

Experiment - I

- 1) WAP to declare a class student having data members as rollno, name. Accept & display for one object
- 2) WAP to declare a class Book having data members as bid, bname, bprice. Accept data for 2 books & display name of book having greater price.
- 3) WAP to declare a class time. Accept time in HH:MM:SS & display total time in seconds.

C - Problems

1) Add 2 no.s

```
- #include <stdio.h>
int main () {
    int a, b, sum;
    printf ("Enter two numbers:");
    scanf ("%d %d", &a, &b);
    sum = a + b;
    printf ("sum = %d \n", sum);
    return 0;
}
```

2) Arithmetic operations using switch

```

#include <stdio.h>
int main () {
    int a, b;
    char op;
    printf ("Enter two numbers : ");
    scanf ("%d %d", &a, &b);
    printf ("Enter operator (+, -, *, /):");
    scanf ("%c", &op);
    switch (op) {
        case '+': printf ("Result = %d\n", a+b); break;
        case '-': printf ("Result = %d\n", a-b); break;
        case '*': printf ("Result = %d\n", a*b); break;
        case '/':
            if (b != 0)
                printf ("Result = %d\n", a/b);
            else
                printf ("Division by zero error!\n");
            break;
        default: printf ("Invalid operator\n");
    }
    return 0;
}

```

3)

check even or odd

```
#include <stdio.h>
int main() {
    int num;
    printf("enter a number :");
    scanf("%d", &num);
    if (num % 2 == 0)
        printf("%d is even\n", num);
    else
        printf("%d is odd\n", num);
    return 0;
}
```

4) Print 1 to 10 using for loop

```
- #include <stdio.h>
int main () {
    for (int i = 1; i <= 10; i++) {
        printf("%d", i);
    }
    printf("\n");
    return 0;
}
```

5) Print 2 to 10 using while loop

```
- #include <stdio.h>
int main () {
    int i = 1;
    while (i <= 10) {
        printf("%d", i);
        i++;
    }
    printf("\n");
    return 0;
}
```

c) a) -

```
#include <stdio.h>
int main () {
    int i, j, space;
    for (i = 1; i <= 3; i++) {
        for (space = 3; space > i;
             space--) {
            printf(" ");
        }
        for (j = 1; j <= 2 * i - 1; j++)
        {
            printf("*");
        }
        printf("\n");
    }
    return 0;
}
```

b) -

```
#include <stdio.h>
int main () {
    int i, j;
    for (i = 1; i <= 5; i++) {
        for (j = 1; j <= i; j++)
        {
            printf("%d", j);
        }
        printf("\n");
    }
    return 0;
}
```

c) -

```
#include <stdio.h>
int main() {
    int i, j;
    for (i = 1; i <= 5; i++) {
        for (j = 1; j <= i; j++) {
            printf ("%d", i);
        }
        printf ("\n");
    }
    return 0;
}
```

Ques
1318

11/7/25

Experiment - 1

1)

```
#include <iostream>
```

```
using namespace std;
```

```
class student {
```

```
private:
```

```
int roll no;
```

```
string name;
```

```
public:
```

```
void accept () {
```

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

```
cin >> roll no;
```

~~cout << "Enter name:";~~

~~cin >> name;~~

```
void display () {
```

```
cout << "Roll number:" <<
```

```
roll no << endl;
```

```
cout << "Name:" << name <<
```

```
endl;
```

```
}
```

```
};
```

```
int main () {
```

```
student s1;
```

```
s1.accept ();
```

```
s1.display ();
```

```
return 0;
```

}

2)

```
#include <iostream>
```

```
using namespace std;
```

```
class Book {
```

```
private:
```

```
int bid;
```

```
string bname;
```

```
float bprice;
```

```
public:
```

```
Void accept () {
```

```
cout << "Enter book ID:";
```

```
cin >> bid;
```

