

Practical - ①

★ 2) Write a C++ program to perform arithmetic operators using switch case.

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

```
int main () {
    float a, b;
    char op;
```

```
cin >>a >>op >>b;
```

```
switch (op) {
```

```
case '+': cout << a + b; break;
```

```
case '-': cout << a - b; break;
```

```
case '*': cout << a * b; break;
```

```
case '/': cout << a / b; break;
```

```
if (b != 0) << a / b;
```

```
else cout << "can't divide by zero";
```

```
break;
```

```
default: cout << "Invalid operator";
```

```
}
```

```
return 0;
```

```
}
```

3. C++ code to check whether a no. is odd or even.

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

int main () {
    int num;

    cout << "Enter a number: ";
    cin >> num;
    if (num % 2 == 0)
        cout << "Even" << endl;
    else
        cout << "Odd" << endl;
    return 0;
}
```

4 #include <iostream>
using namespace std;

int main() {
 for (int i=1; i<10; i++) {
 cout << i << endl;
 }
 return 0;
}

5 #include <iostream>
using namespace std;

```
int main() {  
    int i = 1  
  
    while (i <= 10) {  
        cout << i << endl;  
        i++;  
    }  
    return 0;  
}
```

6) C++ to code the below pattern

Q) *

* *

** *

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

```
int main() {
    int rows = 3;
```

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

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

}

~~```
 for (int k=1; k<= i; k++) {
```~~~~}~~~~```
        cout << endl;
```~~~~}~~~~```
 return 0;
```~~~~}~~

b) 1

2 2

3 3 3

4 4 4 4

5 5 5 5

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

```
int main() {
 for (int i = 1; i <= 5; i++) {
 for (int j = 1; j <= i; j++) {
 cout << endl;
 }
 }
 return 0;
}
```

Q 1  
12  
123  
1234  
12345

#include <iostream>  
using namespace std;

```
int main () {
 for (int i = 1; i <= 5; i++) {
 for (int j = 1; j <= i; j++) {
 cout << endl;
 }
 }
 return 0;
}
```

1) write C++ program for adding 2 no's 2 no. 5

```
#include <iostream>
```

```
void main ()
```

```
{
```

```
int a,b,c;
```

```
cout << "Enter values of a&b";
```

```
cin >> a >> b;
```

```
c=a+b;
```

~~```
cout << "addition = " << c;
```~~

3

~~Q
8/7/15~~

Practical - I

- a. Write a C++ program to declare class student & also display the data for one obj.

```
#include <iostream>
using namespace std;
class student ()
{
    string name, clg;
    int roll_no;
public:
    void accept()
    {
        cout << "enter name : ";
        getline (cin, name);
        cout << "enter roll no. ";
        cin >> roll_no;
        cout << "enter class : ";
        getline (cin, clg);
    }
    void display()
    {
        student s1;
        s1.accept();
        s1.display();
        return 0;
    }
}
```

Output: Enter name: IshacIn
Enter Roll No: 50
Enter class: SYAIDS

Enter Name: Ishaaan

Enter Roll no: 50

Enter class: SY AIDS

b) write a cpp program to declare a class book having data members as id, name & price accept data for 2 books & display data of book having greater price.

```
#include <iostream>
#include <string>
using namespace std;
class book {
public:
    string book_name;
    float price;

    void accept () {
        cout << "Enter book ID:" ;
        cin >> id;
        cout << "Enter book name:" ;
        cin >> name;
        cout << "Enter book price:" ;
        cin >> price
    }

    void display () {
        cout << "Book ID: " << id << endl;
        cout << "Book Name: " << name << endl;
        cout << "Book Price: " << price << endl;
    }

