

Explain what is meant by memory leak in C++.

- 1) Write a Cpp program to declare a class - student having data members as roll no and name. Accept and display data for single data.]

```
# include <iostream.h>
using namespace;
```

```
class student {
```

```
    int roll-no;
```

```
    String name;
```

```
Public:
```

```
    void accept () {
```

```
        cout << "Enter roll no. of student and Name: " ;
```

```
        cin >> roll_no >> name;
```

```
}
```

```
    void display () {
```

```
int main () {
```

```
    Student S1;
```

```
    S1.accept ();
```

```
    S1.display ();
```

```
cout << RollNo: " << roll_no << ", Name: " << name << endl;
```

```
return 0;
```

```
003}, 003, return printf
```

Our Output

003

Enter roll no. and name: 20 shraddha

Roll no: 20 , Name: shraddha.

OOP = Object

2) Write a Cpp program to create a class Book having data members as bname, bprice ,bpages. Accept the data for two books and display the name of book having greater price.

→ # include <iostream.h>

using namespace :

Class Books

```
{
    Public
        bname ;
        bprice ;
        bpages ;
    void accept ()
}
```

cout << "Enter the bname of the book ; bprice and bpage ; "

cin >> bname > bprice > bpage;

```
}
void main ()
{
```

Book b1 , b2 ;
b1 accept ();
b2 accept ();

if (b1 . price > b2 . price

{ b1 . disp ();

else
 b2 disp ();

Output : ()

Enter book name , price , no. of pages

Harry Potter , 900 , 200

" " " " " " " :

Sherlock Holmes 700
600

Book name : Harry Potter
price = ₹ 900
pages = 900



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

3) Write a program to declare 'class "Time"'; accept time in HH:MM:SS format, convert it into total seconds and display them.

→ # include <iostream>
using namespace std;

```
class time sec {  
    int h,m,s;  
    char c;  
public:  
    void getTime () {  
        cout << "Enter time in hh:mm:ss format: ";  
        cin >> h >> c >> m >> c >> s;  
    }  
    void convert to seconds () {  
        int total seconds = h * 3600 + m * 60 + s;  
        cout << Total seconds << endl;  
    }  
int main () {  
    timesec t;  
    t.getTime ();  
    t.convert to seconds ();  
    return 0;  
}.
```

Q1
15/1

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

Exp 2:

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

```

→ include <iostream>
include <string.h>
using namespace std;

class city {
public:
    string name;
    int population;

    void accept() {
        cout << "Enter name and population";
        cin >> name >> population;
    }

    void display() {
        cout << "Largest population" << "Name: " <<
            name << endl << "population:
                           " << population;
    }
};

int main() {
    city c[5], lp;
    lp = c[0];
    for (int i=0 ; i<5 ; i++) {
        c[i].accept();
        if (c[i].population > lp.population) {
            lp = c[i];
        }
    }
}

```

```

    lP. display();
    return 0;
}

```

- 2) WAP to declare a class 'Account' having data members as Account no. and balance. Accept this data for 10 accounts and give interest at 10%. where balance is equal or greater than 5000 and display them.

```

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

class account {
public:
    string acc_no;
    int balance;

    void accept() {
        cout << "Enter account number: ";
        cin >> acc_no;
        cout << "Enter balance: ";
        cin >> balance;
    }

    void display() {
        if(balance >= 5000) {
            int I=0.1 * balance;
            cout << endl << "Account number: " << acc_no << endl <<
                "Since balance is greater than 5000, you got
                interest," << endl << "Original balance: " << balance
                << endl << "Interest: " << endl << "Total: " << balance
                + I;
        }
    }
};

```

} else {

cout << endl << "Account number : " << acc_no
 << endl << "Balance : " << balance;

}

{ Input account 'account' into a[10] at index 0 to 9.

