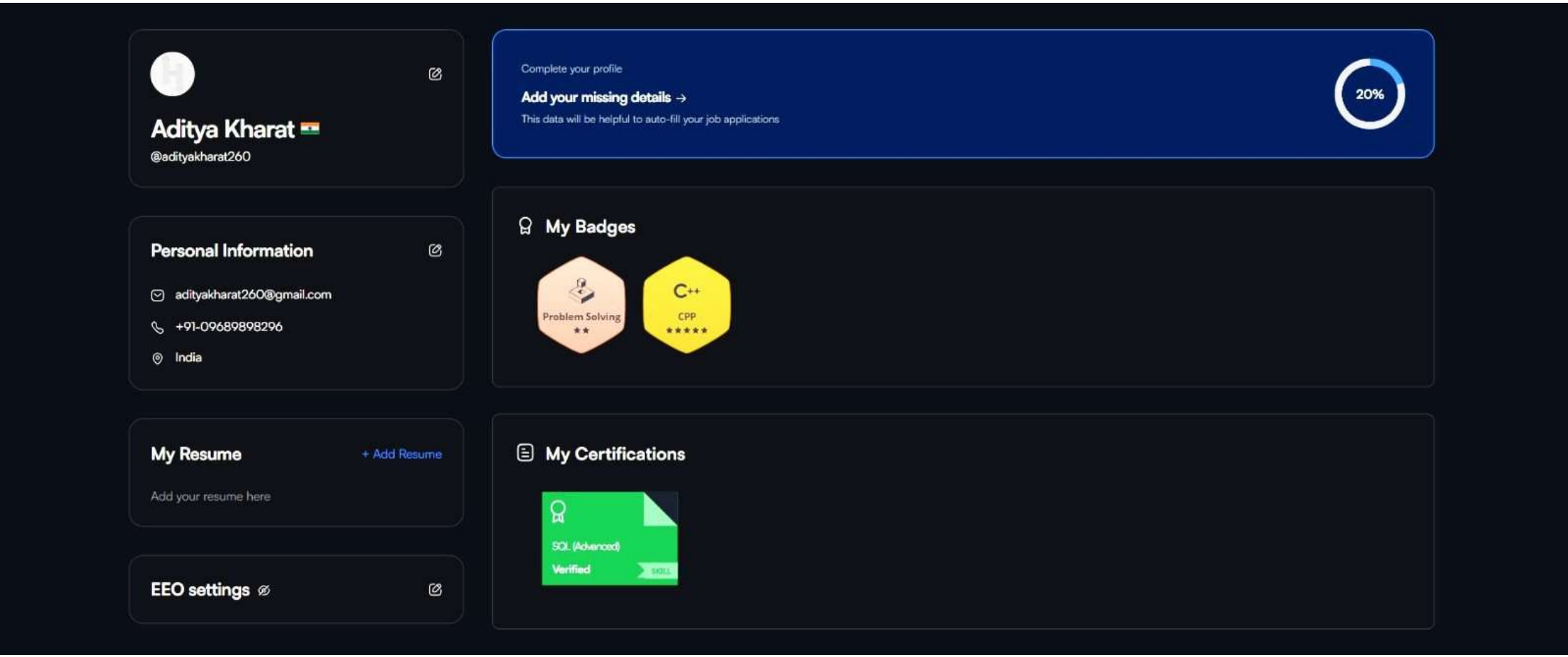

Aditya Kharat
@adityakharat260
Personal Information
adityakharat260@gmail.com
+91-09689898296
India
My Resume
+ Add Resume
Add your resume here
EEO settings
Complete your profile
Add your missing details →
This data will be helpful to auto-fill your job applications
20%
My Badges
Problem Solving
C++
CPP
My Certifications
SQL (Advanced)
Verified

M T W T F S S

Page No.: YOUVA

Date:

1] Add 2 numbers

```
#include <iostream>
 int main (){
 int a, b, sum;

Std:: cout <<" Enter two no =";
Std : cin >> a >>b;
  Sum = a+b;
Std :: cout<< " sum is " <<sum;
 return 0
}
```

output:

Enter 2 number 3 4
sum is 7

(2) Arithematic operations

```
#include <iostream>
int main (){
int a, b;
char op;

Std:: cout <<" Enter first number";
Std :: cin >> a;
Std :: cout <<" Enter Second number";
Std :: cin >> b;

Std: cout << " Enter an operation (+, -, *, /) ";

Switch
```

```
switch (op)}

Case (+)
Std:: cout <<:Result ="<<a+b<<endl;
break
Case '-';
Std: : cout <<" Result ="<<a-b<<endl;
break
case (*)
Std ::out <<Result ="<<a*b<<endl;
break
case :/'
Std:: ( out <<result ="<<a/b <<end l;
break
}}
Output;
```

3) Check no. is even odd

```
#Include <iostream>
int main (){
int num, even, odd;
Std :: Cout <<"Enter an integer:";
Std ::cin >>num.;
If (num % 2 == 0)
{
std:: cout <<num <<" is even:";
} else {
 sta:: cout <<num " is odd."
}
return 0:}}
```

M T W T F S S

Page No.: YOUVA

Date:

4) Print 1 to 10 numbers using for loop

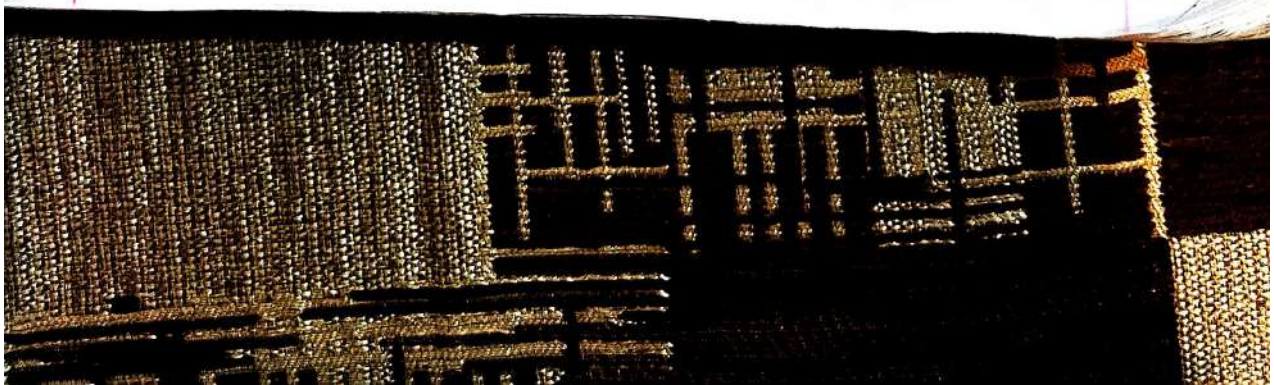
→
```
#Include <iostream>
int main(){
int i;
for (i=1;i<=10; i++){
Std :: cout <<i<<Std :: \n;
}
return 0;
}
```

output

1
2
3
4
5
6
7
8
9
10

Using while loop

```
#include <iostream>
int main (){
int i;
while (i<=10){
Std:: cout <<i<<" \n";
i= j +1;
}
return 0;
}
```



6) Print
```
*
* *
* * *
```

```
#Include <iostream>
int main (){
 int rows = 3;
 for (int i = 1; i<=rows; i ++){
 for (int j =1 ;j <= i; j ++){
 std:: std:: cout << "*" ;
 }
 Std :: cout << endl;
 }
```

2)
```
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
```

```
# Include <iostream>
int main (){
int main (){
int rows = 5;
for (int i = 1 ; i <= rows ; i ++){
for (int j = 1; j <= i; j ++){
Std:: cout << i;
}
Std :: cout << endl
}
return 0
}
```