    float get_price () {
        return price;
    }
}
```

{3}

3;

int main () {

int n;

cout << "Enter number of books: ";

cin >> n;

Books book[n];

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

cout << "\nEnter details of book " << i + 1 << ":";

endl;

book[i].accept();

}

int maxIndex = 0;

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

if (book[i].getPrice() > book[maxIndex].getPrice())

getPrice();

maxIndex = i;

}

{3}

cout << "\nBook with the greatest price: " <<

endl;

books[maxIndex].display();

return 0;

{3}

Output:

Enter number of books: 2

Enter details of book 1

BOOK ID: 65 65

Book name: kRii

Book price: 987

Enter details of Book 2:

BOOK ID: 45

Book name: Harry Potter

Book price: 65

- 3) WAP to declare a class time having data members as H, M & S. Accept data for one object & display total time in seconds.

#include <iostream>
using namespace std;

class Time {

int H, M, S;

public

void accept {

cout << "Enter hours: "

cin >> H;

cout << "Enter seconds: "

cin >> S;

cout << "Enter minutes: "

cin >> M;

}

void display Total seconds () {
int totalseconds = h * 3600 + m * 60 + s,
cout << "Total Time in seconds: " << total
seconds
endl;

{

3.

```
int main () {  
    time t;  
    t. accept ();  
    t. display Total seconds ();  
    return 0;
```

{

Output:-

Enter hours: 4

Enter minutes: 23

Enter seconds: 54

Total time in seconds: 15834

Ques
121)

ASS-2

1. #include <iostream>
using namespace std;

class account {

private:

int acc_no;
int balance;

public:

void accept () {

cout << "Enter acc number:" endl;

cin >> acc_no;

cout << "Enter balance:" endl;

cin >> balance;

}

void add_int (account a []) {

cout << "New balance after 10% interest
<< endl;

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

if (a[i].balance >= 5000) {

a[i].balance = a[i].balance

+ (a[i].balance * 10 / 100);

cout << "Acc. Nym. = " << a[i].acc_no;

cout << "balance = " << a[i].balance << endl;

}

3

3

```
int main () {  
    account a[10];  
    for (int i=0; i<10; i++) {  
        cout << "Enter details for acc: " + i + 1;  
        a[i].accept();  
    }  
    arr[0].add_int(a);  
    return 0;  
}
```

O/P :

Enter details for account 1
enter account number: 12
Enter balance: 345

Enter details for account 2:
Enter account number: 15
Enter balance: 870

Enter details for account 3:
~~Enter account number: 45~~
~~Enter balance: 87~~

Enter details for account 4:
Enter account number: 67
Enter balance: 965

Enter details for accounts:
Enter account number: 83
Enter balance: 365

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

```
class city
```

```
{
```

```
public :
```

```
string name;
```

```
int population;
```

```
void accept ()
```

```
{
```

```
<< "city name" >> name << endl;
```

```
<< "population" >> population
```

```
3
```

```
void display ()
```

```
{
```

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

```
cout << "Population: " << population << endl;
```

```
3
```

```
3;
```

```
int main()
```

```
{
```

```
vector<city> cities (5)
```

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

```
{
```

```
cout << "\nEnter data for city " << i + 1 << endl;
```

```
cities [i].accept ();
```

```
3
```

city highest popularity city=cities [0]
for i in range(1, len(cities)):
 if cities[i].population > highest_population:
 highest_population = cities[i].population
 city = cities[i]

3

cout << "The city with highest population
(in miles) is: ";

return 0;

3

Output

Enter 1: Data for 5 cities

City 1:

Enter city name & population: ABC 100

City 2:

Enter city name & population: DEF 250

~~City 3:~~

Enter city name & population: GHI 60

City 4:

Enter city name & population: JKL 420

3. #include <iostream>
#include <string>
using namespace std;
class Staff {
 string name;
 string post;
public:

void accept() {
 cout << "Enter name : ";
 cin >> name;
 cout << "Enter post : ";
 cin >> post;
}

void display() {
 if (post == "HOD") {
 cout << "Name " << name << endl;
 }
}

int main() {
 Staff S[5];
 for (int i = 0; i < 5; i++) {
 cout << "Enter details for Staff " << i + 1;
 S[i].accept();
 }
 cout << "In Staff with post HOD (" <<

