;
}

→ Output:
Distance between points

(28) #include <iostream>
using namespace std;
class Audit;

class Bank account {
private:
int balance;

public:
Bank Account (int b) {
balance = b
}

friend void audit Balance (Bank Account
Audit);

};

class Audit {

public:

void Start Audit (Bank Account acc)
audit Balance (acc, *this);
}

Cout << endl;
Cout << "Top most element : " << endl;
Stack¹. top () << endl;

Cout << endl;
Cout "Stack element (top to bottom)
<< endl;

disp ();
Cout << endl;
Cout << "Pop function : " << endl;
stack¹. pop ();
disp ();
stack¹. pop ();
disp ();
stack¹. pop ();
disp ();
}
}

PAGE NO. / / /
DATE / / /

PAGE NO. / / /
DATE / / /

element : "22
ap to buttas
<< endl;
" << endl;
#include <iostream>
#include <queue>
using namespace std;
int main() {
 queue<int> q;
 for (int i = 0; i <= 10; i++) {
 q.push(i * 10);
 }
 cout << endl;
 cout << "front element " << q.front() << endl;
 cout << endl;
 q.pop();
 cout << "After one pop, front = " << q.front()
 << endl;

cout << endl;
cout << "queue element (front to back)
:";
while (!q.empty()) {
 cout << q.front() << endl;
 q.pop();
}
}

Q
1111

Cout << " cube has greater value: " <<

Volume : << Volume
<< endl;

} else &

Cout << " Both have equal Volume: "
& Volume << endl;

}
}

int main () {

Berx box1 (100);

Cube cube2 (80);

ext, cube);

Final greater (box1, cube1);

return 0;

}

~~Output :~~

Berx has greater Volume : 100

(cube);

CDE

: <<

ed);

e3) #include <iostream>

#include <string>

using namespace std;

class Ilogin {

protected:

string name, password;

public:

Virtual void accept () {

cout << "Enter name: ";

(cin >> name);

cout << "Enter password: ";

(cin >> password);

}

Virtual void display () {

cout << "Name: " << name << endl;

cout << "Password << endl;

}

class EmailLogin : public Ilogin {

string email;

public:

Void accept () override {

cout << "Enter ^{member} Email ID: ";

(cin >> email);

Ilogin::accept();

}

Void display () override {

cout

cout

int

El

login();

a4 #include <iostream>
using namespace std;

class Complex {
private:
 int real;
 int img;

public:

Complex(int r=0, int i=0) {
 real = r;
 img = i;
}

friend

cout << real << " + " << img << "i"
<< endl;

friend Complex add(Complex, Complex);
};

~~Complex add(Complex c1, Complex c2)~~
~~Complex result;~~

result.real = c1.real + c2.real;

result.img = c1.img + c2.img;

return result;

}

```

cout << "Enter test percentage : ";
cin >> percentage;
cout << "Enter Spent Grade : ";
cin >> grade;
cout << "Enter Total Marks : ";
cin >> tot_marks;
    }
```

void display () {

```

CollegeStudent::display();
cout << "Test Percentage : " << percentage << endl;
cout << "Spent Grade : " << grade << endl;
cout << "Total marks : " << tot_marks << endl;
    }
```

int main () {

Result r;

```

cout << "-- Enter Student Details -- " << endl;
r.accept();
    }
```

```

cout << "\n-- Student Result -- " << endl;
r.display();
return 0;
    }
```

Ques
1/1

(Q5) #include <iostream>
using namespace std;

```

class student {
private:
    string name;
    int marks1, marks2, marks3;
public:
    student (String n, int m1, int m2, int m3)
        name = n;
        marks1 = m1;
        marks2 = m2;
        marks3 = m3;
}

```

{ friend void calculateAverage (Student s, Student s1)

$$\text{float avg} = (s.\text{marks1} + s1.\text{marks2} + s.\text{marks3}) / 3;$$

cout << "Student name : " << s.name << endl;

cout << "Average marks : " << avg << endl;

}

int main () {
Student s1 ("Ani", 85, 90, 95);
calculateAverage (s1);
return 0;

}

```

@2 #include <iostream>
using namespace std;

class Student {
    string name;
    float per;

public:
    Student (String n, float p) {
        name = n;
        per = p;
    }

    void display () {
        cout << "Name : " << name << endl;
        cout << "Percentage : " << per << endl;
    }
};

int main () {
    Student s ("Kanan", 44.3);
    s.display ();
    return 0;
}

→ Output :
Name : Kanan
Percentage : 44.3

