Date
Page

Experiment - 1.

Q1] WAP to decide a class student having data members as name, roll no. Accept & display data for one student.

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

 string name ;
 int roll_no ;
 public ;
 void accept () ;
 void display () ;
} ;

void student :: accept (){
 cout << "Enter name and roll no :" ;
 cin >> name >> roll_no ;
}

 void student :: display ()
{
 cout <<"\n student name :" <<name ;
 cout <<"\n Student roll no : "<< roll_no ;
}
```

```
int main (){

    student SI ;
    SI . accept () ;
    SI . display () ;

    return 0 ;
}
```

Output :-

Enter name and roll no : abc 22

student name :- abc
student roll no :- 22.

Q2] 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 {
public :

int id ;
string name ;
float price ;
```

```
void input () {

cout << "Enter id, name, price: ";
cin >> id >> ws ;
getline (cin, name);
 cin >> price ;
}

void display () {

cout << "ID: " << id << name << ", price: "
    << price << endl ;

 }

};

int main () {

Book b1 , b2 ;
cout << "Book 1: \n ; b1 . input () ;
cout << "Book 2 : \n ; b2 . input () ;

 cout << "\n Book with higher price : \n" ;
if (b1 . price > b2 . price)
    b1. display () ;
else if (b2. price > b1. price)
    b2. display () ;
else {

  cout << "Both have the same prices: \n";
```

```
b1.display();
b2.display();

}

return 0;
```

Output :-

Book 1 :

Enter id, name, price : 123456 RD 459

Book 2 :

Enter id, name, price : 123456 PD 256

Book with higher price :

ID : 123456
Name : RD
Price : 459.

Q3] WAP to declare a class time having data members as H, m & s. Accept data for one object & display Total time in seconds.

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

int H, M, S;
public :

void accept ();
void displayTotal ();

};

void Time :: accept () {
cout << "Enter time in hours, minutes and seconds: ";

cin >> H >> M >> S;

}

void Time :: displayTotal () {
int totalSeconds = H * 3600 + M * 60 + S;
cout << "Total time in seconds: << totalSeconds << endl;

}

int main () {

Time t1;
t1.accept ();
t1.displayTotal ();
return0;
}

Output :-

Enter time in hours, minutes and seconds : 3
22
10

Total time in seconds : 12130

5/8

Experiment - 2.

Q1]

→
```
#include <iostream>
using namespace std;
class city
{
public :
  string name ;
  int population ;

void accept () {

  cout << "Enter the name of city";
  cin >> name ;

  cout << "Enter the population";
  cin >> population ;

  }
};

int main () {
    city cities [5];

  for (int i=0 ; i<5 ; i++)
    cout << " Enter the details";
      cities [i] . accept () ;
        maxIndex = i ;
}  }
```

```
cout << "\n City with the highest population" ;
    << cities[maxIndex].name << " cities[maxIndex].
    population << ")" << endl ;

return 0;

}
```

Output :-

Enter City name :- Pune
Enter population :- 67

Q2]

→

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

int account_no ;
float balance ;
Public :

void accept (){

cout << "Enter the account no" ;
cin >> Account no ;
cout <<"Enter the account balance";
cin >> balance ;
}

bool ApplyInterest () {

    if (balance >= 5000) {
        balance += balance * 0.10 ;
        return true ;
    }
        return false ;
}

void display () {
  cout << " Account no" << account_no <<
  " Balance" << balance << endl ;
```

```
}

};

int main () {

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

cout<<"\n Enter the details"<<i+1<<":\n";
accounts [i]. accept ();

}

return 0;

}
```

Q3]

→

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

```
void display ()
{

cout << "\n Staff Name :" << Name;
cout << "\n Staff Post :" << Post;
}

}
int main ()
{
  staff s[5];
  int i;