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

```
cin >> bname;
```

```
cout << "Enter book price:";
```

```
cin >> bprice;
```

```
}
```

```
float getPrice () {
```

```
return bprice;
```

```
}
```

```
string getName () {
```

```
return bname;
```

```
}
```

```
};
```

```
int main () {
```

```
book b1, b2;
```

```
cout << "Enter details for Book 1 : \n";
```

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

```
cout << "Enter details for Book 2 : \n";
```

```
b2.accept ();
```

```
cout << "\n Books with higher price : ";
```

```
if ( b1.getprice() > b2.getprice() ) {  
    cout << b1.getname() << endl;  
} else if ( b2.getprice() >  
            b1.getprice() ) {  
    cout << b2.getname() << endl;  
} else {  
    cout << "Both books have the  
same price" << endl;  
}  
f () {  
    return 0;  
}
```

3) ~~#include <iostream>~~

```
#include <iostream>  
using namespace std;  
class Time {  
private:  
    int hours, minutes, seconds;  
public:  
    void accept() {  
        cout << "Enter time in hh mm ss  
format :";  
        cin >> hours >> minutes >> seconds;  
    }  
    void displaytotalseconds() {  
        int totalseconds = hours * 3600  
        + minutes * 60 + seconds;  
        cout << "Total time in seconds :"  
        << totalseconds << endl;  
    }  
};
```

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

Qn
13/8

6/7/25

Experiment - 2

1)

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

int main() {
    city cities[5];
    for (int i = 0; i < 5; i++) {
        cout << "Enter name of city" <<
        i + 1 << endl;
        cin >> cities[i].name;
        cout << "Enter population";
        cin >> cities[i].population;
    }

    int maxindex = 0;
    for (int i = 1; i < 5; i++) {
        if (cities[i].population >
            cities[maxindex].population) {
            maxindex = i;
        }
    }

    cout << "The city with highest
population" << cities[maxindex].name << endl;
    return 0;
}
```

2)

```
#include <iostream.h>
using namespace std;
class Account {
public:
    int accno;
    float balance;
};

int main () {
    Account acc[10];
    for (int i = 0; i < 10; i++) {
        cout << "Enter account number"
        << i + 1 << endl;
        cin >> acc[i].accno;
        cout << "Enter balance";
        cin >> acc[i].balance;
        if (acc[i].balance >= 5000) {
            acc[i].balance += acc[i].balance * 0.10;
        }
    }
    cout << "\n Accounts with
updated balance >= 5000 \n";
    for (int i = 0; i < 10; i++) {
        if (acc[i].balance >= 5000) {
            cout << "Account no " <<
            acc[i].accno << ", Balance" <<
            acc[i].balance << endl;
        }
    }
    return 0;
}
```

(3)

```
#include <iostream>
using namespace std;
class staff {
public:
    string name;
    string post;
};

int main () {
    staff staff[5];
    for (int i=0; i<5; i++) {
        cout << "Enter name of staff" " ";
        << i+1 << endl;
        cin >> staff[i].name;
        cout << "Enter post" " ";
        cin >> staff[i].post;
    }

    cout << "In staff members who are
HOD \n" ;
    for (int i=0; i<5; i++) {
        if (staff[i].post == "HOD") {
            cout << staff[i].name <<
endl;
        }
    }
    return 0;
}
```

Experiment - 3

1)

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

int main () {
    Book b;
    Book *ptr;
    ptr = &b;
    ptr->accept ();
    ptr->display ();
    return 0;
}
```

2)

```

- #include <iostream>
    using namespace std;
    class student {
        public:
            int roll_no;
            float percentage;
        void accept () {
            cout << "Enter roll number ";
            cin >> this->roll_no;
            cout << "Enter percentage ";
            cin >> this->percentage;
        }
        void disp () {
            cout << "In Roll number "
                << this->roll_no;
            cout << "In percentage "
                << this->percentage;
        }
    };
    int main () {
        student s;
        s.accept ();
        s.display ();
        return 0;
    }