```

for (int i = 0; i < size; i++) {
    S[i].display();
}
return 0;
}

```

O/P:

Enter details for staff 1:

Enter staff name: J

Enter staff post: HOD

Enter details for staff 2

Enter staff name: K

Enter staff post: Head

Enter details for staff 3

Enter staff name: L

Enter staff post: Head

Enter details for staff 4

~~Enter staff name: M~~

~~Enter staff post: Head HOD~~

staff who are HOD

| | |
|---------------|----------------|
| name of staff | who are HOD: J |
| name of staff | who are HOD: L |

Qn
Ans

Experiment 3:

1) WAD to declare

```
#include <iostream>
using namespace std;
class book
{
public:
    string book_title;
    int price;
    string author_name;
```

```
void accept()
```

```
{  
    cout << "Enter book name:";  
    cin >> book_title;  
    cout << "Enter the name of the author:";  
    cout << "Price of the book?";  
    cin >> price  
}
```

~~```
void display()
```~~~~```
{  
    cout << "Book name is: " << book_title;  
    cout << "\n author name: " << author_name;  
    cout << "\n price is: " << price  
}
```~~~~```
}
```~~

```
int main()
{
 book b1;
 book * p;
 b1.accept();
 b1.display();
 return 0;
}
```

O/P:

Enter book name : python  
Enter name of author: M  
Enter price of book: 5  
Author name is M priciss

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

{

```
int roll_no;
```

~~float~~

```
int percentage;
```

```
public:
```

```
void accept();
```

```
cout << "Enter roll no.: " << endl;
```

```
cin >> this->percentage;
```

{

```
void disp() {
```

```
cout << "Roll : " << this->roll << endl;
```

```
cout << "Penentage: " << this->percentage << endl;
```

{

~~};~~

```
int main() {
```

```
student s1;
```

```
s1.accept
```

```
s1.disp
```

```
return 0;
```

{

Output:

Enter roll no.: 12

Enter percentage: 15.1

roll no: 12

percentage: 15%

3) #include <iostream>  
using namespace std;

class Library {

public:

class Book

public:

string title;

void setTitle(string t) {

title=t;

}

void show() {

cout << "Book Title: " << title << endl;

}

};

};

~~int main () {~~

Library::Book b1;

b1.setTitle("The Alchemist");

b1.show();

}

Book title: The Alchemist

①  
12/11

## Assignment - 4

1 #include <iostream>  
using namespace std;

```
class Swap {
 int a, b;
public:
 void get() {
 cout << "Enter two numbers: ";
 cin >> a >> b;
 }
```

```
void swapValues(Swap s) {
```

```
 int temp = s.a;
```

```
 s.b = s.a;
```

```
 s.a = temp;
```

```
 cout << "After swapping" << s.a << ", b = " << s.b
 << endl;
```

```
}
```

```
};
```

```
int main() {
```

```
 Swap s1;
```

```
 s1.get();
```

```
 s1.swapValues(s1);
```

```
 return 0;
```

```
}
```

---

Enter two numbers: 12, 21  
After swapping: 21, 12

2. #include <iostream>  
using namespace std;

```
class swap
{
 int a, b;
public:
 void get() {
 cout << "Enter two numbers: ";
 cin >> a >> b;
 }
 void display() {
 cout << "a = " << a << ", b = " << b << endl;
 }
 friend void swapValues(swap &s);
}
```

```
void swapValues(swap &s) {
 int temp = s.a;
 s.a = s.b;
 s.b = temp;
}
```

```
int main() {
 swap s;
 s.get();
 cout << "Before swapping: ";
 s.display();
 swapValues(s);
 cout << "After swapping: ";
 s.display();
 return 0;
}
```

Enter two: 12, 21  
After swapping: 21, 12

2. #include <iostream>  
using namespace std;

```
class swap
{
 int a, b;
public:
 void get() {
 cout << "Enter two numbers: ";
 cin >> a >> b;
 }
 void display() {
 cout << "a = " << a << ", b = " << b << endl;
 }
 friend void swapValues(swap &s);
}
```

```
void swapValues(swap &s) {
 int temp = s.a;
 s.a = s.b;
 s.b = temp;
}

int main() {
 swap s;
 s.get();
 cout << "Before swapping: ";
 s.display();
 swapValues(s);
 cout << "After swapping: ";
}
```