```

Experiment 7: Class with (50)

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

class Area {

public

float calculate (float length, float
 breath);
 return length * breath;

}

float calculate (float side) {

return side * side;

}

int main () {

Area a;

cout << "Area of laboratory: " << a.calculate (10,6) ~~float~~
 cout << "Area of square: " << a.calculate (5) ~~float~~

College student :: accept () ;
cout << "Enter Test percentage : " ;
cin >> percentage ;
}

void display () {
College student :: display () ;
cout << "Test Percentage : " << percentage
<< " % " endl ;
}

Class exports : virtual public College Student
protected :
char grade
public :

void accept () {
cout << "Test Percentage : " << percentage
<< " % " endl ;
cout << "Enter exports grade : " ;
cin >> grade ;
}

void display () {
cout << "Spent Grade : " << grade << endl ;
}

Class Result : public test public Spent
& float +0 -mark ;
public :

void accept () {
College student :: accept () ;

case 'L' : cout << result = "(((n1(n2));
 break;
 } // main function

default:

~~cout << result~~
~~cout << "Result = " << ch~~

}

cout << "The value is " >> res

"return.

; IN << res

/* * maximum value */

: SN << res
 } (q0) return

else, +, -

(SN + IN) >> " = Number" >> res
 ; Max

else, -

(SN - IN) >> " = Number" >> res
 ; Min

(SN * IN) >> " = Number" >> res
 ; Mul

4)

Experiment 3: (1) `input.i2`

a) `#include <iostream>`

`using namespace std;`

`class Book {`

`public:`

`string title;`

`string author;`

`float price;`

`};`

`int main() {`

`Book b;`

`Book * ptr = &b;`

`Cout << "Enter book title : ";`
`getline (cin, title);`

~~`Cout << "Enter book title author name: price:" ;`~~
~~`getline (cin, author);`~~

~~`Cout << "Enter price : ";`~~

~~`Cin >> price;`~~

~~`cin.ignore();`~~

~~`}`~~

`void disp()`

`Cout << "In Book Details : \n ";`

`Cout << "Title : << title << endl ;`

#include <iostream>
using namespace std;

class College Student {

protected:

int student_id;

string college;

public:

void accept () {

cout << "Enter Student ID: ";

cin >> student_id;

cout << "Enter College Code: ";

cin >> college;

}

void display () {

//end;

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

cout << "College Code: " << college //end;

}

~~Class Test : virtual public College Student {~~

~~protected :~~

~~float percentage;~~

public :

void accept () {

cout << "Author: " << author << endl;
cout << "Price: " << price << endl;

}

int main () {

Book * ptr;
BOOK b;
ptr = &b;

cout << "Enter details for the book:\n";
ptr -> accept ();

cout << "\nDisplaying book details:\n";
ptr -> display ();

}

int main () {

BOOK * ptr;
Book b;
ptr = &b;

cout << "Enter details for the book:\n";
ptr -> accept ();

cout << "\nDisplaying book details:\n";
ptr -> display ();

return 0;

}

Experiment 2:

i) `#include <iostream>`
 by using namespace std;

class Account & class City {

public:
 int accNO, String name;
 int population;

void input () {

cout << "Enter city names :";
 cin >> name;
~~cout >~~
 cout << "Enter population :";
 cin >> population;

}

int main () {
~~city c[5];~~
 int maxIndex = 0;

for (int i = 0; i < 5; i++) {
 cout << "In City " << i + 1 << endl;
~~c[i].input();~~

if (c[i].population > c[maxIndex].population) {

Count << "In -- Membership Login Details
 --- " && endl;

Count && "Membership ID :" << memberID
 && tLogin :: display();
 }
 };

```
int main () {
    Login * login;
    Email login e;
    login->Membership login m;
    login -> accept ();
    login -> display ();
```

login = & m;
 login -> accept ();
 login -> display ();

return 0;

- Ch
 |||

```
int temp = n1.value  
n1.value = n2.value  
n2.value = temp;
```

{

```
int main () {
```

```
    Number a (10);
```

```
    Number b (20);
```

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

```
    a.show ();
```

```
    b.show ();
```

```
swap (Number (a, b));
```

```
Cout << "After swap : " << endl;
```

```
    a.show ();
```

```
    b.show ();
```

```
return 0;
```

{

Output :

Before Swap :

Value : 10;

