Experiment - 1

WAP in C++ lang to declare a class Student having data members as roll_no and ~~na~~ name. Accept and display data for single student

```
# include <iostream.h>
using namespace std;
class student
{
int rollno;
String name;
Public :-
void accept()
{
```

```
cout << " Enter Student name and roll_no
        cin >> name >> roll_no;
  }
  cout << " Name of Student :" << name;
  cout << " Role number of Student : " << roll_no;
  }
int main ()
{
 Student S1;
 S1. accept ();
 S1. display ();
 return 0;
 }
```

Q Write a C++ code to create a class book having data members as bname, bprice bpages. Accept the data for two book and display the name of book having greater price.

```
# include <iostream.h>
 Using namespace std;
 class classbook
 { public :
   ~~Str bprice ;~~
   ~~int bpages ;~~
   int bprice;
 Public :-
  void accept ()
  {
  cout << " Enter book name :" << bname;
  cout << " Enter book price : " << bprice;
```

```
cout << " Enter book pages :" << bpages;
  }

void display ()
  {
   cout << " Name of the book :" << name,
   cout << "Price of the book :" << bprice
   }
   }
   int main()
   {
    book b1, b2;
    b1. accept();
    b2. accept();
    if (b1 bprice > b2 bprice)
      {
       b1 display ();
      }
      else
      {
        b2 display ();
      }
      }
```

output :-

Enter the name, id, price of the book
a 459 1234

Enter the name, id, price of the book
b 500 1245

Name of the book : b
Price of the book : 500
Id of the book : 1235

~~② Write a c++ program to declare a class Book having data members as id, name, price. Accept data for 2 books and display data and book having great~~

③ Write a program to declare a class time having data members as H, M, S. Accept and display data for one object and total time in Seconds

```
-> # include <iostream>
   using namespace std;
   Class time
   {
    Public :
      int H;
      int S;
    Public :
    void accept ()
    {
     cout << " Enter time in hours, minutes second
     cin>> H>> M>>S; }
     void calculate ()
     { H = H * 3600;
       M = M * 60;
       S = S + H + M; }
     void disp()
    { cout << "Total time in Seconds = " << S; }
```

```
int main()
{ time T1;
T1.accept();
T1.calculate();
T1.disp();
return 0; }
```

Output:-
Enter the time in hours minutes seconds: 2 46 31
Total time in seconds = 9997

29/7/25

Experiment -2

① WAP to declare a class 'city' having data members as name and population. Accept this data for 5 cities and display name of city having highest population.

→
```
# include <iostream>
using namespace std;
class city
{
public:
String name;
int population;
void accept()
{
cout << " Enter city name:";
cin >> name;
cout << " Enter city population:";
cin >> population; }
void disp()
{
cout << "city having highest population:" << name
}
};
int main()
{
city C[5];
int i, max;
for (i=0; i<5; i++)
{
C[i].accept(); }
max = C[0].population;
```

```
for (i=0; i<5; i++)
  {
  if ((c[i].population) > max)
  { max = i; }}
  c[max].disp();
  return 0;
```

Output:-

Enter city name: mumbai
Enter city population: 786
Enter city name : Pune
Enter city population: 212
Enter city name : Kuwait
Enter city population: 108
Enter city name : Salmiya
Enter city population: 81
Enter city name : Uttrakhand
Enter city population: 92
City having highest population: mumbai.

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;
float balance;
void accept()
{
cout << " Enter Account Number: ";
cin>> acc_no;
cout << " Enter balance:";
cin>> balance;
}
void disp()
{
cout << " \n Account Number:" << acc_no;
cout << " \n Account Balance:" << balance;
}
};
int main()
{
Account A[10];
int i;
for (i=0; i<10; i++)
{
```

```
{
 A[i].accept();
  }
 for (i=0; i<10; i++)
 {
  if (A[i].balance >=5000)
  {
   A[i].balance = A[i].balance+ (0.1* A[i].balance);
   A[i].disp();
   }
   }
 return0;
  }
```

Output:-