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

Enter two numbers 10 20  
 Before swapping: a=10 b=20  
 After swapping: a=20 b=10

3. #include <iostream>  
 using namespace std;

```
class B;
class A {
 int x;
public:
 void get() cout << "Enter A: "; cin >> x;
 friend void swapValues(A &a, B &b);
 void show() { cout << "A = " << x << endl; }
};

class B {
 int y;
public:
 void get() cout << "Enter B: "; cin >> y;
 friend void swapValues(A &a, B &b);
 void show() { cout << "B = " << y << endl; }
};

void swapValues(A &a, B &b) { swap(x, y); }

int main() {
 A a; B b;
 a.get(); b.get();
 cout << "Before:\n"; a.show(); b.show();
```

```
swapValues (a, b);
cout << "After : \n"; a.show(); b.show();
}
```

Enter A: 12  
Enter B: 32

Before:

A = 12

B = 32

After:

A = 32

B = 12

4. #include <iostream>  
using namespace std;

class Result2;

class Result1;

float marks1;

public:

Void getMarks() {

cout << "Enter marks of student 1 : "

cin >> marks1;

}

friend void average (Result1, Result2);

}

```
class Result2 {
```

```
 float marks2;
```

```
public:
```

```
 void getMarks() {
```

```
 cout << "Enter marks of student in Result2:";
```

```
 cin >> marks2;
```

```
}
```

```
 friend void average(Result1, Result2);
```

```
}
```

```
 void average(Result r1, Result2 r2) {
```

```
 float avg = r1.marks1 + r2.marks2 / 2;
```

```
 cout << "Average marks = " << avg << endl;
```

```
}
```

```
int main () {
```

```
 Result r1;
```

```
 Result r2;
```

```
 r1.getMarks();
```

```
 r2.getMarks();
```

```
 average(r1, r2);
```

```
 return 0;
```

```
}
```

Enter marks of student in Result1: 32

Enter marks of student in Result2: 65

Average: 48.5

5. #include <iostream>  
using namespace std;

class B {

class A {

int x;

public:

void get() {

cout << "Enter number from class A: ";  
cin >> x;

}

friend void greatest(A, B);

}

class B {

int y;

public:

void get() {

cout << "Enter number from class B: ";  
cin >> y;

}

friend void greatest(A, B);

,

void (A a1, B b1) {

i + (a1.x > b1.y)

cout << "Greatest number is from class A: " <<

a1.x << endl;

else

cout << "Greatest number is from class B: " <<

b1.y << endl;

3

```
int main () {
```

```
 A a;
```

```
 B b;
```

```
 a.get();
```

```
 b.get();
```

```
 greatest(a,b);
```

```
 return 0;
```

}

Enter number from class A: 21

Enter number from class B: 18

Greatest number is from class A: 21

Ques  
12/11

## Assignment - 5

1 #include <iostream>  
using namespace std;

```
class number {
```

```
 int num;
```

```
public:
```

```
 number() { }
```

```
 cout << "Enter a number: " << endl;
```

```
 cin >> num;
```

```
 int sum = 0;
```

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

```
 sum = sum + i;
```

```
}
```

```
 cout << "The sum of numbers upto "
```

```
 "is : " << sum << endl;
```

```
}
```

```
};
```

```
number n;
```

```
return 0;
```

```
}
```

Enter a number: 5

The sum of numbers upto 5 is: 15

2. #include <iostream>  
using namespace std;

```
class Student {
 string name;
 float per;
 Student (string n, float p){
 name = n;
 per = p;
 }
```

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

```
int main () {
 Student s ("Ishaan", 99.2);
 s.display ();
 return 0;
}
```