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

   s[i] . accept (); }

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

    if (s[i] . post == "HOD")
  {

  s[i] . display ();
  }

  }

  return 0;
  }
```

Output :-

Enter the name of the staff : Riya.
Enter post : HOD
Enter the name of staff : Siya
Enter post : Accountant
Enter the name of the staff : Diya
Enter the post :- HOD
Enter the name of the staff : Piya
Enter the post :- Accountant
Enter the name of the staff : Ina
Enter the post of the staff : Accountant.

The name of the staff who are HOD :-

Riya
Diya.

5/8

Experiment - 3

Q1]

```
#include <iostream>
using namespace std;
class book {
        private :
          string title ;
          string author ;
          float price ;

public :
    void accept (){

    cout << "Enter book title : " ;
     cin >> title ;

    cout << " Enter author name : " ;
    cin >> author ;

    cout << "Enter the price of book : " ;
    cin >> price ;
}

void display (){

    cout << " \n Book details : \n" << endl ;
    cout << "title : " << ~~auth~~ title << endl ;
    cout << "author : " << author << endl ;
    cout << "price : " << price << endl ;
}
};
```

```
int main () {

    book * ptr ;
    book b1 ;
    ptr = & b1 ;

    ptr -> accept () ;
    ptr -> display () ;

    return 0;

}.
```

Output :-

```
Enter book title : abc
Enter author name : rengoku
Enter the price of the book : 3599

Book details :

title : abc
author : rengoku
price : 3599.
```

Q2]

```
#include <iostream>
using namespace std;
class student {
        private :
        int roll_no ;
        float percentage ;

       public :
       void accept () {

         cout << " Enter roll no : " ;
         cin >> this -> roll_no ;
         cout << " Enter percentage " ;
         cin >> this -> percentage ;
}

  void display () {

     cout << " Student Details : " ;
     cout << " roll_no : " << This -> rollno << endl ;
     cout << " percentage : " << This -> percentage << endl
}

};

int main () {

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

Output :-

Enter roll no : 67
Enter percentage : 100

Student details :

roll no : 67
percentage : 100

Q3]

```
#include <iostream>
using namespace std;
class student
{
 public :
 string name ;
 int roll;
 void accept () {
  cout << " Enter roll no : ";
  cin >> roll ;
  cout << " Enter name : ";
  cin >> name ;
 }
  void display ()
 {
```

```
cout <<" Roll no :" << roll;
cout <<" Name" << name ;
}

class marks
{

  int m1 , m2 ;
  public:
  void accept ()
{

 cout << " Enter marks of two subjects :";
 cin>>m1>>m2 ;
}

  void display ()
  {

   float pr = ((m1 +m2) / 100 .0) * 100 ;
   cout << "percentage =" << pr ;
  }
};

  int main ()
{

  student s1 ;
  student :: marks m;
   s1. accept () ;
   s1 . display ();
```

```
m. accept ();
m. display ();

return 0;
}
```

Output :-

Enter roll no: 68

Enter Name : Advait

Enter marks : 12
20

percentage = 32.

Q
5/8/25

Experiment no. - 4

1] WAP to swap 2 numbers from same class using object as functions argument. Write swap function as member function.

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

class Numbers {
        int value ;

Public :
    // functions to set value
    void set value (int v) {
        value = v ;
}

    // Function to display the value
    void display () {
        cout <<"value : " << value << endl ;

    void swap (Number & obj) {
        int temp = value ;
        value = obj . value
        obj . value = temp ;
    }
};
```

```
int main (){
    Number num1 , num2 ;

 num1 . set value (10) ;
 num2 . set value (20) ;

 cout <<" Before swapping :" << endl ;
cout <<"num1 : "; num1 . display () ;
cout <<" num2 : "; num2 . display () ;

 num1. swap (num 2) ;

cout <<" In After swap :" << endl ;
cout <<" num1 :" ; num1 . display () ;

 return 0 ;

}
```

Output :-

```
Before swap :
num1 . value : 10
num2 . value : 20

After swap :
num1 . value : 20
num2 . value : 10 .
```

Q2]

→ #include <iostream>
using namespace std;

class number {
int value ;

Public :

void set value (int v) {
value = v;

}

void display () {
cout << " Value : " << value << endl ;

}

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

};

void swap (Number &a, Number&b);
int temp = a. value ;
a. value = b. value ;
b. value = temp ;

}

int main () {
Number num1 , num2 ;

num1 . set values (6) ;
num2 . set values (15) ;