```
Enter Account Number: 2564
 Enter balance : 2095
Enter Account Number: 4879
 Enter balance : 7890
Enter Account Number: ~~678~~ 4867
Enter ~~Account~~ Balance: 3528
Enter Account Number: 2478
Enter balance: 3678
Enter Account Nuber: 2634
Enter balance: 9705
Enter Account Number: 5738
Enter balance: 57228
Enter Account Number : 5835
Enter balance: 8789
Enter Account ~~Balance~~ Number : 4628
Enter Balance: 8906
Enter Account Number : 7568
Enter Balance: 4689
Account Number: 7890
 Account Balance : 7467.9
Account Number : 2634
Account Balance : 10675.5
Account Number : 5738
Account Balance : 62950.8
Account Number : 4627
Account Balance : 9796.6
```

③ WAP to declare a class 'Staff' having data members as name and post. Accept this data for 5 staff and display names of staff who are "HOD"

```
-> #include <iostream>
using namespace std;
 class Staff
 {
  String name;
  Public ~~Name~~:
  String post;
  void accept()
  {
   cout << "Enter the staff name:")
   getline (cin, Name);
   cout << "Enter the staff post:";
   getline (cin, post);
    }
   void disp()
    {
    cout << "\n ~~cout~~ 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].disp();
}
},
return 0;
}
```

output:-

Enter the name of the staff Riya
Enter the post of the staff HOD
Enter the name of the staff Siya
Enter the post of the staff Accountant.
Enter the name of the staff Diya.
Enter the post of the staff HOD.
Enter the name of the staff Priya.
Enter the post of the staff teacher.
Enter the name of the staff Ena.
Enter the post of the staff Accountant.
The names of the staff who are HOD are
Riya
Diya.

29/7/25

Experiment - 3

a WAP to declare a class 'book' containing data members as book title, author_name & price Accept and display the info for 1 object using pointer to that object

```
# include <iostream>
using namespace std;
class book
{
string title;
string author_name;
int price
public:
void accept()
{
cout << " Enter book title";
cin >> title;
cout << " Enter author name";
cin >> author name;
cout << " Enter book price";
cin >> price;
}
void display()
{
cout << " Book title:" << title;
cout << " Author name:" << author_name
cout << " Book price" << price;
}
};
int main()
{
book b1;
```

```
book* p;
p = &b1;
p -> accept();
p -> display();
}
```

output:

```
Enter book ~~name~~ title: Harry
Enter author name : Ron
Enter Price of book: 25000
Book title : Harry
Author Name: Ron
Book price: 25000
```

② WAP to declare a class 'Student' having data member roll.no & percentage. Using 'this' pointer invoke member functions to accept & display this data for one object of class.

→
```
#include <iostream>
using namespace std;
class Student
{
int rollno;
float p;
public :
void accept ()
{
cout << " Enter roll_no: ";
cin >> this -> roll_no;
cout << " Enter percentage: "
cin >> this -> p;
}
void display ()
{
this -> accept();
cout << " Rollno " << roll_no;
cout << " Percentage " << p;
}
};
int main ()
{
Student s1;
s1 display();
return 0;
}
```

Output:-

```
Enter Rollno: 72
Enter Percentage : 100
Rollno: 72
Percentage: 100
```

③ WAP to demonstrate the use of nested class.

```
# include <iostream>
using namespace std;
class student
{
 public;
string name;
int roll;
void accept ()
{
cout << " Enter roll number \n";
cin >> roll;
cout << " Enter name \n";
cin >> name;
}
void display ()
{
cout << " Roll no \n" << roll;
cout << " Name \n" << name;
}
class marks
{
int m1, m2;
public:
void accept ()
{
cout << " Enter mark of two subject \n";
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 number
75
Enter name
mufaddal
Roll no : 75
name : mufaddal
Enter marks of two subject
54
61
percentage = 115.
```

5/8/25

experiment - 4

① WAP to swap two numbers from same class using object as function arguments write swap function as member function.