```

3)

```

- #include <iostream>
    using namespace std;
    class outer {
        public:

```

```
class outer inner {
```

```
public:
```

```
void show () {
```

```
cout << "This is a nested class "
```

```
<< endl;
```

```
}
```

```
};
```

```
}
```

```
int main () {
```

```
outer :: inner obj ;
```

```
obj . show () ;
```

```
return 0 ;
```

```
}
```

Qn
13/8

Experiment - 4

```
#include <iostream>
using namespace std;
class number {
    int a;
public:
    void getdata() {
        cout << "Enter number";
        cin >> a;
    }
    void display() {
        cout << " number " << a;
    }
    void swap(number &n) {
        int temp = a;
        a = n.a;
        n.a = temp;
    }
};

int main() {
    number n1, n2;
}

cout << "Enter first number";
n1.getdata();
cout << "Enter second number";
n2.getdata();
cout << "Before swapping";
n1.display();
n2.display();
```

```
n1.swap(n2);  
cout << "\n after swapping";  
n1.display();  
n2.display();  
return 0;  
}
```

2)

```
#include <iostream>
```

```
using namespace std;
```

```
class number {
```

```
int q;
```

```
public:
```

```
void getdata () {
```

```
cout << "enter number";
```

```
<in >> q;
```

```
}
```

```
void display () {
```

```
cout << "number" << q << endl;
```

```
}
```

```
friend void swapnumbers (number &n1, number &n2) {
```

```
int temp = n1.q;
```

```
n1.q = n2.q;
```

```
n2.q = temp;
```

```
}
```

```
int main () {
```

```
number n1, n2;
```

```
cout << "Enter first number\n";
```

```
n1.getdata();
```

```
cout << " Enter second number\n";
```

```
n2.getdata();
```

```

cout << "In Before swapping ";
n1.display();
n2.display();
swapnumbers( n1, n2 );
cout << "In After swapping ";
n1.display();
n2.display();
return 0;
}

```

③

```

#include <iostream>
using namespace std;
class class1;
class class2 {
    int b;
public:
    void getdata() {
        cout << "Enter number for "
class2"; cin >> b;
    }
    void display() {
        cout << "class2 number " << b << endl;
    }
friend void swapnumbers( class2 &, class2 & );
class class1 {
    int q;
public:
    void getdata() {
        cout << "Enter number for class1";
        cin >> q;
    }
}

```

```

} // class result

void display () {
    cout << "class number" << a << endl;
}

friend void swapnumbers (class1 &c1, class2 &c2);
};

void swapnumbers (class1 &c1, class2 &c2)
{
    int temp = c1.a;
    c1.a = c2.b;
    c2.b = temp;
}

int main ()
{
    class1 obj1;
    class2 obj2;

    obj1.getdata ();
    obj2.getdata ();

    cout << "In Before swapping" ;
    obj1.display ();
    obj2.display ();

    swapnumbers (obj1, obj2);

    cout << "In after swapping" ;
    obj1.display ();
    obj2.display ();
    return 0;
}

```

4)

```
#include <iostream>
```

```
using namespace std;
```

```
class result2;
```

```
class result {
```

```
int marks ;  
public :  
    void getmarks () {  
        cout << "Enter marks for  
        Result 1" ;  
        cin >> marks ;  
    }  
friend float average (Result1 , Result2 ) ;  
};  
class Result2 {  
    int marks ;  
public :  
    void getmarks () {  
        cout << "Enter marks for Result2" ;  
        cin >> marks ;  
    }  
friend float average (Result1 , Result2 ) ;  
};  
float average ( Result1 r1 , Result2 r2 ) {  
    return ( r1 . marks + r2 . marks )  
        / 2.0 ;  
}  
int main () {  
    Result1 obj1 ;  
    Result2 obj2 ;  
    obj1 . getmarks () ;  
    obj2 . getmarks () ;  
    cout << "Average marks" << average (obj1 , obj2 )  
        << endl ;  
    return 0 ;  
}
```

