Experiment 1

Q Write a C++ code to declare a class and and access roll no. and na

Code:-

```
# include<iostream>
using namespace std;
Class Student {
    int roll_no;
    char name;
    public:
    void accept () {
        cout << "enter name and roll no";
        cin >> name >> roll_no;
    }
    void display () {
        cout << "name of student" << name << endl;
        cout << "roll no -" << roll_no;
    }
};

int main () {
    Student s1;
    s1.accept ();
    s1.display ();

    return 0;
}
```

output

enter name and roll no e e
s
name - e

Q2 Code

```
# include <iostream>
using namespace std;
class Books {
        public:
         int bprice;
         int bpages;
         string bname;
         void accept() {
             cout << "print book name pages and price" << endl;
             cin >> bname >> bpages >> bprice
         }
         void disp() {
             cout << "the name is " << bname << endl <<
             "the pages" << bpages << endl;
         }
    };

int main()
{
    Book b1, b2;
    b1.accept();
    b2.accept();
    b1.disp();
    b2.dispp();

    if (b1.price > b2.bprice) {
        cout << "the pric of first book is higher" << endl;
    }
```

else {
cout << "The price of second is higher" << endl;
}

Output :-

~~The name is~~
Book name, pages and price e
if a f1
4
2

fs
3
6

The name is if a f1
The pages are 4
The price is 2

The name is fs
the pages are 3
the name is 6

The second book has a higher pric

Q Write a program to declare Class time accept time in HH:mm:ss format, convert it to total second and display the

Code :-

```
# include < iostream>
using namespace std;
class time {
    public:
    int hour, min, sec;
    void accept () {
    cout<<"enter time in hh mm ss";
    cin >> hour>> min >>sec;
    }
    void disp() {
    cout<<"the time in format :"<<hour<<":"<<
    min<<":"<<sec<<endl;
    int sh = hour* 3600;
    int sm = min * 60;
    int tt = sh+sm+sec;
    cout<< "total times in tt"<<tt;
    }
};

int main() {
    class time t;
    t.accept ()
    t. disp ();
    return 0;
    }
```

Output
enter time in hh mm n ss 5
35
7
the Time in format is 5:35:7
total time in seconds 20107

1517

Experiment 2:

a) WAP with class city accept name and pop display ~~highest~~ most populated city

Code:-

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

class city {
public:
    string name;
    int pop;

    void accept () {
        cout << "Enter city name and popu.";
        cin >> name >> popi
    }
    void dis () {
        cout << "city: " << name << ", pop" << pop << endl;
    }
};

int main () {
    city c[5];
    for (int j=0 ; i<5 ; j++) {
        c[i].accept ();
    }
    for (int i=0 ; i<5 ; i++) {
        c[i].disp () ; }
```

```
for (int i=0; i<s; i++){
    [i].disp();
}

for (int i=0; i<s; i++){
    int j;
    for (j=0; j<s; j++){
        if ([j].pop > [i].pop){
            break;
        }
    }
    if (j==s){
        cout << "Intro mosop popcity is" <<
        [i].name << "with pop" << [i].pop
    }
}
return 0;
}
```

Output

Enta city name and pop: mumbai

11

Enta city name & pop: pune

1123

enta city name & pop: banglore

2222

enter city name & pop: pune

22221

enta city name & pop: delhi

3111

The most populous city is: delhi with population

Q2

Code

```
#include stdio.h<iostream>
using namspace std;
class Account{
    public:
     int bal;
     void accept (){
      cout << "prim acc bal";
      cin >>bal;
     }
     void disp (){
       cout<<bal <<endl;
     }
  };

int main (){

Account acc[10];
int i=0;
for (i=0; i< 10;i++){
    acc[i].accept();
    ~~int p~~
    if (acc[i].bal>5000){
      acc[i].bal+= acc[i].bal*0.1;
    }
```

```
for (i=0; i<10; i++) {
        acc[i].disp();
    }
    return 0;
}
```

output

```
print acc bal 1000
prin acc bal 5500
        3000
        6600
        7000
        100
        30000
        100000
        300
        6000
```