```
cout <<"Before swap :" << endl ;
cout <<" num1 : ; n1. display ();
cout << " num2 :" ; num2. display ();

swap (num1 , num2) ;

cout <<" After swap : " << endl ;
cout <<" num1: " ; num1 . display ();
cout <<" num2 : " ; num2 . display ();

return 0;

}
```

Output :-

Before swap :

num1 : value : 5
num 2 : value : 15

After swap :

num1 : value : 15
num 2 : value : 5

Q3]

→

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

class class B;

class class A; {

int value A;

Public :
        void set value (int v) {
            value A = v;
}
    void display () {
        cout <<" value in class A:" << value A
                                << endl;
}
    void display () {
        cout << " value in class A : " << value
                                A <<endl;
}
  Friend void swap (class A & class B);
};

    class class B {
        int value B;

Public :
        void set value (int v) {
```

```
value B = v ;
}
void display () {
cout << " value in class B : " << value B << endl;
}
friend void swap (class A & class B);
};

void swap (class A &a, class &b){

int temp = a . value A;
a . value A = b . value B;
b . value B = temp B;
}

int main () {
class A obj A;
class B obj B;

cout << " Before swap : " << endl;
obj A . display ();
obj B . display ();

return 0;
}
```

Output :-

Before swap :
Value in class A : 10
Value in class B : 20.

After swap :
Value in class A : 20
Value in class B : 10.

Q4]

→ #include ~~<stdio.~~ <iostream
using namespace std;
class result 2 ;

class result {
float marks 1;

public :
void read marks () {
cout << "Enter marks for Result 1 :";
cin >> marks 1 ;

}
void display () {
cout << "Marks in Result 1 :" << Marks 1 << endl

}
friend float compute averge (result 2)

} ;
class Result 2 {
float marks 2;

```
public :
    void read marks (){
    cout << "Enter marks :";
    cin >> marks 2 ;

}
    void display (){
    cout << " Marks in Result 2 :" << marks 2 <<
                                        endl;

}
    friend float compute average (Result)
};

    int main (){
    Result 1 obj 1;
    Result 2 obj 2;

obj 1 . read marks ();
obj 2 . read marks ();

obj 1 . display ();
obj 2 . display ();

float avg = compute Average (obj 1, obj 2);
cout << " Avg. of marks :" << avg << endl;

    return 0;

}.
```

OUTPUT :-

Enter ~~name~~ marks ~~fro~~ for result 1 : 85
Enter marks for Result 2 : 75
marks in Result 1 : 85
marks in Result 2 : 75.

Avg marks : 80.

Q5]
→

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

  Class class A {
        int num A;

Public :
    void set data (int a) {
        num A = a;
}   Friend void Find Greatest (classA, classB);

};  class class B {
        int num B;
public :
    void set data (int b) {
        num B = b;
}   Friend void Find Greatest (classB, classA).
};
```

```
void Find Greatest (class A a, class B b){
  if (a.numA>b.numB) {
  cout << " Greatest number is:" << a.numA <<
                                        endl;
} else if (b.numB > a.numA){
  cout << " Greatest number is: " << b.numb
                                    << endl;
} else {

      cout << " Both no. are equal: " << a.numA
                                    << endl;
  }
}

int main (){
      class A obj B;
      Class B obj B;

int val1, val2;
cout << " Enter no. of class A: ";
cin >> val 1;
cout << " Enter no. of class B: ";
cin >> val 2;

obj A . set data (val1)
obj B . set data (val2);

Find Greatest (obj A , obj B);
    return 0;
}
```

Output :-

Enter no. ~~of~~ for class A : 30
Enter no. For class B: 45

Greatest no. is 45.

6]

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

private:
        int num A;

public:
        A(int A): numA (a) { }

friend int sum (const A & objA, const B &
                                  obj B);
};
    class B {
    private:
            int num B;

Public:
        B (int B): numB (b) {

    friend int sum (const A & objA,
                    const B & objB).
};
```

```
int sum (const BA & objA , const B & objB)
    return objA . numA + objB . numB ;

}  int main () {
        A a (55) ;
        B b (20) ;

    cout << "Sum :" << sum (a, b) << endl) ;

    return 0 ;

}
```

Output :-

Sum :- 75 .