5)

```
#include <iostream>
using namespace std;
class B;
class A {
    int num;
public:
    void get() { cin >> num; }
    friend void greatest (A, B);
};

class B {
    int num;
public:
    void get() { cin >> num; }
    friend void greatest (A, B);
};

void greatest (Ax, By) {
    cout << "Greatest"
    << (x.num > y.num ? x.num : y.num);
}

int main () {
    A a; B b;
    cout << "Enter number for class A";
    a.get();
    cout << "Enter number for class B";
    b.get();
    greatest (a, b);
    return 0;
}
```

friend function questions

1) - #include <iostream>
using namespace std;
class classB;
class class A {
 int a;
public:
 class A (int val): a(val) {}
 friend int sum (class A, class B);
};
class class B {
 int b;
public:
 class B (int val): b(val) {}
 friend int sum (class A, class B);
};
int sum(class A objA, class B objB) {
 return objA.a + objB.b;
}
int main () {
 class A a(10);
 class B b(20);
 cout << "sum = " << sum(a, b) << endl;
 return 0;
}

2) - #include <iostream>

using namespace std;
class number {
 int value;
public:

```

Number (int val) : value (val) {}
    void display () { cout << value
        << endl; }

    friend void swapnumbers (number &n1, number &n2);
}

void swapnumbers (number &n1, number &n2) {
    int temp = n1.value;
    n1.value = n2.value;
    n2.value = temp;
}

int main () {
    number n1(5), n2(15);
    cout << "Before swap = " ; n1.display(); n2.display();
    swapnumbers (n1, n2);
    cout << "After swap = " ; n1.display();
    n2.display();

    return 0;
}

```

3) - ~~#include <iostream>~~

```

using namespace std;
class cube;
class Box {
    int volume;

public:
    Box (int v) : volume (v) {}

    friend void findgretter (Box, cube);
}

class cube {

```

```

int volume;
public:
    cube (int v) : volume (v) {}
friend void findgreater (Box b, cube c);
}
void findgreater (Box b, cube c) {
cout << "Greater Volume = " << cb.volume >>
volume; b.volume : (.volume) << endl;
}

int main () {
Box Box (218);
cube cube (90);
findgreater (box, cube);
return 0;
}

```

4) - #include <iostream>

```

using namespace std;
class complex {
    int real, img;
public:
    complex (int r=0, int i=0) : real (r),
    img (i) {} void display () {
        cout << real << " + " << img <<
        "i" << endl; }
    friend complex add (complex, complex);
}
complex add (complex c1, complex (z)) {
    return complex (c1.real + c2.real,
    c1.img + c2.img); }
int main ()

```

```

complex c1 (+,5), c2 (2,3);
complex result = add (c1,c2);
cout << "sum of complex numbers:";
result . disp();
return 0;
}

```

5) -

```

#include <iostream>
using namespace std;
int m1, m2, m3;
public:
student (string n, int a, int b,
int c) : name(n), m1(a), m2(b), m3(c) {}
friend void calculateaverage (student);
}
void calculateaverage (student s) {
float avg = (s.m1 + s.m2 +
s.m3) / 3;
cout << "Average marks of " << s.name <<
" = " << avg << endl;
}
int main () {
student s ("Swami", 90, 93, 91);
calculateaverage (s);
return 0;
}

```

6) -

```

#include <iostream>
using namespace std;
class Beta; class Gamma;
class Alpha {
int q;

```



public :

Alpha (int val) : a (val) {}

friend void printsum (Alpha, Beta,
Gamma) ;

}

class Gamma {

int c ;

Public :

Gamma (int val) : c (val) {}

friend void printsum (Alpha, Beta, Gamma) ;

void printsum (Alpha x, Beta y,
Gamma z) {

cout << "Sum = " << x.a + y.b +
z.c << endl ;