3) #include <iostream>
using namespace std;
int main () {
int r = 5;
for (int i = 1; i <= r; i++) {
for (int j = 1; j <= i; j++) {
cout << "* ";
}
cout << endl;
} return 0; }

9/7/25

M T W T F S S
Page No.:
Date:
YOUVA

1) Write a program to declare a class Student, having data members also roll no, name, accept and display data for one object

→
```
#Include <iostream>
using name space Std;
int roll;
String name;
public
void accept ()
{cout <<"Enter the roll no.";
 Cin >> roll
 cout <<" Enter the name of Student";
 cin >>name
}
Void display ()
{cout <<" The roll no of Student is
 " << roll <<end l;
 cout <<" The name of student is <<name <<endl;
 }
 }
int main (){
Students;
S. accept ();
Cout <<" Data :" <<end l;
 S. display ();
 return 0;
 }
```

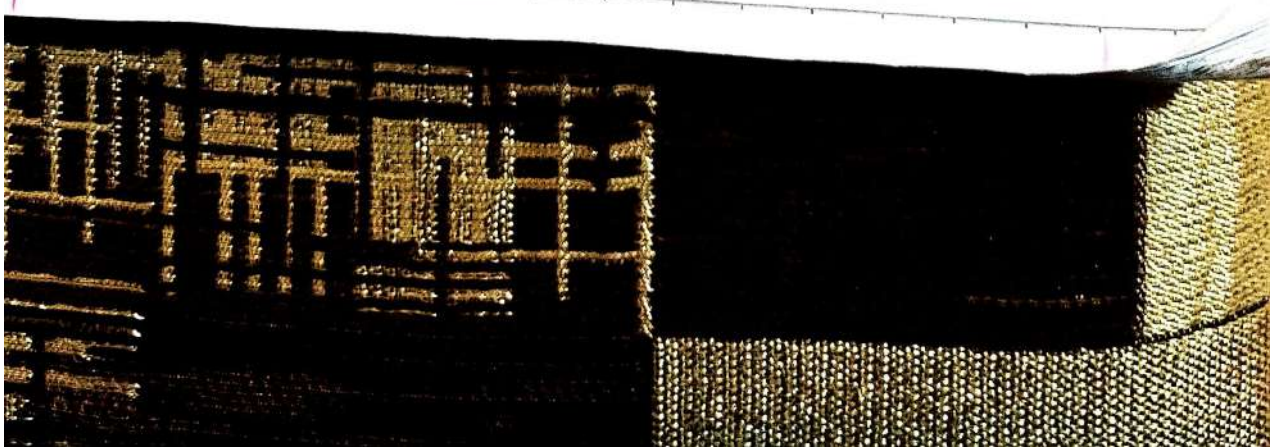
output :-
Enter the roll no:7
enter the name of student : Aditya

2) #Include <iostream>
using namespace Std;
class book
{
String name:
float price:
int pages:
Public
void accept c)
{
Cout <<"Enter the name of book",
Cin >> name;
Cout<<" Enter the price of book"
Cin >> price
Cout << "Enter the number of pages";
Cin >> name
Cout << " Enter number of pages";
Cin >> Pg;
}}
int main (){
book b1, b2
b1. accept
b2. accept
If (b1. price > b2. price){
Cout << " The book with greater price
is: "<<b1. name;
}
else{
Cout <<" The book with Greater price is:
<<b2. name;}
}
return 0;}

Date:

☆ out

Enter the name of the book: Hello
Enter the price of the book: 199
Enter the no. of pages of the book: 200
Enter name of the book : world
Enter the price of the book: 600
world is more expensive.



```
#Include <iostream>
using name space std;
class time
{
public:
 int h, m, s, con;
 char colon 1, colon 2
 void accept c){
 Cout <<" Enter time (HH:MM:SS): ";
 char colon 1, colon 2;
 cin >> n >> m >> colon 2>>s;
 }
 Void convert c){
  con = n * 6·3600+m * 60+s;
  cout << con;
   }}
 int main c){
  time t;
  t accept c);
  t convert;
  }
```

30/7/25

```
out
Enter no. of hours : 06
Enter no. of minute : 45
Enter no. of Seconds: 24
time is : 6: 45: 24
total time in Seconds in 24324
```

Experiment-2

Q1)

```
#Include <iostream>
 using name space Std;
{
 public:
String name;
 int p;

 void accept()
{
 cout << "Enter the name of city".
 cin >> name;
 cout << "Enter the population of city";
 cin>>p;
}
 Void display()
{
 cout<<" city:" <<name <<", population" <<p<<endl;
}
};
 int main()
{
 city cities[5];
```

P.T.O

```
for (int i=0; i<5; i++){
 cout << "Enter details of city" <<i+1<<"\n";
cities [i].accept();
}
int maxIndex = 0;
 for (int i=1; i<5; i++){
if (cities [i] .p> cities [maxIndex].p){
}
}
cout <<"city with highest population is :" <<cities
[maxmIndex]. name;
cout<<"(" <<cities [max Index]. p<<"people )"
endl;
```

* output

Enter city nam & population : city 1
6567 34

Enter city name & population: city 2
654321

Enter city name ~~of~~ & population: city 3
672738

Enter city name & population : city 4
60001

Enter city name & population: city5
~~09~~ 61234

~~The~~ The population 6576734 of the city1

Q2)

```
#Include <iostream>
using namespace std;
 class acc {
public:
 int acc_no [3];
float bal [3];
int i;
float balance:

Void accept C){
Cout<<"Enter the accaunt number and Balance";
for (i=0; i<3;i++){
cin>>acc_no[i]>>bal [i];
    }
 }
void display C){
 for (i=0; i<3;i++){
   if (bal [i]>=5000){
   balance=bal [i] + (bal [i]* 10* 1)/100:
   Cout << "Account name and balance after
   10% increase:" <<acc_no [i]<<" " <<
   balance <<endl;
      }
   }
 }
```

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

```
Enter account no. and balance 1
                              2900
Enter account no. and balance 2
                              3400
Enter account no. and balance 3
                              7800
Enter account no. and balance 4
                              5000
Enter account no. and balance 5
                              2900
Enter account no. and balance 6
                              7800
Enter account no. and Balance 7
                              1300
Enter account no. and Balance 8
                              6700
Enter account no. and Balance 9
                              4000
Enter account no. and Balance 10
                              7000
The interest offered will be 10%.
to the acount 3 account after
getting: 8580
```

Q3)

```
#Include <iostream>
# include <strings>
using name space std;
class staff
public:
 char name[50];
 char designation [50];
 void accept (){
  cout<< "Enter the name:";
  cin>> name;
  cout <<"enter the designation:";
  cin >> designation;
}
void display (){
    cout <<" the name is :" <<name;
    cout << " the designation is :" << designation;
}};

int main (){
 staff s[5];
 char hod [] ="hod";
for (int i=0; i<5; i++){
 s[i] .accept ();
  }
for (int i=0; ) <5; i++) {
```

```
S[i].display();
}
for(int i=0; i<5; i++){
if(strcmp(S[i].nam
```

* Output

Enter name and Post: AA
physic
Enter name and Past: BB
HOC
Enter name and Past: CC
~~match~~ + maths teacher
Enter name and post: EE
class teacher
Name of HOD: BB