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

private :
        int value ;

public :
        Number (int v) : value (v) {

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

```
friend void swapNumbers(Number &n1,
                        Number &n2).
};
    void swapNumbers.
      int temp= n1.value;
      n1.value = n2.value;
      n2.value = temp;

}
  int main (){
  Number n1 (10);
  number n2 (20);

cout <<"Before swapping:" <<endl;
cout <<" num 1 = ";

num1.display ();
cout <<" num2" ;
num2.display();

swapNumber (num1, num2);

cout<<" After swapping :" <<endl;
cout <<"num1=";
num1.display();
cout <<"num2=";
num2.display();

  return 0;
}.
```

8]

→ #include <iostream>
using namespace std;
class cube :

class box {
private :
double volume ;

public :
Box (double length, double width, double height) {

volume = length * width * height ;
}
friend void find Greater (Box, cube);
};
class cube {
private :
double volume ;

public ;
cube (double side) {
volume = side * side * side ;
}
friend void find Greater (Box, cube);

};
void find Greater (Box b, cube c){
if (b.volume > c.volume)

```
else if (c.volume > b.volume)
    cout << " Cube has greater volume :";
        c.volume << endl ;

else
        cout << " Both have equal volume :";
            << b.volume << endl;

}
    int main () {
      Box box (3, 4, 5);
      Cube cube (4);

    find Greater (box, cube);

    return 0 ;
```

* Output :-

cube volume is greater than 64.

9]

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

private :
        double real ;
        double image.

Public :
```

```
complex (double r=0 ; double i=0 ): real
  imag(i) {}.

Friend complex add (const complex & const
                        complex (2)) ;

    }
};
    complex add (const example & a, const
                        complex & (a 2) ;

  return complex (a.real + (2 · real) (1 · image,
                        (2. image).
}
  int main () {
        complex c1 (3,4)
        complex c2 (5, -2) ;

  complex c3 = add (c1 ,c2) ;

  cout << " c1 = " ; c1 · display ();
  cout << " c2 = " ; c2 · display ();
  cout << " sum = " ; c3 . display ();

  return 0; }.
```

output.

c1 = 3+4i

c2 = 5-2i

Experiment 5

1] i] Default Constructor :-

→
```
#include <iostream>
using namespace std;
class sum
{
  int n, s;
  public:
  sum()
  {
    n=10;
  }
  void calculator ()
  {
    for (int i=1; i<=n; i++)
    {
      s=s+i;
    }
  }
  void display ()
  { cout <<" The sum from 1 to n is:"
    <<s;
  }
};
```

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

→ Output :-

The sum of from 1 to n is : 55.

ii]

```
#include <iostream>
using namespace std;
class sum
{
    int n, s = 0;
    public :
    sum (int num)
{
    n = num;
}
    void calculate ()
{
    for (int i = 1 ; i <= n ; i++)
{
    s = s+i;
}
}
```

```
cout << " The sum from 1 to n is:"<<s;
    }
};
    int main()
{
    sum s1(10);
    s1.calculate();
    s1.display();
      return 0;
}
```

→ Output :-

The sum from 1 to n is : 55.

iii] ~~#~~ # include <iostream>

```
using namespace std;
class sum
{
    int n, s=0;
    public:
    sum (int num)
{
      n= num;
}
```

```
sum (sum & s1)
{
    n = s1.n;
    for (int i=1 ; i<=n ; i++)
{
    s = s+i;
}
cout << " The sum from 1 to n is:"<<s

    }
};
    int main ()
{
    sum s1 (10);
    sum s2 (s1);
    return 0;