}

int main () {

Alpha a (10) ; Beta b (20) ; Gamma c (30)

printsum (a, b, c) ;

return 0 ;

7)

include <iostream>

include <math>

using namespace std ;

class point {

int x, y ;

Public :

point (int a, int b) : x (a), y (b) {}

```

friend double distance (point, point) {
} ;

double distance (point p1, point p2) {
    return sqrt (pow (p2.x - p1.x,
+ pow (p2.y - p1.y, 2));
}

int main () {
    point p1 (0, 0), p2 (3, 4);
    cout << "Distance = " << distance
(p1, p2) << endl;
    return 0;
}

```

8) - #include <iostream>

```

using namespace std;
class BankAcc {
    double balance;

public:
    BankAccount (double b) : balance
(b) {}

    friend class Audit;
};

class Audit {
public:
    void printBalance (BankAcc acc) {
        cout << "Audited Balance = " << acc.balance
        << endl;
    }
};

int main () {
    BankAcc acc (5000.75);
}

```

Audit Audit ;

Audit - printBalance (acc);
return 0;
}

~~Qn~~
~~26/8~~

Experiment - 5

1)

```
#include <iostream>
using namespace std;
class Sumcalculator {
private:
    int n;
    int sum;
public:
    Sumcalculator(int num) {
        n = num;
        sum = 0;
    }
    for (int i = 1; i <= n; ++i) {
        sum += i;
    }
    void displaysum() {
        cout << "Sum of numbers from 1 to "
        << n << " is: " << sum << endl;
    }
};

int main() {
    int number;
    cout << "Enter a number: ";
    cin >> number;
    Sumcalculator sc(number);
    sc.displaysum();
    return 0;
}
```

(2)

```
#include <iostream>
#include <string>
using namespace std;
class student {
private:
    string name;
    float Percentage;
public:
    student(string n, float p)
        name = n;
        Percentage = p;
    void display() {
        cout << "student name: " << name << endl;
        cout << "Percentage: " << Percentage
            << "%" << endl;
    }
};

int main() {
    string studentname;
    float studentpercentage;
    cout << "enter student name: ";
    getline(cin, studentname);
    cout << "Enter student percentage: ";
    cin >> studentpercentage;
    student s(studentname, studentpercentage);
    s.display();
    return 0;
}
```

(3)

```
#include <iostream>
#include <string>
using namespace std;
class college {
private:
    int rollno;
    string name;
    string course;
public:
    college (int r, string n,
              string c) {
        rollno = r;
        name = n;
        course = c;
    }
    void display () {
        cout << "In rollno :" << rollno << endl;
        cout << "Name : " << name << endl;
        cout << "course : " << course << endl;
    }
};

int main () {
    int roll1, roll2;
    string name1, name2, course2;
    cout << "enter roll number for
student 1 ";
    cin >> roll1;
    cin.ignore();
    cout << "enter name for student 1 ";
    getline (cin, name1);
```

college student1 (roll1, name1);
cout << "In student details";
student1 . display ();
return 0;
}

④

=

```
#include <iostream>
#include <string>
using namespace std;
class student {
private:
    int rollno;
    string name;
    float percentage;
public:
    student () {
        rollno = 0;
        name = "Unknown";
        percentage = 0.0;
    }
```

~~Void display () {~~
~~cout << "In rollno " << rollno << endl;~~
~~cout << "Name " << name << endl;~~
~~cout << "Percentage " << percentage << "%"
<< endl;~~
};
};

Ques
(2)(1)

```
int main () {
    student s1 ( 111, "xyz");
    cout << "student 1";
    s1 . display ();
    return 0;
}
```