Name: Ishaan  
Percentage: 99.2

3. #include <iostream>  
using namespace std;

```
class college {
 int roll;
 string name;
 string course;
public:
 college(int r=00, string n="Unknown", string c="Computer Engineering")
 roll=r;
 name=n;
 course=c;
}
```

Void display () {

```
cout << "Roll Number : " << roll << endl;
cout << "Name : " << name << endl;
cout << "Course : " << course << endl;
```

}

};

int main()

```
college s1(50, "Ishaan");
college s2(93, "John");
```

s1.display();

s2.display();

return 0;

}

Roll no: 50

Name : Ishaaan

Course : CSE

Roll no: 93

Name : John

Course : CSE

4. #include <iostream>  
using namespace std;

```
class student {
 int roll;
 string name;
public:
 student() {
 name = "Unknown";
 roll = 0; }
}
```

```
student(string n) {
 name = n;
 roll = 0; }
}
```

```
student(string n) {
 name = n;
 roll = r; }
}
```

```
void display() {
 cout << "Name " << name << endl;
 cout << "Roll No." << roll << endl;
}
3;
```

```
int main() {
 student s1;
 student s2 ("Ishaan");
 student s3 ("Alex", 99);

 s1.display ();
 s2.display ();
 s3.display ();

 return 0;
}
```

Name: Unknown

Roll no: 0

Name: Ishaan

Roll no: 0

Name: Alex

Roll no: 99

~~Qn  
(21)~~

## ★ Assignment 6

1. #include <iostream>  
using namespace std;

class department {  
protected:

    string dname;  
};

class student : protected department {  
protected:

    string sname;  
    int roll;

};

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

public:

void accept ()

cout << "Enter department : " << endl;

cin >> dname;

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

cin >> sname;

~~cout << "Enter marks 1 : " << endl;~~

~~cin >> m1;~~

cout << "Enter marks 2 : " << endl;

cin >> m2;

}

void calculate() {

```
int per = (m1 + m2) / 2 ;
cout << "Department: " << dname << endl;
cout << "Name: " << sname << endl;
cout << "Percentage: " << per << endl;
}
```

int main () {

```
mark m;
m.accept();
m.calculate();
return 0;
}
```

Enter Department: AIDS  
Enter name: Ishean  
Enter marks(m1): 91  
Enter marks(m2): 92

Department: AIDS

Name: Ishean

Percentage: 91.

2. #include <iostream>

using namespace std;

class department {protected: string dname;};  
class student {protected: string sname; int m1, m2};

class marks : protected department, protected student {

int m1, m2;

public :

void accept () {

cout << "Enter department, name, mark1, mark2  
:\n";

cin >> dname >> sname >> m1 >> m2;

}

void calculate () {

cout << "Department: " << dname

<< "\n Name: " << sname

<< "\n Percentage: " << (m1 + m2) / 2 << endl;

}

};

int main () {

marks m;

m.accept();

m.calculate();

}

Enter department, name, marks1, marks2:  
 AIDS, Fshaan, 78, 97

Department: AIDS

Name: Fshaan

Percentage: 97%.

3. #include <iostream>  
 using namespace std;

class person {

protected:

string name; int age;

public:

void getPerson()

{ cout << "Enter name & age";

"\n" >> name >> age;

}

};

class student : public Person {

int roll; float per;

public:

void getStudent() {

cout << "Enter student Name: " << name <<

"\n" << "Age: " << age

<< "\n" << roll << "\n" << "Percentage: " << per

<< endl;

};

class Staff : public Person {

    int id; string sub;

public :

    void getStaffID {

        cout << "Enter ID & subject:";

        cin >> id >> sub;

}

    void ShowStaff() {

        cout << "In staff Name: " << name <<

        " In Age : " << age << " In ID: " << id <<

        " In subject: " << sub << endl;

}

3.

int main() {

    Student s; Staff t,

    s.getPerson(); s.getStudent();

    t.getPerson(); t.getStaff();

    s.ShowStudent(); t.ShowStaff();