}
```

Output :-

The sum from 1 to n is : 55.

2] i] Default :-

```
# include <iostream>
using namespace std;
class student
{
    string name ;
    float per ;
    Public :
    Student ()
{
    name = " Adwait" ;
    per = 98 ;
}
    void display ()
{
cout << " The name of the student
        is " << name << endl;

cout << " The percentage of the student
        is :" << per << "%" ;

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

→ Output :-

The name of the student is : Adwait
percentage : 98.1.

ii] Parameterized :-

```
#include <iostream>
using namespace std;
class student
{
  string name ;
  float per ;
  public :
  student (string n , float p).
{
    name = n ;
    per = p ;
} void display ()
{
cout << " The name of the student is:
        << name << endl.
cout << " The percentage : << per << " % "
}
};
int main ().
{
```

```
Student s1 ("~~Disha~~ Advait", 98);
  s1.display();
    return 0;
}
```

Output :-

The name of the ~~function~~ student : Advait.
The ~~name~~ ~~of~~ percentage : 98.01.

iii] Copy :-

```
# include <iostream>
 using namespace std;
   class student.
{
    string name;
    float per;
     public :
     student ( string n , float p)
{
      name = n;
      per = p;
}
     student (student & s1)
{
     name = s1.name;
     per = s1.per
}
     void display ().
{
```

```
cout << " The name of the student is:
        << per << " % " << endl;

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

Output :-

The name of the student : Adwait
percentage : 98 %.

3]
```
#include <iostream>
using namespace std;
class college
{
    string name, course;
    int roll no;
    public :

college (string n, int r, string c = computer
                                engineering");
```

```
{
    name = n;
    course = c;
    roll = r;

}
    void display ()

{

    cout << " The name of the student: ";
    cout << "The course is : " << course << endl;
cout << "The roll no. : " << roll << endl;

    }

};
    int main ().

{
    college c1 ("Adwait", 67);
    c1.display ();
    college c2 ("Disha");
    c2.display ();
    return 0;

}
```

→ Output :-

4]

```
#include <iostream>
using namespace std;
class rectangle.
{
    int l, b;
    public:
    rectangle ()
{
    l = 2;
    b = 5;
}
    rectangle ( int x)
{
    l = x;
    b = y;
}
    rectangle (int x, int y)
{
    l = x;
    b = y;
}
    rectangle (~~int x, int y~~) (rectangle &r3)
{
    l = r3.l;
    b = r3.b;
}
```

```
void calculate ()
{
    cout<<"The area is: <<(l*b)<<endl;
  }
};
    int main ()
{
    rectangle r1;
     r1. calculate() ;
    rectangle r2;
     r2. calculate();
     rectangle r3;
      r3. calculate ()
      rectangle r4;
       r4. calculate ();
        return 0;
  }.
```

Output :-

```
The area is: 10
The area is 9
The area is: 36
The area is: 36.
```

Experiment - 6.

1] Single inheritance:-

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

};
    class student : protected person.
{   int roll no.;
    public:
    void accept ()
{

 cout<<" Enter the name :");
 cin>> name;
cout<<" Enter the age:;
cin>> age;
cout<<" Enter the roll no.:";
cin>> roll no.;

}
```

```
void display ().
{
  cout << "The name of the student : <<
                                name <<endl;

cout << "The age of the student : <<
                                age <<endl;

  cout << "The roll no. : << roll no << endl;

    }
};

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

→ Output :-

```
Enter department : Adwait
Enter marks 1 : 88
Enter marks 2 : 91
Department : CSE
  Name : Adwait
  Percentage : 89.
```

Q2] WAP to implement multiple inheritance Assume suitable data.

→
```
# include <stdio.h>
using namespace std;
class department {
protected :
        string dname ;
};

class student {
protected :
    string sname ;
    int roll ;
};

class marks : protected department , protected
                                student {

    int m1 , m2 , percentage ;
        public :

    void accept () {
    cout << " Enter department :" << endl ;
    cin >> dname .
    cout << " Enter name :" << endl ;
    cin >> sname ;
    cout << " Enter marks 1 :" << endl ;
    cin >> m1 ;
    cout << " Enter marks 2 : " << endl ;
    cin >> m2 ;
```