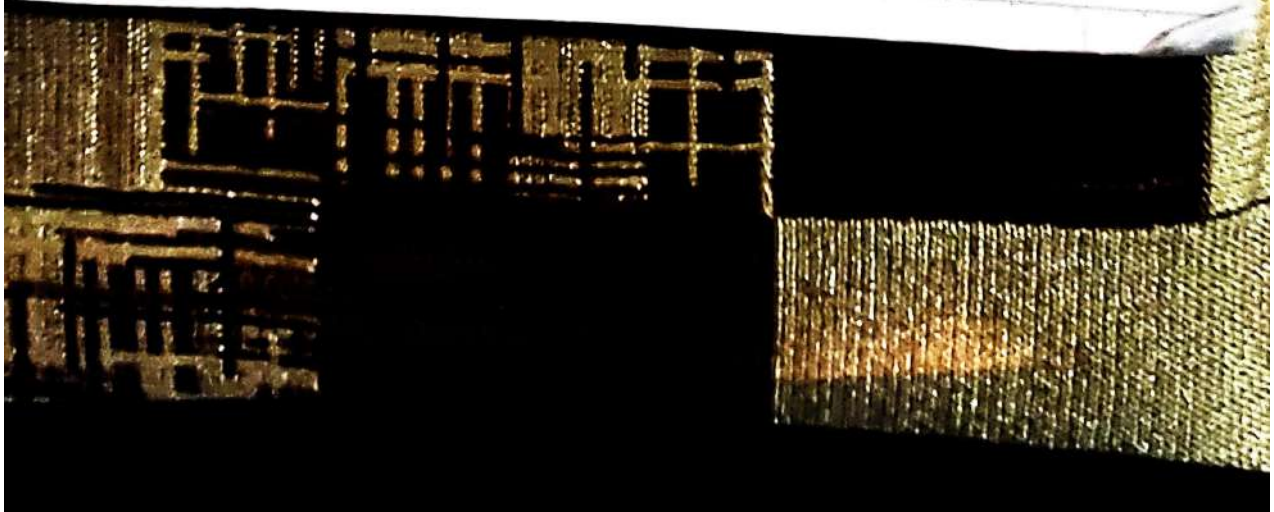
Nam of HOB: DD

30/7/25

Exp-3

a]

→
```
#Include <iostream>
using namespace std
class Book {
Private
int price;
char title [20],
 char author [20].
public:
void accept() {
Cout << " Enter the price of the book \n";
Cin >> price;
Cout << " Enter the title of the book \n";
Cin >> title;
Cout << " Enter the author of the book \n";
Cin >> author;
] };
int main() {
 book b1;
book *, p.
 p = & b1;
p → accept ();
p → display
```



h]

```
#Include <iostream>
using namespace std;
class student{
private
 int roll;
 float per;
Public
void accept (){
 cout << " Enter the roll no. of the student :"
cin >> this -> roll;
cout << " Enter the percentage of student :";
cin >> this -> per;
}
Void display (){
   cout << Roll " << this -> roll << endl;
   cout << Percentage " << this -> per << endl;
    }
}
int main (){
Student s1;
s1. accept ();
s1. display ();
```

c]

```
#include <iostream>
using namespace std;
class student {
public.
clas. marks{
private:
int roll;
floal per;
Public,
Public:
Void accept (){
cout << "Enter the roll number of student:";
cin >> roll;
cout <<"Enter the percentage of student:",
cin >> per;
}
void display (){
cout <<" Roll number :" << roll << endl;
cout << " Percentage ." << per << endl;
}.
};
int main (){
 Student:1 marks m1
m1. accept ();
m1 . display ();
```

}

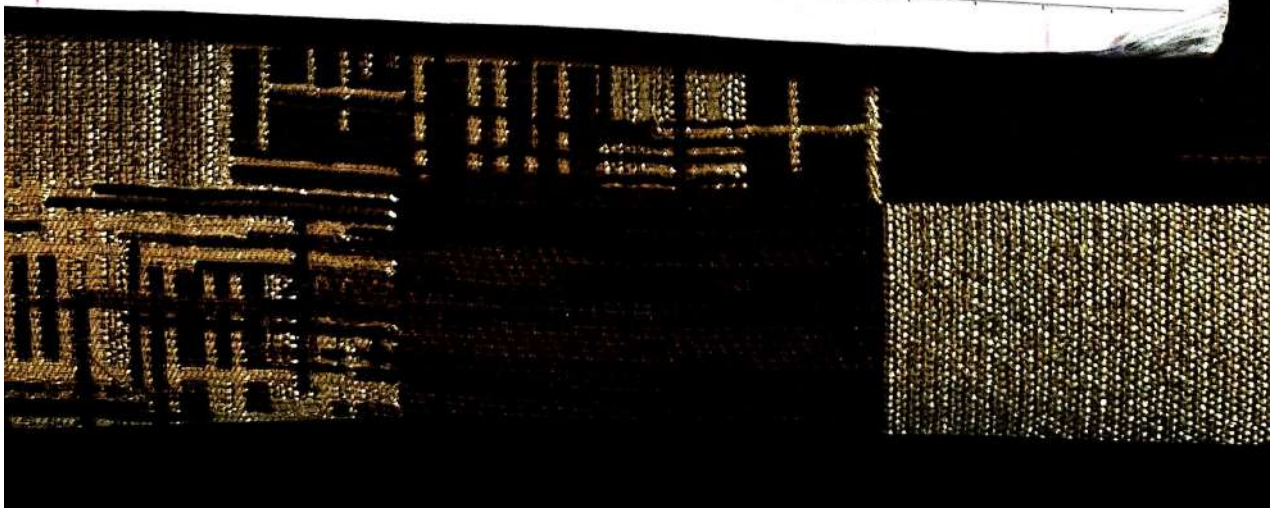
Experiment 14

1 →

```
#include
using namespace std;
class result2;
class result1;
{
int m1;
public:

void accept1()
{
  cout<<"Enter the marks of M1=");
  cin>>m1;
}
friend void calculate(result1 p, result2 r);
};
class result2
{
 int m2;
Public:

Void accept2()
{
 out<<"Enter the mark of M2=";
  cin>>m2;
}
friend void calculate (result1 ,p, result2 r)
};
void calculate (result1 ,p, result2 r)
```

P.T.O

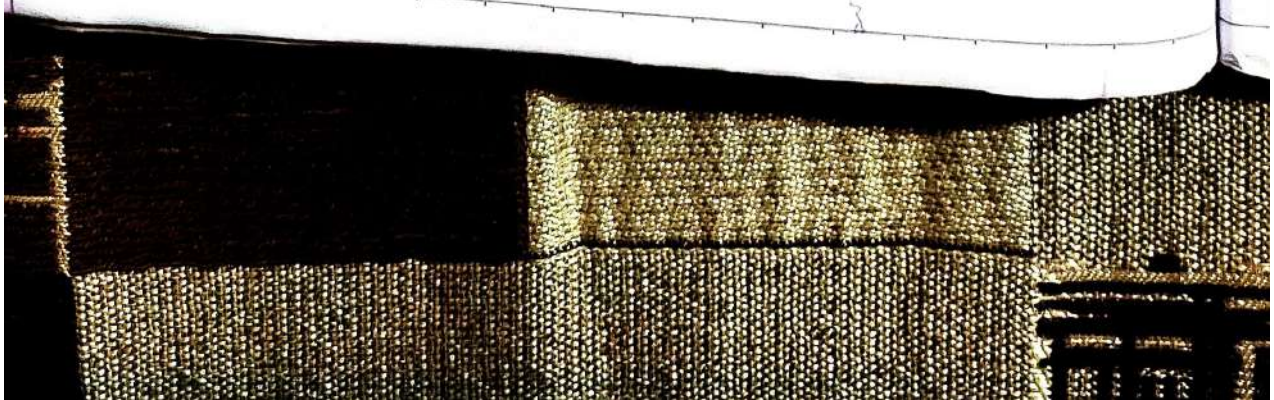
```
}
int total = (P.m1 + r.m2) / 2;
 cout << " Average of the result = " << total << endl;
}
int main ()
{
 result P;
 result 2 r;
P. accept ();
r. accept 2 ();
calculate (P, r);
}
```