Value : 20;

After Swap :

Value : 20

Value : 10

2)

#include <iostream>

using namespace std;

int main() {

int n;

cout << "Enter a number";

cin <> n;

if (n % 2 == 0) {

cout << "Even";

} else {

cout << "odd";

}

}: // main

}: // b72

; 0 marks

a2) #include <iostream>
using namespace std;

class sum {

public:

```
int total (int a[], int n) {
    int s = 0;
    for (int i=0; i<n; i++) {
        s += a[i];
    }
    return s;
}
```

float total (float a[], int n):

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

int main () {

sum s;

int marks [10] = {45, 56, 67, 78, 89,
76, 88, 42, 85};

float grades [5] = {4.2, 8.7, 9.5, 8.9}

cout << "Sum of 10 student marks : " <<

s. total (marks, 10);

cout << "Sum of 5 student marks : " <<

s. total (grades, 5);

return 0;

}

6)

#include <iostream>

int main () {

~~int rows, cols;~~

int rows = 5;

for (int i = 1; i <= rows; i++) {

(rows >> "read from std::cout" >> tree)

for (int j = 1; j <= rows - i; j++) {

cout << endl << count << "\n";

if (i + j == 1) cout

" " >> tree

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

count << "*";

tree >> tree

if (k < i) {

cout << " ";

}

cout << endl;

}

return 0;

}

```

Void swapNumber (A&X, B &Y) {
    int temp = X.num A
    X.num A = Y.num B
    Y.num = temp
}
    
```

```

int main () {
    A obj1 (10);
    B obj2 (20);
}
    
```

```

obj1.show (); obj2.show (); cout << endl;
swapNumbers (obj1, obj2);
obj1.show (); obj2.show ();
}
    
```

→ Output:

```

10 20
20 10
    
```

a12) ~~#include <iostream>~~
~~using namespace std;~~

```

class Result 2;

```

```

class Result 1 {

```

~~Heart waartis;~~
~~Public:~~

```

Void read () {

```

```

cout << "Enter marks for Result 1";
cin >> marks;
    
```

```

friend Void average (Result 1, Result 2);
}
    
```

```
int main () {
    marks m;
    m.accept();
    m.calculate();
}
```

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

```
class Person {
protected:
    string name;
    int age;
public:
```

void accept Per();

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

cin >> name;

cout << "Enter age: " << endl;

cin >> age;

}

```
class student : public Person {
    int roll;
    float per;
public:
```

void accept stud();

cout << "Enter roll number " << endl;

cin >> roll;

cout << "Enter Percentage " << endl;

cin >> per;

}

3)

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

int main () {
    float n1, n2;
    char op;

    cout << "Enter the first number";
    cin >> n1;

    cout << "Enter expression(*, -, +, /)";
    cin >> n2;
    switch (op) {
        case '+':
            cout << "Result = " << (n1 + n2);
            break;
        case '-':
            cout << "Result = " << (n1 - n2);
            break;
        case '*':
            cout << "Result = " << (n1 * n2);
            break;
    }
}
```

S[i].input(); // Enter Employee Details

Count("In list of HODs:\n");
for (int i = 0, i < 5; i++) {
 S[i].display If HOD;

}

} // End print
// End print
// End loop

Ques
29/7/2025

3 () main fun

Begin P

ds = 5 * 50

" : Hit and Run " >> true)

(Hit and Run)

" : Hit and Run " >> true)

(Hit and Run)

" : Hit and Run " >> true)

(Hit and Run)

" : Hit and Run " >> true)

(Hit and Run)

" : Hit and Run " >> true)

(Hit and Run)

Experiment 11:

(Q1)