```
}
    void calculate (){
    int per = (m1 & m2) /2;
    cout << " Department : " << dname << endl;
    cout << " Name : << sname <<endl;
    cout << " percentage : " << per << endl;
    }
};

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

Q3] WAP to implement heirarchical inheritance

→
```
# 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 no. " << endl;
    cin >> roll;
    cout << " Enter percentage : " << endl;
    cin >> per;
}

void display stud () {
    cout << " Name : " << name << endl;
    cout << " Age : " << age << endl;
    cout << " Roll no : " << roll << endl;
    cout << " Percentage : " << percentage << endl
}
};

class staff : public person {
    int emp_id;
    string subject;
    public :
void accept staff () {
    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;
cout << " EmpID : << emp_id << endl;
cout << " subject taught : " << subject taught
                                << endl;
}
};
int main () {
    student s;
    staff t;

    s. accept Per ();
    s. accept stud ();
    t. accept Per ();
    t. accept staff ();

    s. display stud ();
    t . display staff ();

    return 0;
}
```

Q4] WAP to implement hybrid inheritance.

→

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

class college {
    protected :
    string name ;
```

```
};
    class Employee : protected college {
      protected:
        string cname;
};

    class employee : ~~emp~~ protected college{
      protected :
        string emp_name;
        int id;
};

class staff : public Employee {
string name;
int deptId;
public :
void accept Emp () {
  cout <<"Enter college name" : << endl;
  cin>> cname ;
  cout <<" Enter Emp name :" << endl;
  cin>>emp_name;
  cout<<" Enter ID :" << end;
  cin>> id;
  cout <<" Enter staff name :" << endl;
  cin>> name;
cout<<" Enter dept id : " << endl;
cin>> dept id;

}
```

```
void display Emp () {
    cout << "college :" << cname << endl;
    cout << "Emp name :" << emp_name << endl;
    cout << "ID :" << id << endl;
    cout << "staff name :" << name << endl;
    cout << "Department ID:" << dept Id << endl;
    }
};

class student : protected college {
string stu_name;
int roll;
Public:
void accept std () {
cout << " Enter college name :" << endl;
cin >> cname;
cout << " Enter name :" << endl;
cin >> stu_name;
cout << " Enter roll number :" << endl;
cin >> roll;
}

void display stud () {
cout << " college :" << cname << endl;
cout << "student name : << stu_name << endl;
cout << " Roll no :" << roll << endl;
}
int main () {
    staff staff 1;
    staff 1 . accept emp();
    staff 1 . display emp();
```

```
Student stud1;
stud1.accept stud();
stud1.display stud();

return 0;
}
```

Q5] WAP to demonstrate virtual base class

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

class CollegeStudent {
protected:
    int student_id;
    string code;
public:
void accept Emp ()
{

cout << " Enter college name:
```

Experiment - 7

1]

→ #include <iostream>
using namespace std;

class Area {
Public :
float calculate (float length , float breadth){

return length * breadth ;
}

float calculate (float side) {
return side * side ;
}

};

int main (){
Area a ;
cout << " Area of Laboratory :" << a. cal << endl;
cout << " Area of square :" << a. calculate << endl;

Q2]

→ #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=~~ s+ = a[i] ;
return s ;
}
};
int main () {
Sum S ;
int marks [10] = {45, 56 , 67 , 78 , 89, 90, 76 , 88 , 92 , 85}
float grades [5] = {9.2, 8.7, 9.5 , 8.9 , 9.0}
cout << " Sum of 10 student marks :" <<
s.total (marks , 10) << endl;
cout<<" Sum of 5 students grades points :"<<
s.total (grades , 5] << endl;

return 0;
}

03]

```
#include <iostream>
using namespace std;
class teacher {
    int experience;
Public :
  Teacher (int e) {
    experience = e;
  }
  void display () {
  cout<<" Experience :"<< experience << year<<
                                    endl;
}

void operator - () {
    experience = - experience;
  }
};
    int main () {
    Teacher t1[10];
    t1. display ();
    -t1;
  cout <<" After negation :";
    t1 . display ();
    return 0;
}
```