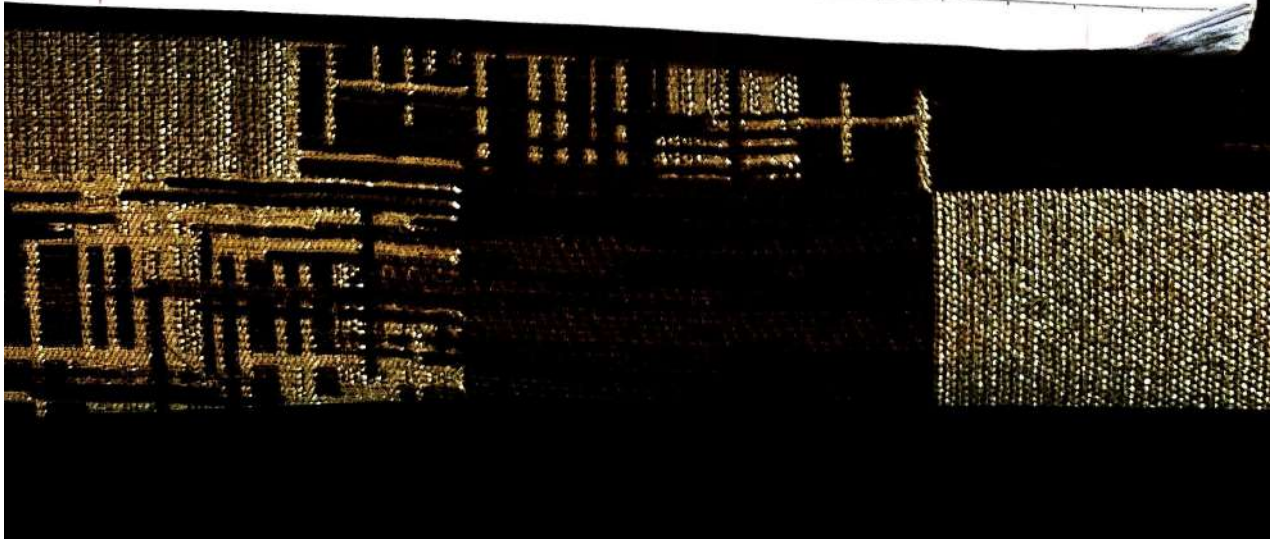
2) WAP ... to find Greatest among two number

→

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

class number2;

class number1{
  int num1;
public:
  void accept1(){
   cout<<"Enter the number 1= ";
   cin>>num1;
  }
  friend void calculator(number1 p, number2
  r);
};
class number2{
   int num2;
public:
  void accept2(){
   cout<<"Enter the number 2 = ";
   cin>>num2;
}
friend void calculate(number1 p, number2 r);
};

void calculate(number1 p, number2 );
};
void calculate(number1 p, number2 r){
if (P.num1>r.num2){
   cout<<" The greatest number
  is: "<<P.num1<<endl;
```

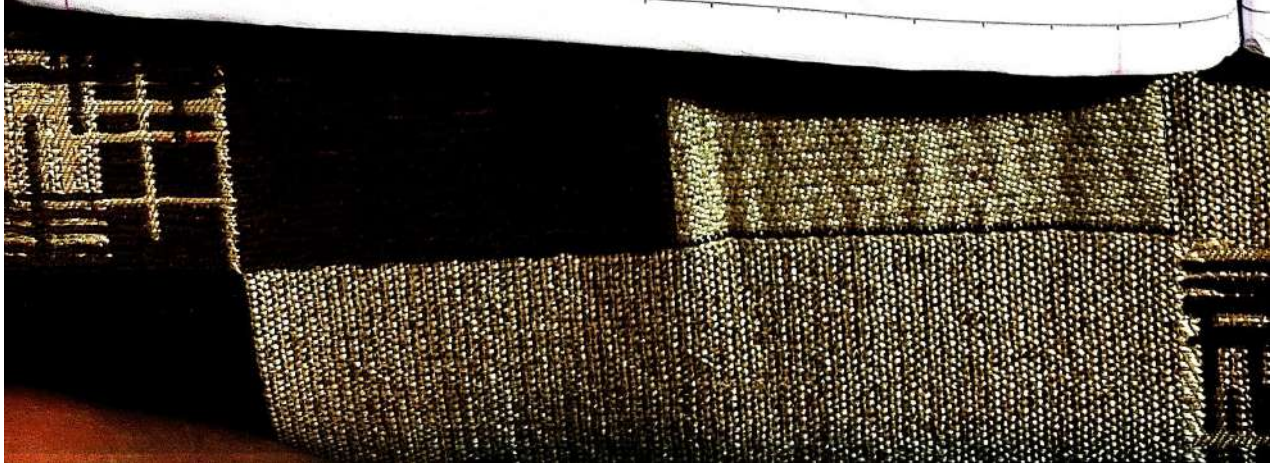


```
}
else{
cout<< "The greatest number is:" <<r.num2
 <<endl;
}
}
int main(){
number lp,
number )r;
p.accept();
r.accept();
calculate(p.r);
}
```



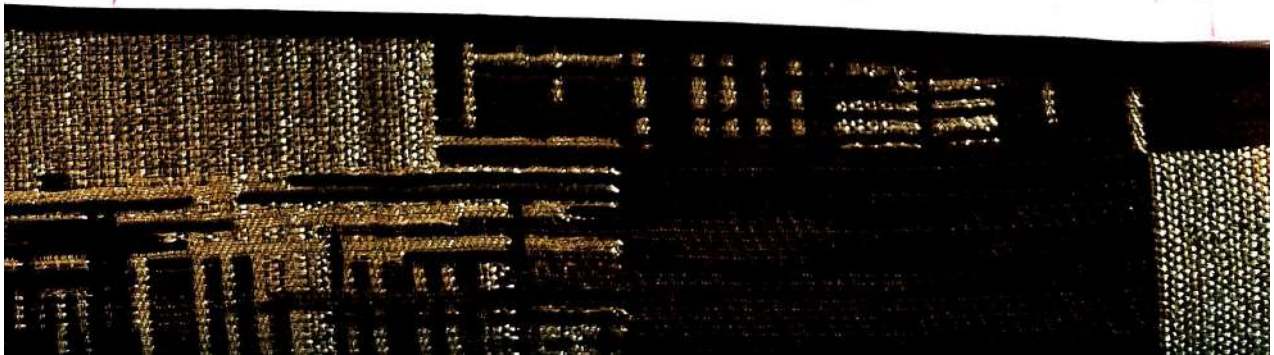
2) Swapping 2 nos without using friend function

→
```
#include <iostream>
using namespace std;
class demo{
public:
  int p, q;
  void accept(){
  cout<<"Enter 2 nos:-"<<endl;
  cin>>p>>q;
  }
  Void display(){
  cout<<" after swapping :-"<<"value of p="<<p
                <<"value of q="<<q;
}
void swap(demo st){
int temp = t.p;
t.p = t.q;
t.q = temp;
}};
int main(){
 demo k;

k.accept();
k.swap(k);
k.display();
}
```



Q8

→
```
#include <iostream>
using namespace std;
class demo{
 int a,b;
 public accept(){
    cout<<"Enter two numbers:-"<<endl;
 }  cin>>a>>b;
 }
    void display(){
    cout<<"value of a:-"<<a;
    cout<<"value of b:-"<<b;
}
 friend void swapnums. (demo st);
};
void swapnums (demo st){
     int temp = t.a;
      t.a = t.b;
      t.b= temp;
}
int main(){
   demo k;
   k.accept();
   Swapnums(k);
   k.display();
}
```



9