```
#include <iostream>
```

```
#include <vector>
```

```
using namespace std;
```

```
int main () {
```

```
    vector<int> v = {1, 2, 3, 4, 5, 6, 7, 8, 9}
```

```
    cout << "Initial vector : " << endl;
```

```
    for (int i = 0; i < 10; i++) {
```

```
        cout << v[i] << " " << endl;
```

```
    }
```

```
    cout << "Multiply by 10 " << endl;
```

```
    for (int i = 0; i < 10; i++) {
```

```
        cout << v[i] << " " << endl;
```

```
        v[i] = v[i] * 10;
```

```
    }
```

```
    cout << "Multiplied New Vector : " << endl;
```

```
    for (int i = 0; i < 10; i++) {
```

```
    }
```

```
    cout << v[i] << " " << endl;
```

```
    cout << "New Vector : " << endl;
```

```
    for (int i = 0; i < 10; i++) {
```

```
        cout << v[i] << " " << endl;
```

```
    }
```

```
    cout << v[i] << " " << endl;
```

```
}
```

Q5) #include <iostream>
using namespace std;

class B;

class A {

int numA;

public:

void read();

cout << "Enter number for class A:";
cin >> numA;

friend void find Greatest (A, B);

}

int numB;

public:

void read();

cout << "Enter number for class B:";
cin >> numB;

}

friend void find Greatest (A, B);

}

void find Greatest (A x, B y) {

if (x.numA > y.numB)

cout << "Greatest: " << x.numA << endl;

else if (y.numB > x.numA)

cout << "Greatest: " << y.numB << endl;

else

cout << "Equal" << endl;

}

Q2 #include <iostream>
using namespace std;

class student {

private:

int roll_no;

float percentage;

public:

void accept () {

cout << "Enter Roll Number : ";

cin >> this -> roll_no;

cout << "Enter Percentage : ";

cin >> *Read this -> percentage;

void disp () {

cout << "In student Details :\n";

cout << "Roll Number : " << this->roll_no << endl;

cout << "Percentage : " << this->percentage << endl;

}

};

int main () {

student s;

s. accept ();

s. disp ();

}

return 0;

}

```

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

int main () {
    int num;
    cout << "Enter the first number" << endl;
    cin >> num;

    for (int i = 1; i <= num; i++) {
        for (int j = 1; j <= i; j++) {
            cout << j << " ";
        }
        cout << endl;
    }
}

```

; num >> num

; O output

(continued)

Experiment 11

PAGE NO.	/ /
DATE	/ /

2)

```
#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: ";
    for (auto it = v.begin(); it != v.end(); ++it)
        cout << *it;
    cout << endl;
    cout << "Multiplying each element by 10..." << endl;
    for (auto it = v.begin(); it != v.end(); ++it) {
        *it = (*it) * 10;
    }
    cout << "New vector after multiplication: ";
    for (auto it = v.begin(); it != v.end(); ++it)
        cout << *it << " ";
    cout << endl;
    return 0;
}
```

O/P

Initial vector:

1 2 3 4 5 6 7 8 9 10

Multiplying each element by 10...

New vector after multiplication:

10 20 30 40 50 60 70 80 90
100

Qn
|||~

```

int main ()
{
    A obj1(5);
    B obj2(7);

    cout << "Sum : " << sum(obj1,obj2) << endl;
}

return 0;
}

```

→ Output:
Sum : 12

a2) ~~# include <iostream>~~
~~using namespace std;~~

```

class Number {
private:
    int value;
public:
    Number (int val) {
        value = val;
    }
    void show () {
        cout << "Value : " << value << endl;
    }
}

```

friend void swapNumbers
(Number & n1, Number & n2);
};

void swapNumbers (Numbers &n1, Numbers &n2){}

a3) #include <iostream>
using namespace std;

```
class Cube;  
class Box {  
private:  
    int Volume;  
public:
```

```
    Box (int v) {
```

```
        Volume = v;
```

```
}
```

```
    friend void findGreater (Box, Cube);  
};
```

```
class Cube {
```

```
private:
```

```
    int Volume; // Volume of cube
```

```
public:
```

~~```
 Cube (int v) {
```~~~~```
        Volume = v;
```~~~~```
}
```~~~~```
friend void findGreater (Box, Cube);
```~~

```
void findGreater (Box b, Cube c) {
```

```
    if (b.Volume > c.Volume) {
```

```
        cout << "Box has greater volume: " <<
```

```
    } else if (c.Volume > b.Volume) {
```

```
        b.Volume << endl;
```

square function

n

>

T

str) {

km)

km)

float

id:

rate

m) {

cnum)

) < 2 m

0)

0)

0)

0)

0)

0)

0)

0)

0)

0)

0)

0)

0)

WAP to build a simple calculate using template:

```
#include <iostream>
#include <math.h>
using namespace std;
```

```
template < class T1, class T2 > class Calc {
```

public:

 T1 n;

 T2 y;

```
    Calc (T1 n, T2 y) {
```

 n = n;

 y = y;

}

```
    void sum () {
```

```
        cout << "Sum : " << n + y << endl;
```

~~void sum () {~~

~~cout << "Sum : " << endl;~~