Q4]

```
#include <iostream>
using namespace std;
class student {
        int count;
public:
        student (int c=0) {
            count = c;
        }

void operation ++ () {
        ++ count ;
        }

void operation ++ (int) {
    count ++ ;
}

  void display {
        cout << " student count :" <<count << endl;
    }
};

  int main () {
        student s1 (50);
    count << " Before increment :" << endl;
    s1 . display ();

    ++ s1 ;
    count << " After pre-increment :" << endl;
        s1. display ();
        return 0;
  }
```

Experiment - 8

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

class Combine {
        string str ;
Public :
        Combine (string s=" ") {
        str=s ;
  }

Combine operator ((Combine & obj ){
  return combine (str+ obj . str);
}
  void display ()
  cout<<" str << endl ;
  }

};
  int main ()
   combines s1 ("xyz"), s2 ("pqr"), s3;
  s3 = s1 + s2 ;
  cout <<" con catenated string :";
  s3 . display ();

}
```

Q2]

→

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

class I login {
  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 : " << Password << endl;
    }
};
  class Email Login : Public I login {
    string email ;

Public :
        void accept () override {
        cout<<" Email Id :" ;
        cin >> email ;
        I Login :: accept () ;
}
```

```
void display () override {
  cout << "\n --- Email Login Details ---- " <<
                                    endl ;
cout << " Email Id : " << email << endl;
 I Login :: display ();
    }
};

class Membership login : public I login {
      string member Id;
Public :
    void accept () override {
  cout << " Enter Membership : " ;
  cin >> member ID;
  I Login :: accept ();
}
  void display () override {
  cout << " \n ---- Membership login details : " <<
                                    endl;
cout << " Membership ID : " << member ID << endl;
I Login :: display ();
    }
};
    int main () {
    I Login * login ;

    Email Login;
    membership Login m;

    Login = & e ;
    Login → accept ();
    Login → display ();
```

```
Login = & m;
Login -> accept ();
Login -> display ();

return 0;

}
```

Experiment - 9

01]

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

int main () {
ifstream infile ("first.txt");
ofstream outfile ("second.txt");

if (!infile){
cout << " Error operating first txt" << endl;
return 1;
}
char ch;
while (infile.get (ch)){
outfile.put (ch);
}

cout << " file copied successfully" << endl;

infile.close ();
outfile.close ();
return 0;

}
```

Q2]

→

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

int main () {
 ifstream file ("first.txt");
if (! file) {
  cout <<" Error opening file" << endl;
   return 1;

 }
   char ch;
int digits = 0 ; spaces = 0;

while (file . get (ch)) {
  if (isdigit (ch)) :
     digits ++
else  if (isspace (ch))
   spaces ++ ;

}

cout <<" Digits " << endl;
cout <<" spaces :" << endl;

  file . close () ;
   return 0;

}
```

Q3] →

```
#include <iostream>
#include <fstream>
#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)
   cout ++ ;

cout << " Total words : " << cout << endl;

file.close ();
  return 0;

}
```

Q4]

→

```
#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>> target;

while (file >> word)
  if (word = target)
    count ++;
}
~~count~~
cout <<" occurance of 1 "<< target << endl;

    file.close ();

}
```

Experiment 10

1]

```
# include <iostream>
using namespace std;
template <class T>
 T arr sum (T arr , int n ) {
T sum = 0 ;
for (int i=0 ; i<n ; i++)
sum + = arr [i] ;
   return sum ;
   }
 int main () {
 int arr 1 [5] = { 1 , 2 , 3 , 4 , 5 }
cout << " Sum of Array : " << arr sum (arr 1)
 }
```

Output :

Sum of array : 15.