}; Input amount from user to account 10 to 19.

int main () { Input Name of rot file with

account a[10]; Input account number 10 to 19 to

for (int i=0 ; i<10 ; i++) { Input Name of rot file with
 a[i].accept ();

cout << endl ;

}

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

a[i].display ();

cout << endl ;

}

return 0;

}

3)

8/07/25

```

#include <iostream>
#include <string>
using namespace std;
class staff {
public:
    string name;
    string post;
    void accept() {
        cout << "Enter Name: ";
        cin >> name;
        cout << "Enter Post: ";
        cin >> post;
    }
    void display() {
        if ((post == "HOD") || (post == "HoD") || (post == "HOD"))
            cout << name << " is and " << post;
    }
};

int main() {
    Staff s[5];
    for (int i=0; i<5; i++) {
        s[i].accept();
    }
    for (int i=0; i<5; i++) {
        s[i].display();
    }
    return 0;
}

```

Q
29/7/25

Exp 8

1)

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

```
class book {
```

```
    string book_name;
    string author_name;
    int book_price;
```

```
public :
```

```
void accept () {
```

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

```
    getline (cin, this->book_name);
```

```
    cout << "Enter author-name:";
```

```
    getline (cin, this->author_name);
```

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

```
    cin >> book_price;
```

```
}
```

```
void display () {
```

```
    cout << "In Name of book :" <<  
        book_name;
```

```
    cout << "In Name of author :" <<  
        author_name;
```

```
    cout << "In Book Price :" << book_price;
```

2) *o* *нину*

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

class STUDENT {

int rollno;

float percentage;

public

void accept(C) throws IOException

```
cout << "Enter student roll no : "
```

`cin >> this` → roll no; < n10

```
cout << "Enter student percentage : ";
```

cin >> this → percentage;

3

:(p/n9>> m>>"m Dn9M">>f(0))

if bus >> sm >> " " , m "LNAME" >> fnum

```

void display () {
    cout << "In Student roll no. " << roll_no;
    cout << "In Student percentage: " << percentage;
}

int main () {
    STUDENT s1;
    s1.accept ();
    s1.display ();
}

return 0;
}

```

3) Nested class

→ # include <iostream>

a. using namespace std;

```

class student {
public:
    class marks {
public:
    int m1, m2;
    void accept () {
        cout << "Enter marks for m1: ";
        cin >> m1;
        cout << "Enter marks for m2: ";
        cin >> m2;
    }
    void display () {
        cout << "Marks m1: " << m1 << endl;
        cout << "Marks m2: " << m2 << endl;
    }
}

```

int main () {

Student :: marks m;

m.accept();

m.display();

return 0;

}

=> O/P

~~Q1~~ Enter marks for m₁: 89

Enter marks for m₂: 84

marks m₁: 89

marks m₂: 84

Qn

518125

? () num. thi

: ch., n. moduli

" " failed to open file" >> fno

: ch., n

" " failed to open file" >> fno"

() qulov top - N

Exp: 4

Q.1)

```

→ #include <iostream>
using namespace std;
class Number {
    int value;
public:
    void getvalue() {
        cout << "Enter value: ";
        cin >> value;
    }
    void display() {
        cout << value << endl;
    }
    void swapvalue(Number obj) {
        int temp = value;
        value = obj.value;
        obj.value = temp;
    }
};

int main() {
    Number n1, n2;
    cout << "Enter the first value of object: ";
    n1.getvalue();
    cout << "Enter first value for object: ";
    n2.getvalue();
}

```

cout << "Enter value for second object : ";
 n₂ . get value ();

cout << "In Before swapping " << endl;
 cout << "n₁ = "; n₁ . display ();
 cout << "n₂ = "; n₂ . display ();
 n₁ . swap values (n₂);

cout << "After swapping : " << endl;
 cout << "n₁ = "; n₁ . display ();
 cout << "n₂ = "; n₂ . display ();
 return 0;

O/P

(1) 97th b7ov

Enter value for first object:
 enter value : 5

Enter value for second object:

enter value: 8

Before swapping : n₁ = 5

n₂ = 8

After : n₁ = 8

Swapping : n₂ = 5

2) "The class having self call by name" \Rightarrow two
 (i)友元函数
 (ii)成员函数

\rightarrow #include <iostream>

using namespace std;
 class number {

private:
 int n, n2;

public:
 void accept()