```
#Include <iostream>
using namespace std;
 class B;
 class A{
   int a;
    public
   Void accept 1 C){
   cout <<" Enter a:-" <<endl;
   Cin >>a;
}
   void display C){
   cout <<" value of a :-" <<a;
}
  friend void swap numbers (A, B &2);
};
  class B{
       int b;
       public:
       void accept 2 C){
       cout <<" Enter b :-" <<endl;
       Cin >>b;
}
     void display2C){
       cout <<" Value of b:-"<<b;
}
   friend void swapping numbers (A&P, B&2);
};
```

Date:

```
Void Swap number (A &p, B &q) {
int temp = p. aj
  p.a = q.b;
  q.b = temp;
}
int main (){
   A k;
   B f;
k. accept1();
f. accept2();
Swap numbers (k, F);
k. display1();
f. display2();
}
```

Q1) Create two classes & classB, each with a private integer. write, a friend function sum() that can access private data from both classes & return the sum.

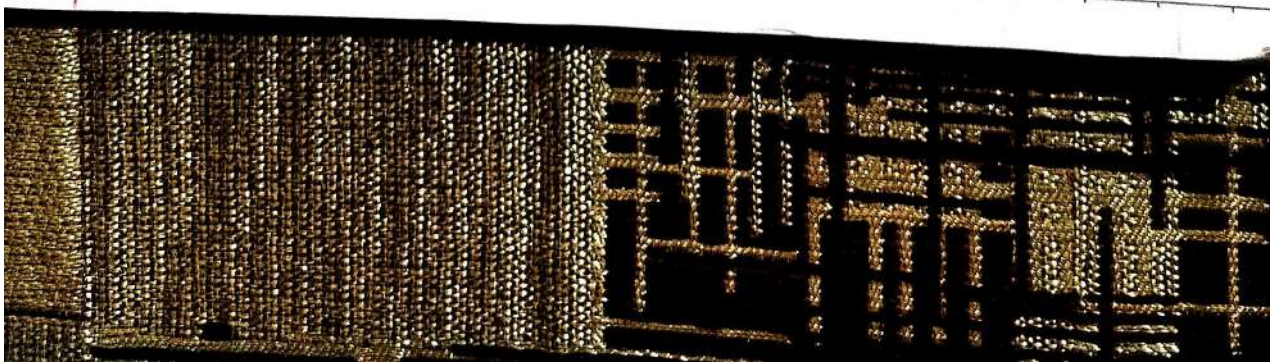
```
#Inclued <iostream>
using namespace Std;
Class B;
class A {
int a;
public:
 void accept (){
  cout <<"Enter the first number :-" <<endl;
  cin >> a;
 }
 frin friend int sum (A p, B f);
};
Class B{
 int b;
 public:
  Void accept() {
   cout <<" Enter the Second number :-" <<endl;
   cin >> b;
}
friend int Sum (A p, B f);
};
int sum (A p, B q){
  int sum;
   sum = p.a + q.b;
   return sum;
```

```
}
int main(){
A k;
B f;
k.accept();
f.accept();
cout << " the Sum of 2 numbers is :- " << sum(k,f);
}
```

Page No.: YOUVA
Date:

Q2) Write a program with a class that contains a private integer. Uses a friend function Swap Numbers (Number &, Number &) to swap the private value of two number object

→
```
# Include <iostream>
using namespace std;
class A{
 int a,b ;
public:
  Void accept (){
  cout <<" Enter two numbers:-"<<endl;
  cin >>a>>b;
  }
  friend void swapnumbers (A &t);

  Void display(){
   cout << Value of a:-"<<a;
   cout << the Value b:-"<<b;
  }
  friend void swapnumbers(A &t);
};
void swapnumbers (A &t){
int temp = t.a;
t.a=t.b;
t.b= temp;
}
int main (){
 A k;
 k. accept ();
 Swapnumbers (k);
 k.display();
}
```



Date: YOUVA

Q) Define two classes Box & cube, each having a private Volume. Write a friend function findGreater (Box, cube) that determines which object has a larger volume

→
```
#include <iostream>
using namespace std;
class cube;
class box {
int l,b,h,v1;
public:
 void accept1(){
 cout<<" enter the dimensions of the box"<<endl;
 cin>> l >> b >>h;
 }
 friend void greaterVolume (Box p, cube q);
};
Void greaterVolume (Box p, cube q){
 p.v1 = p.l * p.h * p.b;
 q.v2 = q.side * q.side * q.side;
 if (p.v1 > q.v2){
 cout<<" the box having greater Volume is
        :- "<<p.v1<<endl;
 }
 else {
   cout<<"the box having greater Volume is
         :- "<<q.v2<<endl;
 }
}
int main (){
  box k;
```

P.T.O

```
Cube f;
k.accept();
f.accept();
greaterVolume(k,f);
}
```

Page No.:
Date:
YOUVA

Q4)

→
```
#Include <iostream>
using namespace Std;
  int re,im;
public;
    void accept(){
cout<<"Enter the real & imaginary parts"<<endl;
cin>>re>>im;
}
friend cmplx add(cmplx.c1, cmplx c2, cmplx result)
void display(){

 cout<<re<<"+"<<in<<"i"<<endl.;
}
Friend cmplx add cmplx (cmplx c1, cmplx c2,
  cmplx result);
};
Complex cmplx add cmplx (cmplx c1, cmplx c2,
cmplx result){
result.re = c1.re+c2.re;
result.im = c1.im+c2.im;
return result;
};
int main(){
   cmplx k, f, r;
    k.accept();
    f.accept();
    r = add cmplx (k, f, r);
```

```
t.display();
}
```

Page No.:
Date:
YOUVA

Q5) Create a class Student with private data members: name & three subject marks. write a friend function calculate Avg (student) that calculates & display the avg marks

```
#include <iostream>
using namespace std;
class student {
    char name[50];
    int m1, m2, m3
public:
    Void accept (){
      cout<<" Enter the name of the Student:-"
      <<endl;
    cin >> name;
    cout <<" Enter m1:-" <<endl;
    cin >>m1;
    cout <<" enter m2:-" <<endl;
    cin >>m2;
     Cout <<" enter m3:-" <<endl;
     cin >> m3;
}
friend void calculate avg (student p){
  int avg;
   avg= (p.m1+p.m2+p.m3)/3;
   cout <<" the avgerage of the student
   mark is:-" <<avg <<endl;
}
int main(){
    Student k;
```

P.T.O

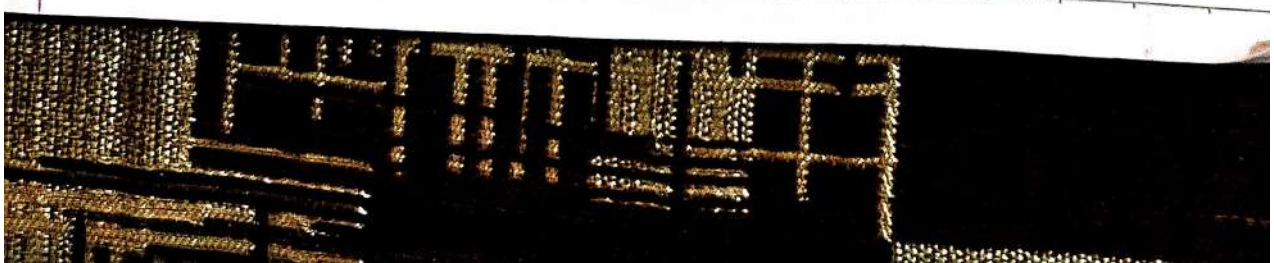
```
k.accept();
calculating(k);
}
```

86) Create classes Alpha, Beta & Gamma, each with a private data member. write a single friend function that can access all three & print their sum

→
```
#include <iostream>
using name space std;
class beta;
class gamma;
class alpha;
int a;
public:
   void accept1C){
cout<<" enter a:- ";
cin>>a;
}
friend void sum Calpha p, beta q, gamma r);
};
class beta{
int b;
public:
void accept2C){
 cout<<" enter b:-";
  Cin>>b;
}
 friend void sum Calpha p, beta q, gamma r);
};
else gamma{
int c;
public:
 void accept3C){
cout<<" enter c:-";
```

P.T.O