```
-> # include <iostream>
   using namespace std;
   class numbers {
   private :
     int a, b;
   public :
   int temp;
   void accept () {
   cout << " Enter the 1st number : ";
    cin >> a;
    cout << " Enter the second number : ";
    cin >> b;
    }
    void display () {
     cout << " In After swapping : " << endl;
     cout << " first number = " << a << endl;
     cout << " Second number = " << b << endl;
     }
    void swap (numbers &n)
    {
    n.temp = n.a;
    n.a = n.b;
    n.b = n.temp;
    }
    int main () {
     numbers n;
     n. accept ();
     n. swap (n);
```

```
n. disp ();
return 0;
}
```

Output :
Enter the first number : 91
Enter the second number : 28

After swapping :
first number : 28
Second number : 91

② WAP to swap two number from same class using concept of friend function.

```
# include <iostream>
using namespace std;
class number {
  private :
  int a, b, temp;
  public :
    void accept () {
  cout << " Enter the first number:" ;
    cin >> a;
  cout << " Enter the second number:" ;
  cin >> b;
  }
void display () {
cout << " In After swapping :" << endl;
cout << " First number = " << a << endl.
cout << " Second number = " << b << endl;
}
```

```
friend void swap ( number &);
};
void swap (number &n ) {
  n. temp = n.a
  n.a = n.b;
  n.b = n.temp;
}
int main ()
{
  number n;
  n. accept ();
  swap (n);
  n. display ();
  return 0;
}
```

O/P :-

Enter the first number : 1
Enter the second number : 9

After swapping
First number : 9
Second number : 1

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

```
# include <iostream>
using name space std;
class num2;
class num1 {
    private :
      int a;
    public :
     void . accept () {
     cout << " Enter the first number 1";
      cin >> a;
     }
     friend void swap ( num1 & num2 &)
     };
     class num 2 {
         private :
         int b;
         public :
       void accept () {
       cout << " Enter second number :";
       cin >> b;
       }
       friend void swap (num1, num2 &);
       };
       void swap (num1 & , number2. &y) {
       int temp;
       temp = x . a;
       x . a = y . b;
       y . b = temp;
       cout << " After swapping :" << endl;
       cout << " second number = " << y.b << endl;
       }
       ~~int a. accept ();~~
       int main () {
       num1 n1;
       num2 n2;
       n1 . accept ();
       swap (n1, n2);
       return 0;
       }
```

output:

Enter the first number = 5
Enter the second number = 6.

After swapping :
First number = 6
Second number = 5 .

4 WAP to create 2 classes result1 and Result2 which ~~Stores~~ Stores the marks of the students. Read the value of marks for both the class objects and compute the average of 2 Results.

```
-> # include <iostream>
   using namespace std;
   class result 1
   {
   private:
     String name;
     float marks;
   Public :
     void accept()
   {
     cout << " Enter student name:";
     getline (cin, name");
     cout << " Enter the total marks obtained
          in first semester out of 10:";
     cin >> marks;
   }
     friend void average (result 1,
                          result2);
   };
   class result 2
   {
     private:
       float marks;
       public:
     void accept ()
       {
```

```
cout << " Enter the total marks obtained in
     Second semester out of 100:";
cin >> marks;
}
friend void average (result 1, result 2);
};
void accept ()
{
cout << " Enter the total marks obtained in
     Second semester out of 100:";
cin >> marks;
}
friend void average ( result 1, result 2);
};
void average ( result1 x, result2 y)
{
  float avg;
    avg = (x.marks + y.marks)/2;
  cout << " \n Average of both the
          result =" << avg;
}
int main()
{

result r1;
result r2;
r1. accept();
r2. accept();
average (r1, r2);
return 0;
}
```

Output:-

Enter the name: Mufaddal
Enter the total marks obtained in first Semester out of 100: 98
Enter the total marks obtained in Second Semest out of 100: 86
average result.

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

→
```
# include <iostream>
using namespace std;
class num2;
class num1;
{
private:
int a;
public:
void accept()
{
cout << " Enter the end number";
cin >> b;
}
friend void get (num1, num2);
};
void get (num1 x, num2 y)
{
if (x.a > y.b)
{
cout << x.a << " is greater than" << y.b;
}
else {
cout << (y.b << " is greater than" << x.a;
}
}
int main () {
num n1;
num2 n2;
n1. accept ();
n2. accept ();
get (n1, n2);
return 0;
}
```

output:-
Enter the first number : 8
Enter the second number : 10
10 is greater than 8

6 Create 2 classes, class A and class B, each with a private. Write a frnd sum() that can access private data from both classes & return the sum.