3

Enter name & age: Ishaan 17

Enter roll no & %: 50 99.9%

Enter name & age: Kevin 49

Enter ID & subject: 83 CFT

### Student

Name : Ishaan age: 17

roll no: 50

% : 99.9

### Staff

Name: Kevin

age: 49

ID: 83

subject: CFT

✓

h. #include <iostream>  
using namespace std;

```
class college {
protected:
 string emp_name;
 int id;
};
```

```
class staff : public Employee {
public:
 string name;
 int deptId;
 public:
 void acceptEmp() {
 cout << "Enter college name" << endl;
 cin >> name;
 cout << "Enter Employee name" << endl;
 cin >> emp_name;
 cout << "Enter ID:" << endl;
 cin >> id;
 cout << "Enter staff name:" << endl;
 cin >> name;
 cout << "Enter department id:" << endl;
 cin >> deptId;
 }
};
```

```
void displayEmp() {
 cout << "College: " << name << endl;
 cout << "Employee name: " << emp_name << endl;
 cout << "ID: " << id << endl;
 cout << "Staff Name: " << name << endl;
```

```
cout << Department ID: << deptID << endl;
}
}
```

class student : protected college {  
 string stu\_name;  
 int roll;  
 public:  
 void acceptStud() {  
 cout << "Enter college name: ";  
 cin >> name;  
 cout << "Enter name: " << roll;  
 cin >> stu\_name;  
 cout << "Enter roll no: " << endl;  
 cin >> roll;  
 }
 }

void displayStud () {  
 cout << "College: " << name << endl;  
 cout << "Student Name: " << stu\_name << endl;  
 cout << "Roll Number: " << roll << endl;
}
}

```
int main () {

 Staff staff1;

 staff1.acceptEmp();

 staff1.displayEmp();
}
```

```
student stud1;

stud1.acceptStud();
```

studl. display struct();  
return 0;  
}

O/P:

Enter college name: MIT  
Enter Employee name: OM  
Enter ID : 23  
Enter staff name: Ram  
Enter department id: HS

College: MIT

Employee name: OM

ID: 23

staff name: Ram

Dept ID: HS

Enter College name: MIT WPU

Enter name: Ishaan

Enter Roll no: 50

~~College : MIT~~

~~student : Ishaan~~

Roll no : 50

Qm  
12/11

## Assignment 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 (Rect) : " << a.
```

```
calculate (6, 7) << endl;
```

```
cout << "Area of Classroom (Square) : " << a.calculate
```

```
(5) << endl;
```

```
return 0;
```

```
}
```

Area of Laboratory (Rect) : 42

Area of Classroom (Square) : 25

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++)

s += a[i];

return s;

}

3.

int main () {

sum s;

int marks [10] = {40, 20, 30, 40, 50, 60, 70, 80, 90, 100};

float grades [5] = {6.7, 8.7, 9.7, 9.7, 5.7};

cout << "Sum of 10 student marks : " << s.total(marks, 10);

cout << "Sum of 5 student grade points : " << s.total(grade, 5);

return 0;

}

Sum of 10 students Marks : 550

Sum of 5 students grade points: 38.5

3. #include <iostream>  
using namespace std;

class Teacher {

int experience;

public:

Teacher (int x) {

experience = x;

}

void display () {

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

}

void operator -() {

experience = -experience;

}

};

int main () {

Teacher t1(10),

t1.display ();

-t1;

cout << "After negation: ";

t1.display ();

return 0;

}

Experience: 10 years  
After negation: -10 years

6. #include <iostream>  
Using namespace std;