```
cin>> c;
}
friend void sum(alpha p, beta q, gamma r);
};
void sum(alpha p, beta q, gamma r){
int sum;
sum= p.a +q.b + r.c;
cout<< "the sum is:-" <<sum<<endl;
}
int main(){
 alpha k;
 gamma z;
 beta q;
 k.accept1();
 q.accept2();
 z.accept3();
 sum( k.q,z);
```



Q7) Create a class Point with private member x and y. write a friend function that calculate and returns the distance between two point object

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

class Point{
private:
   int x, y;

public;
   void accept (){
   cout <<" Enter x and y:";
   cin>> x >>y;
}
  Friend void findDifference (Point p1, point p2);
};
void findDifference (Point p1, point p2){
int diffx = p2.x - p1.x;
int diffy = p2.y - p1.y;

cout<< " difference in x :" << diffx <<endl;
cout << " difference in y :" <<diffy <<endl;
};
int main (){
point a,b;
cout <<" Enter the point 1:" <<endli
a.accept ();
```

```
cout << "Enter point 2:" <<endl;
b.accept();

find difference (a, b);
return 0;
}
```

Q8) Create two classes : Bank account & Audit. Bankaccount holds private info...

→
```
#Include <iostream>
using namespace std;
 class audit;
class Bankaccount {
int Balance;
public:
   Void accept (){
     cout <<" Enter the balance:," <<endl;
     cin >> balance;
  }
friend void accessbalance (Bankaccount k);

};
Class audit {
public:
  friend void accessbalance (Bankaccount k);
};
Void accessbalance (Bank account k){
 int pvtBalance
 pvt balance = k.balance;
 Cout <<" the Balance for auditing is :-"
 << Pvt balance;
}
int main (){
Bank account a;
a.accept ();
access balance (a);
}
```

13/8

Page No.: YOUVA
Date:

Experiment - 5 (Implement 4 types of constructor)

Q WAP to find the sum of no. btwn 1-n using constructor

→

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

class sum{
  int n;
  int sum;

public:
    sum(int num){
      in n=num
        sum  =0
  for (int i = 1; i<=n; i++){
    sum = sum+i;
    }
  }

  void display (){
    cout<<" Sum of numbers from 1 to "<<n<<" is"
    <<sum<<endl;
      }
  };
int main(){
  int n;
  cout <<" Enter value of n:";
  cin >> n;
   sum scn)
   s. display
    return 0;
  }
```

default

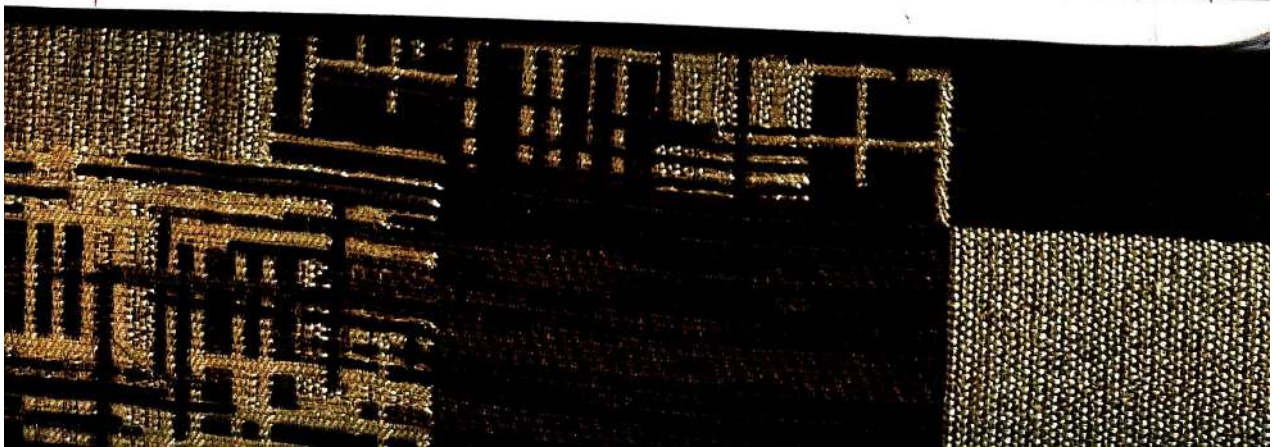
```
sum (){
  n=10
  sum=0;
for (int i=1; i<n; i++){
  sum = sum+i
}
```

def, par, cpy

copy

```
sum (sum &obj){
  n = obj. n
  Sum = obj. sum;
}
```

copy:	default
sum 51 (5)	sum 5;
sum 52 (51)	s. display ()



Q WAP to declare class student having Data members as name and percentage. Write a constructor to ~~tAitait~~ initalize these data members. Accept & display data data for one student

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

class student {
    string name;
    float percentage;

Public:
    Student (string n, float p) {
        name = n;
        percentage = p;
    }
    Void display () {
    cout << " Name : " << name << "\n percentage : " << Per-
    centage;
    }
};
int main () {
    string name;
    float percentage;

cout << " Enter the for student : ";
cin >> name;
cout << " Enter percentage for student : ";
cin >> Percentage;
Student s1 (name, percentage);
```

default:
Sum () {
n = 10;
Sum = 0;
for (int i = 1; i < n; i++
Sum = sum + i;
}

Copy
sum (sum & obj) {
n = obj.n;
Sum = obj.sum; }

```
cout << "\n Student Details :\n";
 S.display ();
 return 0;
}
```

Q3) Accept Data for two object of class & display the Data

→
```
#include <iostream>
using namespace std;
class collage {
  int roll_no;
  String name, course;
public:
 collage (int r, string n) {
  roll_no = r;
  name = n;
 course = " COMPUTER ENGINEERING";
}
Void display (){
   cout << "\n Roll No:" << roll_no << endl;
  cout << "Name :" << name << endl;
  cout << " course:" << course << endl;
   }
};
int main (){
   Collage S1.(5. "Parth");
    Collage S2(6 "Aditya");
    S1.deisplay C);
    S2. display ();
  return 0;
  }
```

d) Write a program to demonstrate constructor Over loading

→
```
#Include <iostream>
using namespace std;
class collage {
  int roll_no;
  String name, course;
 public:
   Collage C){
       roll_no = 4;
      name = "Parth";
      course = "Computer Engineering";
 }
 Collage (int r, string n, string c){
 roll_no = r;
 name = n;
 course = c;
 }
 Void display (){
 cout<<" Roll No: " << roll_no<<endl;
 cout <<" Name:" <<name <<endl;
 cout <<" course :" <<course <<endl;
    }
 };
int main ();
 ~~cout << endl~~; collage s;
 ~~S1.~~dis          collage s1(5, "Aditya", " Civil");

 ~~S1.d~~ ~~S1.~~ S.display ();
          cout <<endl;
           S1.display
```

Exp-6

Q1]