```
# include <iostream>
using namespace std;
class B;
class A {
private:
int a;
public:
void accept () {
cout << "Enter the first number";
cin >> a;
}
```

```
friend void sum (A,B);
};
class B {
    private:
     int b;
    public:
    void accept ()
  { cout << "Enter the second number";
    cin >> b;
  }
  friend void sum (A, b);
   };
   void sum (Ax , By)
   {
     int sum;
     sum = x.a + y.b;
     cout << "sum =" << sum;
   }
   int main () {
     A n1;
     B n2;
     n1. accept ();
     n2. accept ();
     sum (n1, n2);
   }
output :
Enter the first number : 5
Enter the second number: 8
  sum = 11.
```

Q7 WAP with a class Number that contains a priv integer. Use a friend function swap numbers (numbers, Numbers) to swap the private values of two

->

```
# include <iostream>
  using namespace std;
  class Number {
    private :
      int a, b;
    public :
      void accept () {
    cout << " Enter the first number";
     cin>> a;
     }
     void accept () {
     cout << " Enter the second number";
      cin>> b;
      }
     void display () {
      cout << "Second Number =" << b;
      }
     friend void swap (Number & x, Number
      };
     void swap (number & x, Number & y)
       int temp;
       temp = x . a;
        x = a = y . b;
        y . b = temp;
          }
       int main () {
       number n1;
       number n2;
```

```
Swap (n1, n2)
n1. display ();
n2. display ();
return 0;
}
```

Output

Enter the first number : 7
Enter the second number : 4

After swapping
first number : 4
second number : 7

(8) Define 2 classes box and cube, each having a private ~~function~~ volume. Write a friend function friend (Box Cubes) that determines which object has a larger volume

```
# include <iostream>
using namespace std;
class Box;
class cube {
    private
    float volume;
    public :
void accept () {
cout << " Enter the volume of a cube :";
cin >> volume;
}
friend void find grt (Cube, Box);
};
class Box {
private :
float vol;
public :
void accept ()
{
cout << "Enter the volume of the Box :";
cin >> vol;
friend void find grt (Cube, Box);
};
void find grt (Cube x, Box y) {
if (x. volume > y. volume) {
else {
cout << " Box has larger volume";
}
}

int main () {
Cube c;
Box b;
c. accept ();
b. accept ();
find grt (c, b);
return 0;
}
```

Output:
Enter the volume of a cube : 567. 88
Enter the volume of a box : 485. 66.
Box has larger volume.

(9) Create a class complex with real & imaginary parts as private number. Use a friend function to add two complex numbers & return the result as a new complex object.

```
-> # include <iostream>
   using namespace std;
   class complex {
     private :
       int r, i;
     public :
       void accept ()
       cout << " Enter real part:",
       cin >> r;
       cout << " Enter real part:";
       cin >> i;
       }
       void display ()
         cout << r << " + " << i << " i" << endl;
         }
       friend complex sum ( complex, complex)
         },
       complex sum (complex x, complex y) {
         complex temp;
         temp.r = x.r + y.r
         temp.i = x.i + y.i;
         return temp;
         }
     int main ()
     complex c1, c2, c3;
     cout << " Enter the first complex no:
```

```
c1. accept ();
cout << " Enter the second complex number;
c2. accept ();
cout << " complex Num 1";
c1. display ();
cout << " Complex number 2";
c2. display ()
c3 = sum (c1, c2);
cout << " Sum of 2 complex number";
c3. display ();
return 0;
}
```

Output:

```
Enter the 1st complex number:
Enter the real part : 5
Enter the imaginary part: 6
Enter the second complex number.
Enter the real part: 8
Enter the imaginary part: 2
complex Number 1:
5+6i
Complex Number 2:
8+2i
Sum of the two complex numbers = 13+8i
```

10 Create a class Student with private data members: ~~m~~ name and three Subject marks. write a friend function calculate Avg. (Student) that calculates & display avg marks.

-> # include (iostream)
using namespace. std;
class Student &
String name;
int math;
int phy;
int chem;
Public :
void accept ()&
cout (<" Enter Student name:",
cin >> name;
cout (<" Enter maths maks",
cin >> maths;
cout (<" Enter physic marks";
cin >> phy;
cout (<" Enter chem marks" a;
cin >> chem;
}
friend void calculate Average (Student)
};
void calculate Averag (Student x)
&
float avg;
avg = (x. math + x.phy + x.chem)/3;
cout (<" Avg marks:" (< avg;
}.

int main () &
Student s;
s. accept ()
calculate Average (s)
return 0;
}
Output:
Enter Student name: Mufaddal
Enter maths marks: 98
Enter physics marks: 96
Enter chem marks: 92

Avg marks : 95

Experiment -5

a)
i

```
# include <iostream>
using namespace std
class sum {
  int n, i, sum=0;
  public :
    sum ()
    {
      n= 20;
    }
    void calculate ()
    {
    for (i=1; i<=n ; i++)
      {
        sum= sum +i;
      }
    }
    void display ()
    {
    cout <<" sum = " << sum;
    }
    };
int main ()
{
sum n;
n. calculate ();
n. display ();
return 0;
}
```

Output:

sum= 20

(ii) Using parameterized construction

```
# include <iostream>
using namespace std;
class sum
{
int n, i, sum=0;
public:
  sum (int num)
  {
  n= num;
  }
void calculate ()
{
for (i=1; i<=n; i++)
  {
  sum= sum +i;
  }
  }
void display ()
{
cout << " sum=" << sum;
  }
  };
int main ()
{
sumn (20);
n. calculate ;
  n. display ;
  return 0;
  }
```

output:

sum = 210.

b

```
# include (iostream)
Using namespace std;
Class Student
{
int per;
String name;
public:
Student ()
{
name = "Kamya";
per = "98";
}
void display ()
{
cout << "Name:" << name,
cout << " Percentage:" << per;
}
};
int main () {
Student s;
return 0;
}
```

~~Name: k~~

Output

Name: Mufaddal
Percentage: 98

(11) Parametrized Constructor

```
# in Clude (iostream)
Using namespace std;
Student
{
int per;
String name;
public:
Student (String n, int percent)
{
name = n;
per = percent;
}
void display ()&
Cout << " Name:" << name;
Cout << "Percentage:" << per;
}
};
int main()
{
Student s(" ~~Kamya~~ mufadd", 96)
s. display
return 0)
}
```

Output:
Name = kamya
Percentage = 96.

c

```
# include (iostream)
using namespace std;
class college
{
int roll no;
String name, course;
public:
college ()
{
name= " mufaddal"
roll no = 35
course= " computer engineering";
}
college (string a, int b, String c="computer engineering")
{
name= a;
roll no = b;
course = c;
}
void display ()
{
cout (<" \n name is :" << name;
cout << " \n roll no is :" << roll_no;
cout << " \n course is :" << course;
}
};
int main ()
{
college c1 (" mufaddal", 35), c2 (" kamaji", 71);
c1. display ();
return 0;
}
```

d

```
# include (iostream)
using namespace std;
class department
{
int not, nos;
string n;
public:
department ()
{
n= " it";
nof = 20;
nos = 80;
}
department (string a, int b, int c)
{
n = a;
not = b;
nos = c;
}
department (int x, int y)
{
name = " yashwal";
nof = x;
nos = y;
}
void display ()
{
cout << " \n name is :" << n;
```