```
1000
6050
3000
7760
7700
100
33000
10000
300
6600
```

Q3

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

int main() {
  staff s[5];
  int i=0
  for (i=0; i<5 ; i++){
     s[i].accept();
  }
  for (i ; i<5 ; i++){
     if (s[i].post == "hod"){
        cout << s[i].name << "is the hod";
     }
  }
}
return 0
}
```

O/P

enter name and post lisa
hod

29/7/25

Anjush pa
pune

nina
clerk

nenna
corporate

Eiss is the hod

Exp 3

1)

```
#include <iostream>
using namespace sta

class book {
    char bt;
    char an;
    int prio;
    public:
    void acc() {
      cout<<"print bt an price";
      cin >>bt>>an>>prio;
    }
    void disp() {
      cout << bt <<endl<<an<<prio<<"is printed";
    }
};
int main() {
    book* ptr;
    book b;
    ptr = &b;
    ptr -> acc();
    ptr -> disp();

    return 0;
}
```

Output

print bt an price r

f

2

2) #include <iostream>
using namespace std;
class student {
char name;
int roll
public
void accept () {
cout << "write it pls";
cin >> this -> name >> this -> roll >> this
}
void disp () {
this -> accept()
cout << this -> name << this -> roll << this -> p
}
}
int main () {

student s1
s1.disp(),
return 0
}
O/P
write a
3
4
l
3
4

3)
```
#include <iostream>
using namespace std;
class student {
    public:
    class marks {
        int marks;
        public :
        void accept () {
            cout << "enter the mark for the class";
            cin >> marks;
        }
        void display() {
            cout << "the marks are " << marks;
        }
    };
};
int main () {
    student:: marks m1
    ~~m1.disp~~
    m1.accept ();
    m1.display ();
    return 0;
}
```

Op:
Enter the marks for class 50
the marks are 50

5/8/25

Expt 4

Assignment

1) Problem statement - WAP to swap two num by passing objects as function argument by pass by val consider two no.s same da

Code:

```
#include <iostream>
using namespace std;
class num
{
    int val;
    public:
    void accept()
    {
        cout<<"enter val of num1";
        cin >> val;
    }
    cout<<"enter val of num1"
}
void disp ()
{
    cout << val;
}
void swap(num n1, num n2)
{
    int temp = n1.val;
    n1.val = n2.val
    n2.val = temp
    cout<<"swap is "<< n1.val << endl
```

```
int main ()
{
  num n1, n2, temp;
  n1.accept;
  n2.accept;
  temp.swap (n1, n2);
  return0;
}
```

Output-
Enter value of num1 - 5
— — num2 - 6
num1 : 6
num2 : 5

2) WAP to swap two num by passing objects as function argument by pass by ref

Code:

```
#include <iostream>
using namespace std;
class B;
class A {
    int val;
Public:
  void accept () {
      cout << "enter val for first
      cin >> val;
  void display () {
   cout << "Num1, << n1 << endl
    cout << "num2" << n2 << endl2;
  }
```

```
int main (){

    nu.accept
    nu.display()
    swap(nu
    retarno;
  }
```

output

Enter1 num :- 5
Enter second num :- 9
Num1 : 5
Num2 : 9
Swapped
first : 9
second 5

3) Swap two num from diff
Code:-

```
#include <iostream>
using namespace std;
class nu {
public :
    int n1.
void accept (){
  cout << "enter first num" << endl.
  cin >> n1;
  void display () {
   cout << "Num1:" << n1 << endl }
  }
```

```
Void swap (num1, num2)
    int tem= num1.n1
    num1.n1 =num2.n2
    num2.n2 = temp;
    cout
cout << swapped:\n first " << num1.n1 <<\n second
    << num2.n2 << endl

int main (){
num1 .accept();
num2.accept();
num1.display();
num2 .sdisplay
swap(num1 ,num2
return 0;
}
```

output:-

```
Enter first num: 5
enter second num=9
num:5
num 9
swapped
first :9
second:5
```

A) Create 2 class result which stores the m
at the students hence the val of marks
both the less extra