Experiment - 6

1) hierarchical inheritance

=

```
#include <iostream>
using namespace std;
class employee {
protected:
    int empID;
    string name;
public:
    Employee (int id, string n)
    : empID(id), name(n) {}
    void display () {
        cout << "ID:" << empID <<
        ", Name " << name << endl;
    }
};

class manager : public employee {
string dept;
public:
    Manager (int id, string n, string d)
    : Employee (id, n), dept(d) {}
    void display () {
        employee::display ();
        cout << "Dept:" << dept << endl;
    }
};

class Developer : public employee {
string lang;
public:
    Developer (int id, string n, string l)
    : employee (id, n), lang(l) {}
```

```

void display () {
    employee :: display ();
    cout << "Lang " << Lang << endl;
}
};

int main () {
    manager m (101, "Swarni",
    "goonology");
    developer d (102, "Arnav",
    "roting");
    m.display ();
    d.display ();
    return 0;
}

```

2)

multilevel inheritance

```

#include <iostream>
#include <string>
using namespace std;
class vehicle {
protected:
    string brand;
    string model;
public:
    vehicle (string b, string m) : brand(b), model(m) {
        void displayvehicle () {
            cout << "Brand: " << brand << endl;
            cout << "model: " << model << endl;
        }
    }
}

```

class car : public vehicle {

protected :

string type ;

public :

car(string b, string m, string t) :

Vehicle(b, m), type(t) {}

void display() {

displayVehicle();

cout << "Type:" << type << endl;

}

};

class electriccar : public car {

private :

int batterycapacity;

public :

Electriccar(string b, string m, string t, int bc)

: car(b, m, t), batterycapacity(bc) {}

void displayelectriccar() {

displaycar();

cout << "Battery capacity :" << batterycapacity

<< "kwh" << endl;

}

};

int main() {

Electriccar ec ("Tesla", "model s", "sedan",

100);

ec.displayelectriccar();

return 0;

}

3) multiple inheritance

- #include <iostream>

using namespace std;

class Academic {

protected:

float marks;

public:

Academic (float m) : marks(m) {}
void displayAcademic () {

cout << "Academic marks :" << marks
<< endl;

}

};

class sports {

protected:

float sportscore;

public:

sports (float s) : sportscore(s) {}

void displaySports () {

cout << "sports score :" << sportscore
<< endl;

}

};

class result : public Academic, public sports

{

public:

result (float m, float s) : Academic(m), sports(s) {}

void displayTotal () {

float total = marks + sportscore;

displayAcademic();

displaySports();

```

cout << "Total score : " << total << endl;
}
};

int main() {
    result r(85.5, 14.5);
    r.displaytotal();
    return 0;
}

```

4) hybrid

```

#include <iostream>
#include <string>
using namespace std;
class person {
protected:
    string name;
    int age;
public:
    person(string n, int a) : name(n), age(a) {}
    void displayperson() {
        cout << "Name : " << name << endl;
        cout << "Age : " << age << endl;
    }
};

class student : public person {
protected:
    int rollnumber;
public:
    student(string n, int a, int r) :
        person(n, a), rollno(r) {}
}

```

```
void displaystudent () {  
    displayperson ();  
    cout << "roll no: " << rollno << endl;  
}
```

```
};
```

```
class sports {
```

```
protected :
```

```
float sportsscore;
```

```
public :
```

```
sports (float s) : sportsscore (s) {}
```

```
void displaysports () {
```

```
cout << "sports score: " << sportsscore  
<< endl;
```

```
}
```

```
};
```

```
class Academics {
```

```
protected :
```

```
float academicmarks;
```

```
public :
```

```
Academics (float m) : academicmarks (m) {}
```

```
void displayAcademics () {
```

```
cout << "Academic marks: " << Academicmarks  
<< endl; }}
```

```
class Result : public student, public sports,
```

```
public Academics {
```

```
public :
```

```
result (string n, int a, int r, float m, float s)
```

```
: student (n, a, r), sports (s),
```

```
Academics (m) {}
```

```
void displayresult () {
```

```
displayStudent();  
displayAcademics();  
displaySports();  
cout << "Total score :" << academicMarks + sportScore  
<< endl;  
} } ;  
int main() {  
    student r("swaraj", 21, 1001, 88.5, 12.0);  
    r.displayResult();  
    return 0;  
}
```