```
class student struct {
 int count;
public:
 student (int c=0) {
 count = c;
 }
 void operator++ () { ++count; }
 void operator++ (int) { count++; }
 void display () {
 cout << "student count: " << count << endl;
 }
};

int main () {
 student s1(50);
 cout << "Before increment: " << endl;
 s1.display ();
 ++s1;
 cout << "After pre-increment: " << endl;
 s1.display ();
 cout << "After post-increment" << endl;
 s1.display ();
 return 0;
}
```

Before increment

student count: 50

After pre-increment

student count: 51

After post-increment

student count: 52

Qn  
12/1

## Assignment 8

1. #include <iostream>  
#include <string>  
using namespace std;  
public  
combine (string s = " ") { str = s; }

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

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

};  
int main () {

combine s1 ("xyz"), s2 ("par"), s3;  
 s3 = s2 + s1;

cout << "concatenated string : ";

~~s3.display ();~~

return 0;

}

Concatenated string : xyzpar

2. #include <iostream>  
using namespace std;

class ILogin {

protected:

string name, pass;

public:

virtual void accept () {

cout << "Name & password: ";

{in >> name >> pass}

}

virtual void display () {

cout << "Name : " << name << "Password : "

<< pass << endl;

}

};

class EmailLogin : public ILogin {

string email;

public :

void accept () override {

cout << "Email : " , cin >> email;

ILogin :: accept ();

}

void display () override {

cout << "Email Login" << email << endl;

ILogin :: display ();

};

int main () {

Email Login e; Membership Login m;

{ Login \* p = &e p->accept (); p->display ();  
p = &m; p->accept (); p->display ();  
}

Email: xyz@gmail.com

Name & Password: XYZ 123

-- Email Login --

Email : xyz@gmail.com

Name: XYZ

Password: 123

Member ID: 99

Name & Password: XYZ 231

- Membership Login - -

Member ID: 32

Name : XYZ

Password : 231

Qn  
[2/11]

## Assignment 9

```
1. #include <iostream>
#include <iostream>
using namespace std;
{
 int main () {
 ifstream infile ("First.txt");
 ofstream outfile ("Second.txt");
 if (!infile) {
 cout << "Error opening First.txt!" << endl;
 return 1;
 }
 char ch;
 while (infile.get(ch)) {
 outfile.put(ch);
 }
 cout << "File copied successfully!" << endl;
 infile.close ();
 outfile.close ();
 return 0;
 }
}
```

O/P :

Error opening First.txt

2 #include <iostream>

#include <fstream>

using namespace std;

int main ()

if stream file ("First.txt") ;

if (!file)

cout << "Error opening file! << endl;

return 1;

}

char ch;

int digits = 0, spaces = 0;

while (file.get(ch)) {

if (isdigit(ch))

digits++;

else if (isspace(ch))

spaces++;

}

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

cout << "Spaces: " << spaces << endl;

file.close();

return 0;

}

O/P:

Error opening

```
3. #include <iostream>
include <fstream>
include <string>
using namespace std;

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

O/P

Error Opening file!

61. 

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

```
int main () {
 ifstream file ("first.txt");
 if (!file) {
 cout << "Error opening file!" << endl;
 return 1;
 }
 string word, target;
 int count = 0;

 cout << "Enter word to count: ";
 cin >> target;

 while (file >> word) {
 if (word == target)
 count++;
 }
 cout << "Occurrence of " << target
 << endl << count << endl;
 file.close();
 return 0;
}
```

O/P:

Error opening file!

Ques  
Ans

## Experiment 10

1 #include <iostream>  
using namespace std;

template <class T>

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

int main () {

int int\_arr[5] = {1, 2, 3, 4, 5};

float float\_arr[5] = {1.1, 2.2, 3.3, 4.4, 5.5};

double dec\_arr[5] = {0.5, 1.5, 2.5, 3.5, 4.5};

```
cout << "Sum of Integer array" << sumArray(int_arr, 5) << endl;
cout << "Sum of float array" << sumArray(float_arr, 5) << endl;
cout << "Sum of Double array" << sumArray(dec_arr, 5) << endl;
return 0;
}
```

O/P:

Sum of Integer array : 15  
 Sum of float array: 16.5  
 Sum of double array: 12.5

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

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

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

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

```
cout << "Square of integer: " << square (num) << endl;
cout << "Square of double: " << square (dec_num) << endl;
cout << "Square of string: " << square (str) << endl;
return 0;
}
```

O/P:

Square of integer: 25  
Square of double: 6.25  
Square of string: AppleApple

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

template <class T1, class T2, class calc>  
public:  
T1 x;  
T2 y;

calc (T1 n, T2 m) {

x = n;  
y = m;  
}

void sum() {

cout << "sum : " << x + y << endl;  
}

void diff() {

cout << "Difference : " << x - y << endl;  
}

void pro() {

cout << "Product : " << x \* y << endl;  
}

void rem() {

cout << "Reminder : " << x % y << endl;  
}

void power() {

cout << "x raised to y is : " << pow(x, y) << endl;

}

```
void min_num() {
```

cout << "Min of numbers: " << min(a, b);

{}

```
void max_num() {
```

cout << "Max of the numbers" << max(a, b);

{}

```
void sin_x() {
```

cout << "Sin of x: " << sin(x) << endl;

{}

~~```
void cos_y() {
```~~~~cout << "Cos of y: " << cos(y) << endl;~~

{}

{;}

```
int main() {
```

calc <int, int> n(100, 200);

n.sum();

n.diff();

n.pro();

n.q40();

n.power();

n.min_num();

n.max_num();

n.sin_x();

n.cos_y();

{;}

O/P:

Sum = 300

Difference = -100

Product = 20000

Quotient = 0

Remainder below

x raised to y is : inf

Min of numbers: 100

Max of numbers: 200

~~sin of x :~~ -0.506366

~~cos of y :~~ 0.487188

Qn
12/11

Assignment 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;
vector <int>::iterator I;

for (I = v.begin(); I != v.end(); ++I) {
cout << *I << " ";
}

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

for (I = v.begin(); I != v.end(); ++I) {
*I = (*I) * 10;
}

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

for (I = v.begin(); I != v.end(); ++I) {
cout << *I << " ";
}

}

return 0;

}

Initial Vector:

1 2 3 4 5 6 7 8 9 10

Multiply by 10

New Vector:

10 20 30 40 50 60 70 80 90 100

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

```
int main() {
    vector<int> v = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10} << endl;
    for (int i = 0; i < 10; i++) {
        cout << v[i] << " ";
    }
}
```

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

}

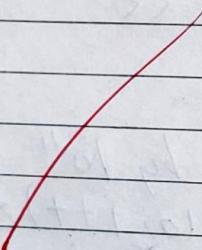
```
cout << "New Vector: " << endl;
for (int i = 0; i < 10; i++) {
    cout << v[i] << " ";
}
return 0;
}
```

Initial Vector:

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10

Multiply by 10
New vector.

- 60
- 20
- 30
- 40
- 50
- 60
- 70
- 80
- 90
- 100



Q
12/11

Experiment 12

★ 1) #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();

cout << endl;

cout << "Queue elements (front to back) are: ";

while (!q.empty()) {

cout << q.front() << " ";

q.pop();

}

return 0;

}

O/P:

Front element: 0

Back element: 100

After one pop, front = 10

queue elements (front to back): 10 20 30
50 60 70 80 90 100

2. #include <iostream>

#include <stack>

using namespace std;

stack<int> stack1;

while (!temp.empty()) {

cout << temp.top() << " " ;
temp.pop();

}

cout << endl;

}

int main () {

int num;

cout << "Enter the number of elements you
want in the stack " << endl;

(cin >> num);

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

int temp;

cout << "Enter element at position " << i <<

endl;

(cin >> temp);

stack1.push(temp);

}

cout << endl;

cout << "Top most elements : " << stack1.top() << endl;

cout << endl;

cout << " stack element (top to bottom) -> " << endl;

disp();

```
cout << endl;
cout << "pop function: " << endl;
stack1.pop();
disp();
stack1.pop();
disp();
stack2.pop();
disp();
```

return 0;

}

Enter the number of elements you want in stack1:

Enter number at position 1: 1

Enter number at position 2: 2

Enter number at position 3: 3

Enter number at position 4: 4

Enter number at position 5: 5

Top most element: 5

stack1 elements (top to bottom) →
5 4 3 2 1

Pop function:

43 21

32 1

~~2 1~~

Qm
12/11