Code:-

```
# include <iostream>
using namespace std.
public
    int r1
    void accept (){
    cout << "enter first result: " << endl;
  } res1
  class
class Result2{
public:
  int r2;
  void accept (){
  cout << "enter second result" << endl;
  cin >> r2; }
} res2
float avg (result1, result 2) {
    return (res1.r1 + res2.r2)/2;
}
int main (){
  res1.accept ();
  res2.accept ();
  float avg = avg (res1, res2)

  return 0
}
```

Output

Enter first result
95
Enter second result
97

Avg :- 97

5) Find the greatest num among 2 num from differen class using friend function

```
#include <iostream>
using namespace std;
class num1{
public:
    int n;
  void accept (){
cout<<" enter first num :" <<endl;
cin >> n1;}
}num1;
class num2{
public:
    int n2;
void accept (){
cout <<" enter sec num :;
```

→ P10

```
int man () {
  n1.accept()
  int avg = gm (n1.n1, m2.n2);
  cout >> "Grantag:<< nr<<endl;
  return 0;
}
```

Friend function practice questio

1) Create two classes A and B each with a private integer write a freind fun sum ()-that can access private data from both the classe and return the sum

Code

```
# include <iostream>
using namespace std;
Class B;
class A {
  int n1;
  public:
  void accept()
  {
    cout << "enter num 1: "
    cin >> n1;
  }
  friend int sum(A, B)
};
class B {
  int n2;
```

```
public:
  void accept()
      cout << "enter no.2" ;
      cin >> n2;
    }
    friend int sum(A, B);
  };

int sum(A num1; B num2)
{ return num1.n1+num2.n2;
}
  int main()
  {
    A .obj1;
    B. obj2;
    obj1.accept();
    obj2.accept();
    cout << "sum;" <<sum(obj1, obj2) << endl;
    return 0;
  }
```

output:

enter num1: 3

enta num2 : 4

Sum: 7

T

i) Problem statement:- WAP with class number contains integer.

code:-

```
#include<iostream>
using namespace std;
class number {
int n;
public:
void accept()
{
cout << "enter a number";
cin >>n;
}
void disp ()
{ cout
cout << "\n number" <<n;
}
friend void swap (number num, number
}
void swap (number num2, number n2)
{
int temp = num1.n;
num1.n = num2.n;
num2.n = temp;
}
```

```
Int main () {

number n1;
 number n2;
 cout<< original values a: \n";
 num1. disp ();
 num2. disp ();
 swap (num1, num2);
 cout << " \n swapped values are: \n";
 num2. disp
 num1. disp
 return0
```

Output.

```
enter num: 5
enter num: 4
original values are
num is 5
num is 4
swapped values are
num is : 4
num is 5
```

3) WAP

Code :-

```
#include <iostream>
using namespace std;
access base;
class box
 {
```

```
void accept ()
{ cout << "enter volume of box"
  cin >> volume
 }
 friend void  findgreater (box vol1, cube
};

class cube
{
 int volume;
 public:-
 void accept ()
 {
   cout <<"enter vol of cube:";
   cin >> volume;
 }
 friend void  findgreater (boxvol1, cube vol2);
};
void findgreater &(boxvol1 , cubevol2)
 {
   if (vol1.volume> vol2.volume)
  { cout<< "enter volume of box is greater";
  }
 else
 {
   cout << "volume of cube is greater";
  }
 }
int main {
 box vol1;
 cube vol2;
 vol2 -accept
```

```
findgreater (vol1, vol-2);
return0;
}
```

output

```
enter volum of box :50
                cube : 77
volume of cube is greater
```

4)

```
Cou
# include <iostream>
 using namespace std;
 class complex
{
  int real;
  int imaginary ;
  public:
  void accept ()
  {
    cout cin >> real;
  }
  friend void add (complex n1, complex n2);

void add (complex n1, complex n2)
{
    int sum_real = n1.real + n2.real
    int sum_img = n1.img + n2.img
    cout << sum_real << sum_img
```

```
int main ()
{
    complex n1, n2;
    n1.accept ();
    n2.accept ();
    add (n1, n2);
    return 0
}
```

Output

```
enter real part : 6
enter img part : 8
                : 3
                : 6
final complex num : 9+14i
```