Qn
12/11

Experiment - 7

1)

```
#include <iostream>
using namespace std;
Void area (int l , int b )
    cout << "Lab Area : " << l * b << endl;
Void area ( int side )
    cout << " Classroom Area : " << side * side
    << endl;
int main ()
    area ( 10 , 20 );
    area ( 15 );
    return 0;
}
```

2)

```
#include <iostream>
using namespace std;
Void sum ( float a , float b , float c , float d ,
float e ) {
    cout << " float sum : " << a + b + c + d + e << endl;
}

Void sum ( int a [ ] ) {
    int s = 0;
    for ( int i = 0 ; i < 10 ; i ++ ) s += a [ i ];
    cout << " Int sum : " << s << endl;
}

int main ()
    sum ( 1.1 , 2.2 , 3.3 , 4.4 , 5.5 );
    int arr [ 10 ] = { 1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 , 10 };
    sum ( arr );
```

```
    return 0;
}
```

(3)

```
- #include <iostream>
using namespace std;
class num {
    int x;
public:
    num(int a) : x(a) {}
    void operator -() {x = -x;}
    void show() {
        cout << "value :" << x << endl;
    }
    int main() {
        num n(5);
        -n;
        n.show();
        return 0;
    }
}
```

(4)

```
#include <iostream>
using namespace std;
class num {
    int x;
public:
    num(int a) : x(a) {}
    num operator ++ () { x++; return *this; }
    num operator ++ (int) { num t = *this; x++; return t; }
    void show()
```

cout << "Value:" << x << endl;

};

int main(){

num n(5);

++n; n.show();

n++; n.show();

return 0;

}

Q
12/11

Experiment - 8

1)

```

#include <iostream>
#include <string>
using namespace std;
class mystring {
    string str;
public:
    mystring(string s): str(s) {}
    mystring operator + (mystring)
    return mystring(str+s.str);
    void show()
    cout << str << endl;
};

int main()
{
    mystring a ("xyz"), b ("pqr");
    mystring c = a+b;
    c.show();
    return 0;
}

```

2)

```

#include <iostream>
using namespace std;
class Ilogin {
protected:
    string name, password;
public:
    virtual void accept()=0;
};

```

```
class emaillogin : public Ilogin {
public:
    void accept() {
        name = "email-user"; password = "email-pass";
        cout << "Email :" << name << ", password :"
            << password << endl;
    }
}

class membershiplogin : public Ilogin {
public:
    void accept() {
        name = "member-user", password = "member-pass";
        cout << "member :" << name << ", password :"
            << password << endl;
    }
}

int main() {
    Ilogin * login;
    emaillogin e; membershiplogin m;
    login = &e; login->accept();
    login = &m; login->accept();
    return 0;
}
```

Ques
12/11

Experiment - 9

1)

```
#include <iostream>
using namespace std;
int main () {
    ifstream fin ("first.txt");
    ofstream fout ("second.txt");
    char ch;
    while (fin.get(ch)) fout.put(ch);
    fin.close(); fout.close();
    return 0;
}
```

2)

```
#include <fstream>
#include <iostream>
using namespace std;
int main () {
    ifstream fin ("first.txt");
    char ch; int digits = 0, spaces = 0;
    while (fin.get(ch))
        if (isdigit(ch)) digits++;
        if (isspace(ch)) spaces++;
    cout << "Digits : " << digits << " , spaces : " << spaces
        << endl;
    fin.close ();
    return 0;
}
```

```

#include <iostream>
#include <iostream>
#include <string>
using namespace std;
int main() {
    ifstream fin ("first.txt");
    string word; int count = 0;
    while (fin >> word) count++;
    cout << "word count : " << count
        << endl;
    fin.close();
    return 0;
}