```
cout<<"\n number of faculty is :"<<nof;
cout<<"\n number of Student is: "<<nos;
}
};
int main ()
{
    department d1();
    d2("william", 80,60);
    d3( 80,75);
    d1. display ();
    d2. display ();
    d3. display ();
    return 0;
}
```

Experiment - 06

①
```
#inClude (iostream)
using name Space Std;
Class Person
{
protected
String name;
int ~~Ad~~ age;
}
Class Student : protected Person
{
int roll no;
public :
void accept ()
{
Cout << "Enter Name";
Cin >> name;
Cout << "Enter Age:";
Cin >> age;
Cout << "Enter Roll number:";
Cin >> roll no;
}
void display ()
{
Cout << "Name:" << name << endl;
cout << "Age:" << age << endl;
cout << "Roll Number :" << roll no;
}
};
int main ()
{
Student S1;
S1. accept ();
S1. display ();
return 0;
}
```

Output:

Enter Name : Mufaddal
Enter age : 17
Enter Roll Number : 75
Name : Mufaddal
Age : 17
Roll Number : 75.

②
```
# include (iostream)
using name space Std;
Class academics
{
protected :
int maths;
}
Clas sports
{
protected :
int ss;
}
Class Result : public Academics, public Sp
{
public:
int ts;
void accept ()
{
```

```
cout << "Enter marks:";
cin >> marks;
cout << "Enter sports score:";
cin >> ss;
}
void calculate ()
{
ts = marks + ss;
cout << "total score:" << ts;
}
int main()
{
result r;
r. accept()
r. calculate ();
return 0;
}
```

O/P:-

Enter marks: 45
Enter sports score: 30
Total of score : 75.

③

```
# include <iostream>
using namespace std;
class vehicle
{
public:
string branch;
string model;
};
class car: public vehicle {
string attribute;
};
class electric car : public car {
varchar battery capacity;
public:
void accept ()
{
cout << "Enter branch:";
cin >> branch;
cout << "Enter model:";
cin >> model;
cout << "Enter attribute;
cin >> attribute
cout << "Enter battery capacity:";
cin >> battery capacity;
}
void display ()
{
cout << "Branch:" << branch;
cout << "Model:" << model;
cout << "Attribute:" << attribute;
cout << "Battery capacity:" << battery capacity
}
}
int main ()
{
electric car c1;
c1. accept ();
c1. display ();
return 0;
}
```

O/P :-
Enter branch : Suzuki
Enter model : land cruiser
Enter Attribute : SUV
Enter battery capacity : Not electric
Branch : Suzuki
Model : land cruiser
Attribute : SUV
battery capacity : Not electric.

```
(9)  # includs <iostream>
     using name space std;
     class employee
     { public:
       int empid;
       string name
       };
       class Manager : Public employee
       {
         String department;
         public:
           void accept()
           {
           Cout << "Enter employe id" << empid;
           Cout << " Name :" << name;
           Cout << "Department: " << department;
           }
           };
           Class developer : Public employee
           {
           String programming language;
           Public:
```

```
void accept()
{
Cout << " Enter employee id";
Cin >> empid;
Count " Enter name";
cin >> name;
Count << " Enter programming language";
Cin >> Programming language;
}
void display()
{
Cout << " Employee id :" << empid;
Cout << " Name :" << name;
Cout << " Programming language :" << programming language;
}
};
int main()
{
Manager m;
m. accept();
developer d;
d. accept();
d. display;
return 0;
}
```

Output:
Enter employee id : 101
Enter name : LUCKY
Enter department name : CSE
employee id : 101
Name : Lucky
Department : CSE

Enter employee id : 201
Enter Name : Rohan
Enter programming language : Java
employee id : 201
Name : B
Programming language : Java.

⑤
```
# include <iostream>
using namespace std;
class Person
{
Protected:
String name;
int age;
}
protected :
void accept ()
{
cout << "Name of Person:";
cin >> name;
cout << "Age of Person:";
cin >> age;
}
void display()
{
cout << "Name:" << name;
cout << "Age:" << Age;
}
};
class sports
{ protected:
int sports score;
};
```

```
Class academics
{
protected:
int academic score;
};
Class result : protected sports, academics
{
protected:
int total;
void accept ()
{
cout << " Enter sports score:";
cin >> sportsscore;
cout << " Enter academic score:";
cin >> academic score;
}
void calculate()
{
total = sportsscore + academicscore;
}
void display ()
{
cout << " Total = " << total;
}
};
int main () {
Students,
s. accept ();
s. display ();
Result r;
r. accept ();
r. calculate ();
```