5)

Code

```
#include <iostream>
using name sp-
class student
{-
    string name;
    int n1, n2, m;
    public
      void accept ()
```

```
{
 cout << "\nenta name of student; ;
 cin >> nam;
 cout << "enta marks of first subject :
 cin >> m2;
 cout << "enter marks of third subject ;
 cin >> m3;
 }
};
void avg (student s1)

{
 int avg = (s1.m1 + s2.m2 + s1.m3)/3
 cout << avg is << av,

int main ()
{
 student s1
 s1.acept
 s2.ac
 avg (s1)
}
```

output

1 : 70
2 : 88
3 : 45
avg : 69

6)

Code

```
# include ~~stdio~~ <iostream>
using namespace std;
Class Beta;
Class Gamma
Class alpha
{
int n;
public
   void accep()
   { cout << enter >>
   cin >> n
   }
   freind void sum (alpha A, Beta B, Gamma C);
};
Class Beta
{
int n
void accept() {
   cin >> n;
}
freind void sum ( alpha A, Beta B, Gamma C);

Class gamma {
int n;
public
void accept
{
   cin >> n; }
```

```
friend void sum(Alpha A, Beta B, Gamma C)
};

void sum(alpha A, Beta B, Gamma C)
{
  int S=A.n+ B.n+C.n;
  Cout << "the sum is" << S;
}
int main()
{
alpha A;
Beta B;
Gamma C;
A.accept();
B.accept();
C.accept()
sum(A,B, C);
return 0;
}
```

Output-

enter val.

alpha : 5

Beta : 6

Gamma : 7

the sum is : 18

2)

Code :-

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

class point ()
  int x, y;
  public
    void accept()
    {
      cin >> x
      cin >> y;
    }
  friend void calculate (point p1, point p2);
};
  void calculate (point p1, point p2)
  {
    double distance = sqrt (p2.x - p1.x)
    (p2.y - p1.y (p2.y - p1.y));

int main ()
{
  point p1, p2;
  p1.accept ()
  p2.accept()
  calculate (p1.p2);
  return 0
}
```

Output
enter x val : 5
enter y val : 6
x : 3
y = 7
distance : 2.23607

8)
Code :

```
# include <iostream>
using namespace std;
class ~~aud~~ audit;
class bank acc
{
strings name;
int bank balance;
public
void accept ()
{
cin >> name
cin >> bank bal;
}
friend audit
};

class audit
public
void start Audit (Bank acc b1)
```

```
{
  cout<<"\n name" <<b1.name;
   cout<<" \n bank balance" <<bank balance;
  }
};
int main()
{
   bank b1;
   audit A1
    b1.accept();
    A1.start audit (b1);
   }
   return 0;
   }
```

Output-

enter name - Eisa

enter balance - 1000

name - shreyasi

bank balance - 1000

12/8

Experiment 5

1) default constructor

Code:

```
#include <iostream>
using namespace std;
class sum {
    int n, summ = 0, i = 1;
    public:
    sum() {
        n=5;
        for(i;i<=n;i++){
            summ summ+i;
        }
        cout<<summ;
    };

int main() {
    sum s1;

    return 0;
}
```

O/P:

The 15

Parameterized
Code :-

```
#include<iostream>
using namespace std;
Class sum {
    int n sumnum = 0; i=1
    public:
    sum(int num){
      n = number;
      for (i; i<=n; i++) {
        sumnum = sumnum + i;
    }
      cout << sumnum;
    }
};

int main () {
  sum s1 (5);
  return 0;
}
```

Copy

Code:-

```
#include<iostream>
using namespace std;
Class sum {
    int n, s;
public:
    sum() {
        n = 4
        s = 0;
    }
    sum(sum & obj){
        n = obj.n;
        s = obj.s;
    }
    void calc()
    {
        s=0
        for (int i=1 ; i<=n; i++)
        {
            s+=i
        }
    }
    void disp(){
        cout << "Number:" << n << endl;
        cout << "sum:" << s << endl;
    }
};
int main() {
    sum s1;
    s1.calc()
    s1.disp();
    sum s2(s1);
    s2.disp();
}
```

2)