```

```

#include <iostream>
#include <iostream>
#include <string>
using namespace std;
int main() {
    ifstream fin ("first.txt");
    string word, target = "hello"; int count = 0;
    while (fin >> word) if (word == target)
        count++;
    cout << "occurrences of " << target << ": "
        << count << endl;
    fin.close();
    return 0;
}

```

Q1
12/11

Experiment - 10

1)

```
#include <iostream>
using namespace std;
template <typename T>
T sum(T a[], int n) {
    T s = 0;
    for (int i = 0; i < n; i++) s += a[i];
    return s;
}

int main() {
    int i[5] = {1, 2, 3, 4, 5};
    float f[5] = {1.1, 1.2, 1.3, 1.4, 1.5};
    double d[5] = {1, 2, 3, 4, 5};
    cout << sum(i, 5) << endl;
    cout << sum(f, 5) << endl;
    cout << sum(d, 5) << endl;
    return 0;
}
```

2)

```
#include <iostream>
#include <string>
using namespace std;
template <typename T>
T square(T x) { return x * x; }
template <>
string square(string s) { return s + s; }

int main() {
    cout << square(s) << endl;
    cout << square(string("Hi")) << endl;
}
```

```
return 0;  
}
```

3)

```
#include <iostream>  
using namespace std;  
template <typename T>  
class calc {  
    T a, b;  
public:  
    calc (T x, T y) : a(x), b(y) {}  
    void show () {  
        cout << "Add: " << a+b << "In sub:"  
            << a-b  
            << "In mul:" << a*b << "In Div" << a/b << endl;  
    }  
};  
int main () {  
    calc <int> c(10, 2);  
    c.show();  
    return 0;  
}
```

4)

```
#include <iostream>  
using namespace std;  
template <typename T>  
class stack {  
    T arr[10]; int top = -1;  
public:  
    void push (Tx) { if (top < 9)  
        arr[++top] = x; }
```

```
void pop()
if (top >= 0) top--;
void show() {
for (int i = 0; i <= top; i++) cout << arr[i]
<< " ";
cout << endl;
}
int main() {
stack < int > s;
s.push(1); s.push(2);
s.show();
s.pop();
s.show();
return 0;
}
```

Qn
12/11

Experiment - 11

1)

```
#include <iostream>
using namespace std;
template <typename T>
class vector {
    T arr[10]; int size;
public:
    void create (T arr[], int n) {
        size = n;
        for (int i = 0; i < n; i++) arr[i] = 0;
    }
    void modify (int index, T val) {
        if (index >= 0 && index < size) arr[index] = val;
    }
    void multiply (T scalar) {
        for (int i = 0; i < size; i++) arr[i] *= scalar;
    }
    void display () {
        cout << "(";
        for (int i = 0; i < size; i++) {
            cout << arr[i];
            if (i < size - 1) cout << ", ";
        }
        cout << ")" << endl;
    }
};

int main () {
    vector <int> v;
    int arr[5] = { 10, 20, 30, 40, 50 };
    v.create (arr, 5);
    v.display ();
}
```

```
V.create (data, s);  
V.modify (2, 99);  
V.multiply (2);  
V.display ();  
return 0;  
}
```

Experiment - 12

(1)

```
#include <iostream>
#include <stack>
using namespace std;
int main () {
    stack <int> s;
    s.push (10);
    s.push (20);
    s.push (30);
    cout << "Top element :" << s.top () << endl;
    s.pop ();
    cout << "Top element after pop" << s.top ()
        << endl;
    cout << "stack size" << s.size () << endl;
    if (s.empty ()) {
        cout << "stack is empty" << endl;
    } else {
        cout << "stack is not empty" << endl;
    }
    return 0;
}
```