```
v. display ();
return 0;
}
```

Output:
Name of Person: A
Age of Person : 13
Name : A
Age: 13
Enter sports Score: 45
Enter academic Score: 45
total = 45+45
total = 90.

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.calculate (10,8)
        << endl;
    cout " Area of square : " a. calculate (5) << endl;
```

②
```
# 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 ] {
```

```
v. display ();
return 0;
}
```

Output:
Name of Person: A
Age of Person: 13
Name: A
Age: 13
Enter sports score: 45
Enter academic score: 45
total = 45+45
total = 90.

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.calculate(10,5)
        << endl;
    cout " Area of square : " a. calculate (5) << endl;
```

②
```
# 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++)
    St = a[i];
return S;
}
};
int main () {
    Sum S;
int marks [10]= { 45, 56,67, 78, 89,90, 76,88,
                 92,85};
float graders [5]= [9.2, 8.7, 9.5, 8.9, 90].

cout << "Sum of 10 student marks:" << S.total
        (marks, 10)<< endl;
cout << "Sum of 5 student grade points :"
        << S. total (grades, 5) << endl;
return 0;
}.
```

③

```
# include <iostream>
using namespace std;
class teacher {
    int experince;
public:
    Teacher (int e) {
    experience = e;
    }
    void display () {
    cout << " Experience:" << experience <<
            "year" << endl;
    }
    void operation. () {
experience = experience;
    }
};
int main () {
    teacher t1(10);
    t1 . display ();
    t1;
    cout << " After negation :";
    t1. display ();
    return 0;
}
```

④

```
# include <iostream>
using namespace std;
class Student {
    int count;
public:
    Student (int c=0) {
    count = c;
    }
    void operator ++ () {
    ++ count;
    }
    void operation ++ (int) {
    count ++;
    }
    void display {
    cout << "Student count :" << count << endl;
    }
};
int main () {
```

```
Student S1(50);
Cout << "Before increment : " << endl;
S1. display();

++ S1;
Cout << "After post- increment :" << end;
S1. display();
S1++;
Cout << "After post-increment :" << endl;
S1. display();
return 0;
}
```

Experiment - 8

①

```
# include <iostream>
# include <string>
using name space std;
class Combine &
    String str;
Public:
    Combine & String S=" "){
    String=S;
    }
~~Public:~~
    ~~Combine (String S=" ")~~
    Combine operator + (combine & obj) &
    return combine (str+ obj. str);
    }
    void display () &
        Cout << str << endl;
    }
    }
int main () &
    Combine S1 ("xyz"), S2 ("pqr"), S3;
    S3 = S1 + S2;
    cout << " Contenated String:"
    S3. display;
    }.
```

②

```
# 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 << "Enter Email ID:";
    cin>> email;
    login:: accept ();
  }
  void display () override {
    cout << "\n ...email login detail...";
    cout << "Email ID:" << email << endl;
    I login:: display ();
  }
```