```
#include<iostream>
using namespace std;
class student{
    int per;
    string name;
    public:
     student(){
       name="eiss";
       per =99;
     }
     void disp (){
        cout<<"the name is" <<name<<endl
        "the per is" << per;
     }
};
int main(){
    student s1;
    s1.display();
    return 0;
}
```

The name is eiss
the per is 99

Parameterized

1)
```
#include <iostream>
using namespace std;
class student {
    int per;
    string n;
    public:
    student (string name, int percentage){
        n=name;
        per=percentage;
    }
    void display (){
        cout << n<endl<<per<<;
    }

};

int main (){
    student.s1 ("eisi", 99);
    s1.display ();
    return 0;
}
```

Copy

```
#include<iostream>

using namespace std;
class student{
  int pea;
  string name;
  public:
    student(){
      name = "eias";
      pea=99;
  }
  student(student& s1){
      name=s1.nam;
      pea=s1.pea;
  }
  void display(){
    try
      cout << endl<< "the name is"<< nam
        << endl<<"the pea is "<< pea;
    }
  };
int main(){
  student s1;
  student s2(s1);
  s1.display();
  s2.display;
}
```

c) #include <iostream>
using namespace std;
class college
int roll;
string name, course,
public:
college (int r, string c="CSE")
n = name;
r = roll;
c = course
cout<<name<<r<<endl<<c;
};
};

int main(){
college C1("Bina", 44);
}

4. Problem statement: Define class "college" member variables as roll no, name, course, WAB using constructor with default value as computer engineering for the course

Code:
#include <iostream>
using namespace std;
class college
{
int roll;

```
string name, course;
public:
    college (int roll, string course="com
    {
      int r=roll;
      string n=name;
      string c=course
      cout<< r<<n<<c;
    }
};
int main()
{
    college c1(1,"Shreyash");
    return 0;
}
```

output

1. Eiss Company

4) constructor overloading

code:-

```
# include <iostream>
using namespace std;
class rectangle {
    int a, b;
    public:
    rectangle()
    {
        a=1
        b=2
    }
```

```
rectangle (int r)
{
    l=x;
    b=x;
}
rectangle( int x, int y)
{
    l=x;
    b=x;
}
void area()
{
cout<< "\nareas" << l*b;
}
};
int main()
{
  rectangle r;
  r.area();
  rectangle r2(2)
   r2.area()
   rectangle r3 (1,2)
   r3.area()
   return0;
}
```

Output

area is 2

area is 16

area is 2

16/9

Experiment 6:

1) Single Code:-

```
#include <iostream>
using namespace std;
class person{
    protected:
     int age;
     string name;
  }
class student: protected person{
      protected:
        int roll;
        public;
        public
        void    display() {
              cout <<"Enter name age n roll no
              cin >> name>> age >> roll;
              cout <<"the name is "<< name<< end
              "the age is <<age<< roll;
        };

int main(){
      student s;
        s.disp();
        return0;
  }
```

OP(

Write name age roll

Ein
M
4

2) multiple
Code ~~include~~ <
include

```
class academics {
        protected:
          int marks;
    };
class sports {
        protected:
          int score;
  };
class result: protected academics, protected sports
  public:
    void accept() {
      cout << "enter the marks score",
      cin >> marks >> score;
      cout << "the result is" << end << " marks"
      << marks << end << "scores" score;
    }
};
int main() {
result ri;
 ri.accept();
 return0;
}
```

?/ BP
Entered marks and scan 23

3
Hierarchy
marks-23
score 3

3) multilevel
Code

#

```
class vehicle {
    protected:
        string brand, model;
};
class car: protected vehicle {
    protected:
        string type;
};

class electric_car: protected car {
    private:
        int battery;
    public:
    void accept() {
        cout << "brand and model
        cin >> brand >> model
        cin >> type >> battery;
```

```
void disp() {
    cout << "brand: " << model << " type: " << [illegible]
}
};
int main() {
    electricPower;
    ev.accept();
    ev.disp();
    return 0
}
```

OID

person

reg

speed;

age

4) hierarchical