→
```
#Include <iostreams
using namespace std;
class Person
protected
 string name;
int age;
};
Class student : Protected person
{
Private
 int roll_no;
Public:
void accept(){
Cout << Enter age:";
Cin >> age;
Cout<<" Enter the name;",
Cout<<" Enter the roll no of student:";
Cin >> roll_no;
}
void display()
{
Cout<<"Name:"<<name<<endl;
Cout<<" Age:"<<age<<endl;
cout<<" Roll numbers:<<roll_no<<endl;
}
};
```

Exp - 6

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

class person{
protected
  String name;
  int age;
Public:
 void getinfo(){
 Cout <<" Enter Name & Age:";
  cin >> name >> age;
 }
 Void disp (){
     cout <<"Name and age of student :<< name
          <<" " <<age <<endl;
  }
 };
Class student: Public Person{
 Protected:
  int roll;
 Public:
 Void acc (){
  cout <<" Enter roll number:";
  cin >> roll;
  }
 void displayRoll(){
 Cout <<" Roll number :" <<roll <<endl;
    }
 };
```

```
Class Academics {
protected
 int m1, m2;
Public:
void acceptMarks (){
Cout <<" Enter marks for m1 and m2; ";
 Cin >> m1 >> m2;
 }
 void displayMarks () {
  Cout <<" marks:" <<m1 <<" " << m2 <<endl;
    }
 };
 Class sports {
 Protected:
  int sports_score;
 public:
  void getSports (){
  Cout <<" Enter sports score:";
  Cin >> sports_score;
  }
 Void showSports (){
  Cout <<" sports Marks:" <<Sports_score <<endl;
    }
  };
 Class Result: Public student, Public Academics,
   public Sports {
 Public:
  Void displayResult (){
   int total = m1 + m2 + sports_score;
   Cout << " Total: " << total <<endl;
  }
 };
```

Date:

```
int main (){
Result r;
 r.getinfo ();
 r . disp ();
 r. acc ();
r. displayRoll ();
r.acceptMarks();
r. displayMarks ();
r.getSports ();
r.showsports ();
r. displayResult();
}
```

Q2]

→
```
#Include <iostream>
using name space std;
class Academic {
protected:
int marks;
};
class Sports{
protected:
int Spt_Score;
};
class result: protected Academic,
 protected Sports{
private:
float per,
Public:
Void accept (){
Cout<<" Enter the marks:"
Cin >> marks
Cout <<" Enter sports Score:";
Cin >> Spt_Score;
}
Void calculate ()
{
int total = marks + Spt_Score;
per = (total/200.0)* 100;
Cout << "Result =" << per <<endl;
  }
};
int main (){
Result r;
r.accept();
```

```
r.calculate();
}
```

Q3)

→

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

class Vehicle{
protected:
string brand;
String model;

Public:
Void aceept Vehicle(){
Cout <<" Enter the brand:";
cin >> brand
cout<<"Enter Vehicle model:";
cin >> model;
}
Void display vehicle(){
cout <<" Brand:" << brand << endl;
cout <<" model: << model << endl;
}
};
class Car :Public Vehicle{
Protected
String type;
Public:
Void accept Car(){
accept Vehicle();
```

```
cout << " Enter car type (suv.etc): ";
cin >> type; }
void display car (){
cout << " Type: " << type << endl; }};

Class Electric car: Public car {
Private: float battery capacity;
Public:
void accept Electric car (){
Cout << " Enter battery capacity: ";
cin >> battery capacity; }
void display Electric car (){
 cout << " Electric car details ";
display car ();
Cout << " Battery capacity : " Battery capacity
<< " kwn " << endl;
 }
    };
int main (){
Electric car el;
el. accept Electric car ();
el. display Electric car ();
return 0;
 }
```

Q4)

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

class Employee {
 protected
int emp_id;
String name;
};
class Manager : Public Employee{
Private:
 String dept;
Public:
Void accept C){
cout <<" Enter emp ID:";
cin >> emp_id;
cout <<" Department:" ;
cin >> dept;
}
void display (){
cout << " employe Id &name:" <<
 emp_id <<" " << name <<endl;
cout <<" Department : " <<dept <<endl;
  }
 };
class Developer: protected Employee{
Private:
 String pro_lang;
Public:
```

P.T.O

```
Void accept (){
cout <<" Enter the programming lang: ";
cin >> pro_lang;
  }
Void display (){
Cout <<" Programming language:"
  << Pro_lang;
    }
  };
int main (){
 manager m1
m1. accept ();
m . display ();

developer d1;
 d1 . accept ();
 d1 . display ();
}
```

M T W T F S S
Page No.:
Date:
YOUVA

✱ virtual base class

→
```
#Include <iostream)
using namespace std;
class clg - student
{
protected
string clg code;
{};
class test: virtual public clg - student
{
protected:
int - score;
};
class sports: virtual public clg - student
{
protected;
int score;
};
class Result: Projected test, sports
{
Public:
Void accept ()
 {
 cout << clg code =" << clgcode;
 cout << "\n percentage =" << perc;
 cout << "\n sports score =" << s_score;
 }
};
 int main()
{
  result r;
```

```
r.accept();
r.display();
}
```

28/9/23

Exp-7 (Demonstrate compile time polymorphism)

Q1) WAP using Function over loading

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

Class Area {
Public:
int calculate (int length, int breadth){
return length * Breadth;
 }
int calculate (int side){
return side * side;
 }
};
int main (){
Area obj;
int length, Breadth, side;
cout << " Enter length and breadth of laboratory
(Rectangle): ";
cin >> length >> Breadth;
cout <<" Area of laboratory (Rectangle) = "<<
obj.calculate (length, breadth) << endl;
cout << "Enter side of classroom (square): ";
cin >> side;
cout << " Area of classroom (square) = " << obj.
calculate (side ) << endl;
return 0;
}
```

Q2)

→
```
#Include <iostream>
class sum1 {
public:
 int i;
Void sum (float a[5]){
float s=0;
for (i=0; i<5;i++){
s +=a[i];
}
cout <<" Sum of 5 float numbers: "<<s<<endl;
}
void sum(int b[10]){
 int s=0;
 for (i=0, i<10;i++){
  s += b[i];
};
int main (){
 Sum1 s1;
float c[5];
int d[10];
cout <<" Enter 5 float numbers:\n";
for (int i=0; i<5;i++){
 cin>> c[i];
}
Cout <<" Enter 10 integer numbers:\n";
for (int i=0; i<10; i++){
 cin >>d[i];
}
 s1. sum(c);
 s1. sum(d);
 return 0;
}
```

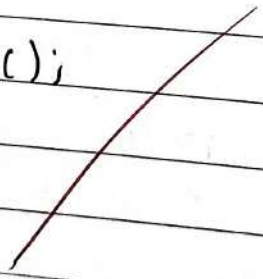
3)

```
#Include <iostream>
class num {
  int a;
public;
Void accept (){
 (out <<" Enter Value of a:";
 (in >> a;
}
Void disp(){
Cout << " Value of a:" <<a;
}
void operator -(){
a=-a;
}
};
int main(){
num n1;
 n1.accept();
-n1;
n1.display();
}
```

4)

→

```
#Include <iostream>
using namespace std;
class num {
   int a, b, c;
Public:
Void accept (){
 Cout << "Enter Value a:";
  (in >>a >> b;
}
Void display (){
 Cout << "Value of a:" <<a;
}
Void operator ++ (){
 a = ++a;
  }
};
int main (){
 num n1;
 n1. accept ();
 ++n1;
 n1. display ();
}
```



Experiment No. 8

→

```
#Include <iostream>
using namespace std;
class mstring {
  string str;
public:
  mString (string s){
    str = s;
  }
  mstring (){
    str = " ";
  }
  Void operator + (mstring obj){
    str = str + obj.str;
  }
  void display (){
    cout << str;
  }
};
int main (){
  mstring s1 ("x y z"), s2 ("pqr"), s3;
  s1+s2;
  cout << "concatenated String:";
  s3 = s1;
  s3.display();
  }
```