```
    void diff () {
```

```
        cout << "Difference : " << n - y << endl;
```

~~void pro () {~~

~~cout << "Product : " << n * y << endl;~~

Q2) #include <iostream>
 using namespace std;

class department {
 protected:
 string name;
 int val;

};

class marks : protected department,
 protected student {
 int m1, m2, percentage;

public :

void accept () {
 cout << "Enter department : " << endl;
 cin >> name;
 cout << "Enter name : " << endl;
 cin >> name;

Cout << "Enter mark : " << endl;
 cin >> m1;

Cout << "Enter marks 2 : " << endl;
 cin >> m2;

}

void calculate () {

int per = (m1 + m2) / 2;

Cout << "Department : " << name << endl;

Cout << "Name : " << name << endl;

Cout << "Percentage : " << percentage

}

} ;

cout << "Name : " & name && endl;
 cout << "Percentage : " & percentage && endl;

```
} ;  

int main () {  

    marks m;  

    m.accept ();  

    m.calculate ();  

    return 0;  

}
```

→ Output:

Enter department:

Human

Enter mark 1:

83

Enter mark 2:

91

Department : CSE

Name : Human

Percentage : 89

(iii) #include <iostream>
using namespace std;

class staff {

{ string name;

string post;

public:

void input();

cout << "Enter Name:";

cout << "Enter post:";

cin >> name;

cin >> post;

};

}

void display();

if (post == "HOD" || post == "head")

if (post == "HOD" || post == "head")

cout << "HOD Name " + name + " ";

<< name << endl;

}

}

};

int main()

staff s[5];

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

cout << "Enter details of staff " <<

i + 1 << endl;

Q2) WAP to main with a square function using template specialization

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

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

```
template <>
string square <string> (string str) {
    return str + str;
```

```
int main () {
    int num = 5;
    double dec_num = 2.5;
    string str = "Apple";
```

```
cout << "Square of integer: " << square (num) <<
cout << "Square of double: " << square (dec_num) <<
cout << "Square of string: " << square (str) <<
```

```
return 0;
```

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

`class Account {`

`public:`
`int accNO;`
`float balance;`

`void input () {`

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

`cin >> accNO;`

`cout << "Enter balance : ";`

`cin >> balance;`

`}`

`void give Interest () {`

~~`if (balance >= 5000) {`~~

~~`balance += balance * 0.10;`~~

~~`}`~~

~~`}`~~

`void display () {`

`cout << "Acc No : " << accNO << "`

`Balance : "`

`<< balance << endl;`

`}`

`};`

Plenty. getPrice() const
Return price;

}, Constructor

int main()

{

Book1, Book2, Book3;

cout << "Enter details for Book1" \n";

book1.input();

cout << "Enter details for Book2" \n";

book2.input();

cout << "Enter details for Book3" \n";

if (book1.getPrice() > book2.getPrice())
book1.display();

else if (book2.getPrice() > book1.getPrice())

book2.display();

else

{ cout << "Both books have same price." \n";

book1.display();

book2.display();

stream

& include <iostream.h> :
using namespace std;

class student {
int count;

public :
student (int c = 0) {
count = c;

}
void display {
cout << "Student Count: " << count;

void operator ++ () {
++ count;

}

void operator ++ (int) {
count ++;

}

void display {

cout << "Student Count: " << count;

38
39

int main () {

student s1 (50);

cout << "Before increment: " << s1.
- s1. display ();
++s1;

cout << "After pre-increment: " << s1.
s1. display ();

s1 ++;

cout << "After post increment: " << s1.
s1. display ();

}
return 0;

OK

Ex

(1)

#i
#i
u

11
cent.

cent

10

Credit << "EmpID": << emp_id << endl;
Credit << "Subject taught": << subject
<< endl;

}
}

int main ()
{ student s;
staff t;

s. accept Rev();

s. accept stud();

t. accept Rev();

t. accept staff();

s. display stud();

t. display staff();

return 0;

}

Final varcl and its Balance (Bank Account, Audit)

}

Varcl and its Balance (Bank Account, Audit)
Count << "Balance : " << acc.balance << endl;

}

int main () {

Bank Account my Acc (50000);
Audit auditor;

auditor. Start Audit (my Acc);
Return 0;

}

→ Output:

Balance : rs. 50000

(a) #include <iostream>

using namespace std;

Class Number {

int num;

public :

Number (int n) { num = n; }

Varcl swap (Number &n) {

int t = num;

num = no num;

no num = t;

}

Experiment - 4:

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

class B;

class A {

private

int n;

public :

A(int val) {

n = val;

} friend int sum(A, B);

}

class B {

private :

int y;

public :

B(int val) {

y = val;

}

friend int sum(A, B);

}

int sum (A a, B b) {

return a.n + b.y;

}

Experiment 9

a1) #include <iostream>
#include <str流stream>
using namespace namespace std;