```
class Employee {
    public
    int empID;
    string name;
    void accept() {
        cout << "Enter Emp ID";
        cin >> EmpID;
        etc;
}
```

```
void show(){
        cout<< "EmpID: "<< EmpID<<endl;
        cout<< "Name" << name<< endl;
    }
};

class manager: public Employee{
    public
        string dpt;
        void accept (){
            Employee::accept()
        }

class Developer: public Employee{
    public:
        string pL;

        void accept (){
            cout << "Enter the prgr
            getline (cin, programming lang);
        }
        void show () {
            Employee ::sho
            cout
        }
};
```

```
int main(){
    Manager m,
    Developer,
    cout << \n Enter the Manager detail in
    m.accept
    d.accept
    m.show()
    d.show()
    return0;
}
```

:) Hybrid inheritance

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

class person {
        protected:
            String name;
            int age;
};

class student : protected person {
        protected:
            int marks;
};
class sports : protected student {
        protected:
            int score;
}
```

```
Class academics : protected student
        protected:
        int grade;
};

class result: protected sports {
        public
        void accept (){
          cin>>marks>>score;
        }
        void disp (){
          cout<<marks<<score;
        }
};
int main (){
        resultr;
        r.accept();
        r.disp();
        return0;
}
```

6) virtual base class

Code

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

class stude {
        protected;
        string name
};

class test : virtual protected stud {
        protected:
        int grade;
};

class result : virtual protected test, virtual protected
sports {
        int res;
        public
        void accept () {
            cin >> name >> per >> grade;
        }
        void disp () {
            res = (per + grade)/2;
            cout << res;
        }
};
```

→ PTO

```
int main (){
    result r;
    r.accept();
    r.disp();
    return 0;
}
```

Q/23'19

enter int

2
3
4
5
6
7
8
9
0

44

enter flow 1.5
1.2

Expt 7

1)

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

class calculate {
    int l, b;
    float h;
    public:
    void area:
     void area (int l, int b)
        cout<<l*b<<endl;
    }
     void area (float h) {
        cout<<h*h;
    }
};

int main () {
    Calculate c;
    c.area(2,2);
    c.area(1.5);
    return0;
}
```

2) Write a program <iostream>
using namespace std;

class summary {
int num, sum = 0,
float fnum, fsum = 0.0,
public
void sumint (int){
for (int i = 0; i<10; i++) {
cout << "enter the integer";
cin>>num;
sum = sum + num
}
void sumint (float){
for (int i=0; i<10; i++){
cout << "enter the float";
cin >> fnum;
fsum = fsum + fnum;
}
}
int main() {
summary s;
s.sumint(0);
s.sumint(0.0f);
return 0;
}

3)
```
# include <iostream>
using namespace std

class eisa {
    public
      int A;
      void operator -()
      {
        A = -A;
      }
    };

int main() {
      eisa B;
      B.A = 25;
      cout<<" Before:" << B.A << endl
        B;
      cout << "After:" << B.A<<endl
      return 0;
    }
```

4) #include <iostream>
using namespace std;

```
class addition
{
    public:
        int A;
        void operator ++()
        {
            A++;
        }
    }

int main()
{
    addition B;
    B.A=23;
    cout << "Befor" << B.A << endl
    ++B
    cout << "After" << B.A << endl
    return 0
}
```

17/10

Expt #8

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

   class strings {
       char str[50];
       public:
       void acc (){
           cout << "enter";
           cin >> st;
       }
       void operator + (strings s2)
       {
           strcat (this -> str, s2.str);
           cout << this -> str;
       }
   };

   int main (){
       strings s1, s2;
       s1.acc();
       s2.acc();
       s1 + s2;
       return 0;
   }
```

Output:

Enter : h

o

KO

2) Problem statement : write a program to create a base class Ilogin have datamembers name and password. Declare accept() function virtual. Derive email login and membership login classes from

Code :-

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

class Ilogin {
    protected
     string name, password;
     public:
     virtual void accept()
        cout << "enter password: ";
        cin >> password;
        }
     virtual void display () {
          cout << "name:" <<
          name << "\n password" << password <<
          endl
}
}
```