2

→
```
#include <iostream>
using namespace std;
class ilogin {
  protected
   string name;
   String pass;
   public
   virtual void accept() {
   cout << " Enter Name and password
   cin >> name >> pass;
   }
   Virtual void display (){
    cout << " Name " << name << " password
     << pass;
   }
   };
  class elogin : Public login {
   string email;
   string pass;
Public:
   Void accept () {
   cout << " Enter Email & password: ";
   cin >> email >> pass;
   }
   Void display () {
   cout << " Email " << email << " Password"
    << Pass;
  } };
  class mlong : public ilogin {
   String mid;
   String pass;
```

```
Public:
void accept (){
   cout << " Enter  M_id  and password";
  cin >> mid >> pass;
}
cout << " M-id " << mid << "password " << pass;}
 };
int main (){
ilogin * iptr
elogin i;
elogin -e;
mlogin m;
iptr = i;
iptr -> accept();
iptr -> disp();
cout << endl;
iptr = &e;
iptr -> disp();
cout << endl;

iptr = &m;
iptr -> accept();
iptr -> disp();
  cout << endl;
  return 0;
  }
```

Q

5/11

Exp-9

→
```
#Include <iostream>
#Includ <fstream>
using namespace std;
int main(){
 ifstream fin;
 ofstream fout;

fout.opean("destination.txt");
fin.open("Source.txt");
if(!fin){
cout<<"error opeaning source file..\n";
return 1;
}
Char ch;
 while ( fin.get(ch)){
 fout.put(ch);
}
 fin.close();
fout.close();
cout<<"File opened successfully\n";
fin.open("Source.txt");
int word count=0;
String word;
while(fin>>word){
   word cout++;
}
cout<<"word count;"<<wordcount<<"\n";
fin.close();
```

copy s from one file to another

```
fin.opean("Source.txt");
String target;
int count=0;
while(fin>>word){
if(word==target){
  cout++;
  }
}
cout<<"Enter target"<<endl;
cin>>target;
cout<<"word occurance"<<count<<endl;
fin.close();

fin.opean(Source.txt");
int ~~dis~~ digit count;
int space count;

while(fin.get(ch)){
  If(isdigit(ch)){
   digitcount++;
}
  If  (isspace(ch)){
    space count++;
}
If (isspace(ch)){
    space count ++;
}}
cout<<"Digit:"<<digit count<<"\n";
count<<"Space:"<<space out<<"\n";
return 0;
}
```

counts the number of digits & space in files

counts occurence given word of file

```
int main (){

Polymorphism
    login * login;
    {mailLogin e;
    Membership login m;

    login = &e;
    login -> accept();
    login -> display();

    login = &m;
    login -> accept();
    login -> display();
    return 0;
    }
```

Q1

S11

Exp 10.

Q1)

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

template <class T>
class A{
Public :
T m1, m2;

Void accept (){
cout << " Enter the first number :",
cin >> m1;
cout << " Enter the second number :",
cin >> m2;
}
Void calc (){
int choice :
cout << "\n -- Simple Calculator Menu --\n";
cout << " 1. addition \n",
cout << " 2. Multiplication \n",
cout << " 3. division \n";
cout << " 4. Subtraction \n",
cout << " 5. Square root \n",
cout << " 6. Percentge (m1 is what % m2) \n";
cout << " 7. Power (Square of Both numbers)
\n";
cout << " 8. Trigonometric (sin values) \n";
```

```
cout << " Enter your choice :";
cin >> choice;

Switch (choice) {
case 1:
cout << " Sum =" << m1 + m2 << endl;
Bre Break;
case 2:
cout << " Multiplication =" << m1 * m2 << endl;
Break;
case 3:
 cout << " Division =" << (m1 / m2) << endl;
Break;
Case 4:
cout << " Subtraction =" << m1 - m2 << endl;
Break;
case 5:
cout << "Square root of " << m1 << " ="
<< sqrt (m1) << endl;
cout << "Square root of " << m2 << " ="
<< Sqrt (m2) << endl;
Break;
Case 6:
cout << m1 << "is" << ((m1 * 100.0) / m2)
<< " % of " << m2 << endl;
Break;
Case 7;
cout << " Square of " << m1 << " = " << pow
  (m1, 2) << endl;
cout << " Square of " << " = " << pow (m2, 2)
 << endl;
Break;
```

```
Case 8:
cout << "Sin (" <<m1 <<") =" << sin(m1)
<< endl;
cout << "sin (" <<m2<< ") = " << sin(m2)
 << endl;
Break;

default:
cout << "Invalid choice!" <<endl;
  }
}

int main () {
A < double> obj;
obj.accept ();
obj.calc ();
return 0;
}
```

11) Exp 11

→
```
#include <iostream>
#include <vector>
#include <cctype>
using namespace std;
int main (){
vector <int> vec (5);
int i;
cout << " Enter Vector 5 Elements: ";
for (i=0; i<5; i++){
cin >> vec[i];
}
cout <<endl;
cout <<" Vector Elements
are: " <<endl;
for (i=0; i<5; i++){
cout<<vec[i]<<endl;
}
cout<<" Modified Elements are: ";
for (i=0; i<5; i++){
vec[i] = vec[i] + i*2;
}
for (i=0; i<5; i++){
cout << vec[i]<<" ";
}
cout<<endl;
int Scalar;
cout << " Enter a Scalar Value to multiply: ";
cin >> Scalar;
cout << " After Multiplying by Scalar: ";
```

```
for (i=0; i<5; i++)
vec[i] = vec[i] * scalar;

for (i=0; i<5; i++)
 cout << vec[i] << "  ";
 cout << endl;
```

12

1)

```
#Include <iostream>
#Include <cstring>
#include <stack>
using namespace std;
template <class T>
class stack{
 Stack <T> my_stack;
Public:
Void accept (){
int size;
cout << " Enter the size of the stack " <<endl;
cin >>size;
T data;
for (int i=0; i<size; i++){
cout << " Enter the data for the stack "endl;
cin >> data;
my_stack. push (data);
}
}
void pop (){
T a;
cout <<" Enter the value which you want
to delete \n" <<endl;
  cin >>a;
if (my_stack. empty ()){
  cout << " the stack is
  empty " <<endl;
}
if (!my Stack. empty ()){
```

```
cout <<"the stack is empty" <<endl;
 }
 if(!my_stack.empty()){
  my_stack.pop(x);
 }
}

void display(){
 if(my_stack.empty()){
   cout <<" The stack is empty" <<endl;
 }

 while(!my_stack.empty()){
   cout << my_stack.top();
  my_stack.pop();
 }
 }
}

int main(void){
 stack <int> k;
 k.accept();
 k.display();
}
```

2) Queue

→

Create a Queue

```
#include <iostream>
#inclu <queue>
using namespace std;
template <class T>
class Queue {
Private:
 Queue < T > q;
Public:
Void Enqueue (T Element){
Q. Push (Element);
Cout << element <<" Enqueued Successfully\n"
}
 Void dequeue (){
If (q. Empty ()){
Cout <<" Queue is Empty cannot dequeue
\n";
} else {
 cout << q. front () <<" dequeued Successfully
!" n"
   q.pop ();
}}
Void display (){
cout <<" Queue Elements :\n";
Queue < T > temp = q;
while (! temp .Empty ()){
cout << temp . front () <<" ";
temp. ppp ();
```

```
}
cout << endl;
}};
int main(){
Queue <int> mq;
mq.Enqueue(30);
mq.Enque(20);
mq.Enque(10);

mq.display();

mq.dequeue();
mq.display();

return 0;
}
```