```
int main () {  
    ifstream inFile ("First.txt");  
    ofstream outFile ("Second.txt");
```

```
    if (!inFile) {  
        cout << "Error opening First.txt";  
        return 1;  
    }  
  
    char ch;  
    while (inFile.get (ch)) {  
        outFile.put (ch);  
    }
```

```
cout << "File Copied successfully"  
inFile.close ();  
outFile.close ();  
return 0;
```

Waiel show() & cout << num << " ";
};

```
int main () {  
    Number a(10), b(20);  
    a.show(); b.show(); cout << endl;  
    a.snap(b);  
    a.show(); b.show();
```

-> Output:

10 20
20 10

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

class

int

→ Output :

Student Name : Raman
Average Marks : 90

(a)

#include <iostream>
using namespace std;

Class Beta;

Class Gamma;

Class Alpha {

private:

int a;

public:

Alpha (int val) {

a = val;

}

friend void total (Alpha,
Beta, Gamma);

}

Class Beta {

private:

int b;

public:

Beta (int val) {

b = val;

}

friend void total (Alpha, Beta,
Gamma);

(4) ~~#include <iostream>
 #include <fstream>
 #include <iomanip>
 using namespace std;~~

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

```
int main () {
    ifstream file ("First.txt");
    if (!file) {
        cout << "Error" << endl;
        return 1;
    }
```

```
String word, target;
int count = 0;
```

```
Count << "Enter word to count:" << endl;
cin >> word;
while (file >> word) {
    if (word == target)
        count++;
}
```

~~Count << "Occurrence of " << target << "l";
 << Count << endl;~~
 file.close();

Q
1111

Void display stud()

{ void cle

cout << "Name : " << name << endl;

cout << "Age : " << age << endl;

cout << "Roll No" << roll << endl;

cout << "Percentage : " << per << endl

cin >> per;

}

Void display stud()

cout << "Name : " << name << endl;

cout << "Age : " << age << endl;

cout << "Roll no : " << roll << endl;

cout << "Percentage : " << per << endl

}

};

class staff : public person {

int emp_id;

string subject;

public :

Void acceptstaff()

cout << "Enter employee ID : " << endl;

cin >> emp_id;

cout << "Enter subject : " << endl;

cin >> subject;

}

Void display staff()

cout << "Name : " << name << endl;

cout << "Age : " << age << endl;

Output: ~~books at 9 AM (E)~~

Enter details for Book1

Enter book ID: 78

Enter book name: HIMYM

Enter book price: 249

Enter details for Book2

Enter book ID: 40

Enter book name: unsent unsold
: sending unwritten

Enter book price: 399

Book with higher price:

Book ID: 40

Book Name: unsent unsold

Book Price: 399

Book1
";

2:1h";

price:
";

(Price())

(Price())

i.e. hi;

```
int main () {
```

```
    A obj1;
```

```
    B obj2;
```

```
    obj1.read();
```

```
    obj2.read();
```

```
    friend greatest (obj1, obj2);
```

```
    return 0;
```

```
}
```

A:";"

→ Output:

Enter number for class A: 17

Enter number for class B: 23

Greater: 23

Qn

|||

3) WAP to declare a class student having data member as name, Roll no. Accept & display data for one student:

MR M I H : man hard work

#include <iostream>

using namespace std;

class Time

{

int hour, min, sec;

public:

void time()

{

cout << "Enter";

cin >> h >> m >> s;

hour = h;

min = m;

sec = s;

void display()

cout << "Hour"

int sum = (h * 3600) + (m * 60);

cout << "Total time = " << sum;

}

int main()

{

Time t;

t.time();

t.display();

}

④

1)

#include <iostream> *Wii* (S)int main () { *pair* }

double n1, n2, sum;

std::cout << "Enter the first

number: " >> n1;

std::cin >> n1;

std::cout << "Enter the second
number: ";

std::cout << n2;

>> n2;

sum = n1 + n2;

std::cout << "The sum is : "

<< sum << std::endl;

return 0;

}