```
class membership login: public << I login {
        string member ID;
  public:
    void accept () override {
    cout << "Enter membership ID:";
    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 = &m;
  login -> accept ();
  login -> display ();
    return 0;
  }.
```

Experiment - 9.

①
```
# include <iostream>
# include <fstream>
using namespace std;
int main () {
ifstream fin ("first.txt");
ofstream fout ("second.txt");
if (!fin) { cout << "first.txt not found", return 0;
String line;
while (getline (fin, line)) {
fout << line << endl;
}
fin.close ();
fout.close ();
cout << " Content copied successfully";
return 0;
}
```

Output:
first.txt not found.

②
```
# include <iostream>
# include <fstream>
using namespace std;
int main () {
ifstream fin ("first.txt");
if (!fin) { cout << " file not found!\n";
return 0; }
Char ch;
int digits=0, spaces=0;
While (fin.get (ch)) {
char ch;
int digits=0, spaces=0;
while (fin.get (ch)) {
if (is digit (ch)) digits ++;
if (ch= ' ') spaces ++;
}
fin.close();
cout << "Digits:" << digits << " spaces" << spaces;
return 0;
```

Output:-
File not found!

③
```
# include <iostream>
# include <fstream>
# include <sstream>
using namespace std.
int main () {
ifstream fin ("first.txt");
if (!fin) {
cout << " file not found!" \n};
return 0;
}
String word;
int count = 0;
while (fin >> word) {
count ++
}
find close ();
cout << " total words:" << count << endl;
return 0;
}
```

Q4
```
# include (iostream)
# include (fstream)
# include (sstream)
using namespace std;
int main () {
int main () {
if (!fin) { cout (("file not found!", return;
String word, target;
int count = 0;
cout (("enter word to search";
cin >> target;
while (fin){
    if (word == target) count ++;
}
fin. close ();
cout (("The word" (( target (("occurred"
    (( count (("time".
return 0;
}
```

output:
file not found!

Q4
11/4

Experiment-10.

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

O/P:-
Sum of array: 15.

②
```
# include (iostream)
# include (string)
using name space std;
template (class T)
T sq (T n) {
    return n*n;
}
template <>
String sq (String > astring ) {
    return s+s;
}
int main () {
int i = 5
String str = "ABC";
```

```
cout << " Square of Integer : " << sq(x) << endl;
cout << " Same of String " << sq(Str) << endl;
}
```

output:
Square of Integer: 25
Square of String: ABCABC

③

```
Calculator (10, operations)
-> #include <iostream>
using namespace std;
template <class T>
class calc { T num1, num2;
  public:
    calc (T a, T b){
  public:
    calc (T a, T b){
    num1 = a;
    num2 = b;
    }
    void display (){
    cout << " Addition :" << num1, num2;
    cout << " Subtraction :" << num1 - num2;
    cout << " multiplication :" << num1 * num2;
    cout << " modulus :" << num1 % num2;
    cout << " Square of num1 : " << num1 * num1;
    cout << " Square of num2 : " << num2 * num2.
    cout << " Cube of num1 : " << num1 * num1 * num1
    cout << " Cube of num2 : " << num2 * num2 * num2
    cout << " Average : " << (num1 + num2)/2
    cout << " max : " << ((num1 > num2) ? num1 : num2
    cout << " min " << ((num1 < num2) ? num1 : num2
```

```
int main ()
int a, b;
cout << " Enter two integers:";
cin >> a >> b;
calc* (int) calc (a,b);
cout << " calculate :" << endl;
cal c. disp();
}
```

④

```
# include <iostream>
using namespace std;
template <class T>
class Stack { T arr[5]; int top;
  public
  Stack () { top = -1; }
  void push (T val) { if (top == 4) cout << "
    Stack overflow";
  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. pop (30);
s. display ();
s. pop ();
s.display ();
}
```

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 (with iterators)

```
# 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;
    # it) &
    cout (( *it (<" ";
```

```
cout (( endl;
cout (( multiply by 10" (<endl;
for (vector (int): iterator it = v.begin(): it (=
    v.end(); ++ it) &
    * it = cout (<" " ;
    }
cout (<endl;
return 0;
}.
```

11/11

experiment-12

①
```
# include (stack)
# include (iostream)
using namespace std;
stack (int) stack;
void disp () {
    stack (int) temp= stack;
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; (++)){
int temp;
cout ((("Enter element at position"
((i+1! (( endl;
cin>> temp;
stack! push (temp);
}
cout(( endl;
cout((("Top most element:" ((stack
top() ((endl;
cout (( endl;
cout (( "Stack elements (top to
Bottom) - - ->" ~~elem~~ endl;
disp()
cout((endl;
```

```
cout (((" Pop function:" ((endl;
stack ! .. pop ();
disp()
stack! . pop();
disp ();
}
```

②
```
# 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(( "Back element" ((q.back() ((endl;
cout(( endl;
q. pop()
cout(((" After one pop . front=" ((q. front() ((endl
cout(( endl;
cout(((" Queue elements (front to back:)";
while (!q. empty (i)){
cout(( q. front () (((" ";
}
}.
```

Qu
1111