2]

```
#include < iostream>
using namespace std;
template <class T>
T sq (T n) {
  return n * n ;
        }
```

```
template ()
string sq <string> (string s) {
   return st.s ,
        }

int main () {
int , = 5;
string str = "ABC" ;
cout <<" square of integer " << sq (i) .<<endl
cout <<" square of string :" << sq (str) << endl
}
```

Output :-

Square of Integer : 25
square of string : ABCABC

c]

```
# include <iostream>
using namespace std;
template < class T>
class case { T num1 , num2;
Public :
   Case (Ta , Tb) {
num 1 = a ;
num 2 = b;
}
   void display () {

cout <<" Addition :" << num1 + num2 ;
```

```
cout << " Subtraction :" << num1 - num2 ;
cout << " Multiplication :" << num1 * num2;
cout << " Division :" << num1 / num2;
cout << " Modulus : " << num1 / num2;
cout << " Square of Num1 : " << num1 * num2;
cout << " Square of num2 : " << num * + num2;
cout << " cube of num 1 : " << num1 * num1 *
                                num1 ;
cout << " Average : " << (num1 + num2) /2;
}
};

int main (){
int a,b;
cout << " Enter two Integers : ";
cin >> a >> b;
case (int x as (a,b) ;
cout << " Calculate :" << endl;
}
```

Output :-

```
Enter ~~too~~ two integers : 10 5
 calculator Addition : 15
    Subtraction : 5
    Multiplication : 50
    Division : 2
     Modulus : 0
   square of num 1 : 100
   square of num 2 : 25
     Average : 7
```

d]

→

```
# include <iostream>
using namespace std;
template < class T>
class stack { Tarr [5] : int top ;
public :
stack () { top = -1 : }
void push (T val {if (top ==4)
cout << "stack overflow" }.
else
  arr [++top] = val. ;
 }

void pop () { cout << "stack underflow".
else
cout << "Popped " << arr [top--] << endl;
void disp () { cout "stack elements:" ;
for (int i = 0 ; i <= top ; i++) )
cout << arr [i] << " " ;
cout << endl ;
}
};
int main () {
stack <int > s;
s push (10);
s push (20) ;
s. push (30) ;
s. display () ;
s. pop ();
s. display () ;
}
```

output :

stack element : 10 20 30
popped : 30
stack element : 10 20.

Experiment 11.

1]

→

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

int main (){
   vector (int) v = {1,2,3,4,5,6,7,8,9,10}

cout <<" Initial vector :" << endl;
for (int i=0 ; i <10; i++ ) {
cout << v[i] <<" " << endl;
}

cout <<" multiply by 10" << endl;
for (int i=0 ; i<10; i++) {
  v[i] = v[i] * 10;
}

cout <<" New vector:" <<endl;
for (int i=0 ; i<10; i++) {
cout <<v[i] <<" " <<endl;
}

return 0;

}
```

2]

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

int main () {

vector <int> v = {1,2,3,4,5, 6,7,8,9,10}

cout << "Initial vector: " << endl;
for (vector <int> : iterator it = v.begin ) ++ ;

cout << endl;
cout << " ~~Num~~ Multiply by 10" << endl;
for (vector <int> : iterator it = v.begin() it i=
    v.end (); ++ it) {
    * it = (* it) * 10;
  }

cout << "New vector: " << endl;
for (vector <int> :: iterator it = v. begin ();
cout << * it << " ";
  }
    count << endl;
    return 0;
}.
```

Experiment 12

1]

→

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

stack <int> stack1;

void display () {
    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);
}
```

```
cout << endl;
cout << "Top most element": << stack.top();
cout << endl;
cout << "stack elements (top to bottom)" << endl;
disp()
cout << endl;
cout << "Top Function: " << endl;
stack 1. pop();
disp();
stack 1. pop();
disp();
stack 1. pop();
disp();
}
```

2]

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

int main () {
  queue <int> a;
  for(int i=0; i<=10; i++) {
    q.push (i*10);
}

  cout << endl;
  cout << "front elements:" << a.front()<<endl;
cout << endl;
  q.pop();
```

```
cout << " After one pop . front = " << front() << endl;
cout << endl;
cout << " Queue elements (front to back ) : ";
while ( q . empty ()) {
cout << q . front () << " ";
    }
}.
```