[illegible]

```
};
class membershiplogin:public login{
        string membershipID;
        public:
        void accept (){
        login::accept ();
        cout << "Enter membership ID;".
        (in >> membershipID
        }
        void display {
        cout << "membershiplogin:\n
        name << password;
}
};
```

```
int main(){
    login* ptr;
    email login e
    membership login m;
    ptr = &e; ~~ptr -> accept~~
    ptr -> accept();
    ptr = &m;
    ptr -> accept

    ptr = &e;
    ptr -> display
    ptr = &m;
    ptr -> display
    return 0;
}
```

O/P:

```
Enter name: abc
Enter password: 123
Enter email: elisa
enter name: sh
enter password 9
enter membership id 4
```

Email login	Membership login
abc	sh
123	9
elisa	4

Expt 1

1) Copy contents from another file

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

int main(){
    fstream first_file, second file;
    first file.open("first.txt", ios::in);
    if (!first_file){
        cout<<"Error opening first.txt!";
        return1;
    }
    char ch;
    while (first_file.get(ch)){
        second_file.put(ch);
    }
    first_file.close();
    second file.close();
    cout<<"file copied";
```

17/10

```
// count digits and spaces

#include <iostream>
#include <fstream>
#include <cctype.h>
using namespace std;

int main()
{
    fstream newfile;
    newfile.open("first.txt", ios::in);
    if (!newfile){
        cout << "Error opening first file";
        return;
    }
    int dig_c=0;
    int spc_c=0;
    char ch;

    while (newfile.get(ch)){
        if (isdigit(ch))
            dig_c++;
    }
}
    newfile.close();
    cout<<"No-of digits";
```

3)

Count words

Code:-

```
#include <iostream>
#include <fstream>
#include <ctype.h>
using namespace std

int main()
{
    fstream new_file;
    new_file.open("first.txt", ios::in);
    if (!newfile) {
        cout << "Error occurred!" << endl;
        return 1;
    }
    char ch;
    int word_c = 0;
    bool inwords = false;
    while (new_file.get(ch)){
        if (isspace(ch)){
            word_c++;
            inword = true
        }
    }
    new_file.close()
    cout << "no. of words:" << word_c << endl
    return 0;
}
```

Expt 10

1)

```
# include <iostream>
using namespace std;
template <typename t> int sumarr(t arr[]){
    int i;
    t sum=0
    for (i=0; i<10 ; i++){
        sum= sum+arr[i];
    }
    Cout <<sum;
    return0;
}
int main (){
int int arr[10]= {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
float floatarr[10]={1.1, 2.2, 3.3, 4.4, 5.5, 6.6, 7.7, 8.8, 9.9, 10.10};

int int sum= sumarr(intarr);
float fsum=sumarr(floatarr
double dsum = sumarr(doublearr);
return0;
}
```

A)

```
# include <iostream>
# include <fstream>
# include <string>
using namespace std;
int main()
{
    fstream new_file;
    new_file.open("first.txt", ios::in);
    if (!new_file){
        cout "Error in opening first.txt!";
        return 1;
    }
    string target = "Marcus"
    String word;
    int count = 0
    while (new_file>> word){
        if (word == target){
            count++;
        }
    }
    new_file.close()
    cout << "the word." << target << "occurs
    << count << "time" << endl;
    return 0
}
```

```
void accept()
{
    cout "enter number";
    cin >> num1 >> num2;
}
void add()
{
    cout << "sum" << num1 + num2;
}
void sub()
{
    cout << "sub" << num1 - num2;
}
void mul()
{
    cout << "mul" << num1 * num2;
}
void div()
{
    if (num2!=0)
    {
        cout "div" << num1 / num2;
    }
    else
    {
        cout << "dont divide by 0";
    }
}
};
```

2)

```
# include<iostream>
using namespace std;
template<typename t> tsquare (t val){
    t val1=val;
    return val* val;
}
temptate <<string square(string val){
    string val1 = val;
    return val + val1;
}
int main(){
    int intNum=5
    string str= "Heelhllo";
    cout << "square of in:" << square(int Num)<<en
    cout << "square of strin:" << square(str
    return0;
}
```

3) Simple Calculator using classtemp

Code:

```
# include <iostream>
using namespace std;
template <class T>
class C
{
    T num1, num2;
    public:
        C() {num1=0; num2=0;}
```

Exp 15

a) Implement stack

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
  stack<char> cars;
  Cars.push("bmw");
  car.push("Audi");
  car.push("Mercedes");
  Cars.push("Mercedes")
  Cars.push("Ferrari");
  cout<<"Top" <<cars.top();
  cout<<"Size"<<cars.size();
  cars.pop()
  Cars.pop();
  while (!cars.empty())
  {
    cout <<"element is stack are<<
    " cars.top();
    Cars.pop();
  }
  return 0;
}
```

O/P:

Top Ferrari

Size 4

Audi, BWM

```
cout << " modifies contents";
for (i=0 ; i<v.size() ; i++)
{
    cout << v[i] << " ";
}
return0;
}
O/P;
Current elements ~~put~~ in the vector are:

a b c d e f g h i ;

Expanding vector ;

A B C D E F G H I J
```

2) without iterator:

```
#include <iostream>
#include <bits/stdc++.h>
using namespace std;
int main()
{
    vector <char> v(10);
    unsigned int i;
    cout << "size=" << v.size();
    for (i=0; K 10; i++)
    {
        v[i] = i+a;
    }
    cout << " current content: "
    for (i=0; i< v.size(); i++) {
        cout << v[i] << " ";
    }
    cout << "\n\n";
    cout << " expanding vector.";
    for (i=0; i<10; i++)
    {
     v.push_back (i+10+'a');
    }
    cout << "size now:" << v.size();
    cout << "current contents: \n";
    for (i=0; i<v.size(); i++)
    {
        v[i] = toupper [v[i]];
    }
```

Expt 11

1) Accessing vector using iterator

```
#include <iostream>
#include <bits/stdc++.h>
using namespace std;
int main()
{
 vector<char> (10);
vector <char>: \iterator p;
int i;
p=v.begin ();
i = 0;
while (p != v.end()) {
 *p = i + a;
   p++;
   i++}
cout << "original content :\n";
 p = v.begin ();
while (p != v.end()) {
        cout<<p<<" ";
        p++;
   }
}
```

O/P

Vector contents

a b c d e f g h i j

```
void display ()
    if (top == -1) {
        cout << "stack is empty";
    } else
    {
        cout << "stack is empty";
    } else {
        cout << "stack elements";
        for (int i = top; i >= 0; i--)
        {
            cout << stack [i] << " ";
        }
    }
};

int main () {
    Stack <int> s;
    s.push(10);
    s.push(20);
    s.push(30);
    s.display();
    s.pop();
    s.display();
    return 0;
}
```

top ptr

10 pushed

20 pushed

30 pushed

10, 20, 30

popped

10, 20

4) Push pop using class template

```
#include <iostream>
using namespace std;
template <class T>
class stack {
    T stack[100];
    int top;
public
    stack() {
        top=-1
    }

    void push (T value){
        if (top=100) {
            cout << "overflow";
    } else
    {
        top++;
        stack [top]= value;
        cout <<val<< "pushed into stack";
    }
}
void pop ){
    if (top==-1){
        cout<< "underflow";
    } else
    {
        cout stack [top]<< "popped from stack";
    }
}
```

```
if (num2 != 0)
{
cout << "div" << num1/num2;
}
else
{
cout << "dont divide by 0"
}
};
int main () {
C <int>obj;
obj.accept ();
obj.add ();
obj.sub()
obj.mul();
obj.div();
return0;
}
```

O/P:

Enter num : 5
6
sum : 11
sub : -1
mul : 30
div : 0

b) Implement queue

```
#include <bits/stdc++.h>
using namespace std;
int main() {
queue <int> age;
age.push(20);
age.push(22);
age.push(23);
age.push(24);
cout << "Front:" << age.front();
cout << "Rear" << age.back();
age.pop();
age.pop();
while (age.empty())
{
cout << "element in the queue are:" << age.front();
age.pop();
}
return 0;
}
```

O/P:

Front element is : 21

Rear element : 20

Element in the queue are 23, 24