{
 cout << "enter two numbers ";
 cin >> n >> n2;
 }

void disp ()

cout << "n1 = " << n1 << endl;

cout << "n2 = " << n2;

friend void swap (Number &n)

int temp = n.n1;

n.n1 = n.n2;

n.n2 = temp;

}

int main () {

Number m;

m.accept();

cout << "Before swap : ";

m.display();

swap (m);

cout << "After swap : ";

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

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

O/P

Enter 2 numbers : 2

3

Before swap: $n_1 = 2$ $n_2 = 3$

$n_1 = 3$

After swap: $n_1 = 3$

$n_2 = 2$

"minimum swap value" \rightarrow 1 step
if $n_1 < n_2$

? () with break

>>> qif(n1 >= n2) " " if

if break >> 1 step

(break & 1 step) instead your logic break
already 1.

? () with while

: moving,

SMIN + 1

- 1 value

? () with break

"minimum swap value" \rightarrow 1 step

- SMIN < 0

3)

→ #include <iostream>
using namespace std;

class class 2;

class class 1;

private :

int num1;

Public

Void accept () {

cout << "Enter first number:";

cin >> num1;

}

void disp () {

cout << "class Value :" <<

num1 << endl;

}

friend void swap Numbers (class 1 & class 2)

};

class class 2 {

private:

int num2;

Public:

Void accept () {

cout << "Enter second number :" ;

cin >> num2;

}

```

void display () {
    cout << "class 2 value: " << num2 << endl;
}

friend void swap Numbers (class 1 & a class 2 & b);
};

void swap Numbers ((class 1 & a . class 2 & b) {
    int temp = a.num1;
    a.num1 = b.num2;
    b.num2 = temp;
}

int main () {
    class 1 A;
    class 2 B;

    A.accept ();
    B.accept ();

    cout << "Before swapping: ";
    A.display ();
    B.display ();

    Swap numbers (A,B);

    cout << "In After swap: \n";
    A.display ();
    B.display ();

    return 0;
}

```

O/P
 enter first number: 5
 " Second .. : 6

S	E	I	T	W	T	F	S
AVUVY	9/2/2019						

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

Before swap:

class 1 value: 5 class 2 value: 6

After swap

class 1 value: 6 class 2 value: 5

class 2 value: 6 class 1 value: 5

swap = swap

swap = swap

A = 200

B = 200

A = 200

B = 200

Unit 10

(A) public A

(B) public B

(A, B) public (A, B)

(A) public A

(B) public B

Q number

4)

A () Moon for

(M) Moon

for S Moon

(P) Moon

→ #include <iostream>

using namespace std;

class Result1 {

Private :

float marks1;

Public :

void accept ()

cout << "Enter marks for first student:";

cin >> marks1;

}

friend float Average (Result1, Result2);

};

class Result2 {

Private :

float mark2;

Public :

void accept ()

cout << "Enter marks for second student:";

cin >> mark2;

};

friend float Average (Result1, Result2);

};

float Average (Result1 a, Result2 b);

{

float average = (a. marks1 + b. marks2) / 2;

return average;

}

```

int main () {
    Result r1;
    Result r2;
    float avg;
    r1.accept();
    r2.accept();
}

```

```

avg = Average (r1, r2);
cout << "Average of two results is : " << endl;
return 0;
}

```

OIP

Enter marks for first student : 70

enter marks for second student : 60

Avg of two results : 65

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

5)

→ #include <iostream>

using namespace std;

class Number 2;

class Number 1 {

private :

int num1;

public :

void accept () {

cout << "enter number:";

cin >> num1;

}

friend void find greatest (Number 1,

Number 2)

};

Class Number 2 {

private :

int num2;

public :

void accept () {

cout << "enter number:";

cin >> num2;

,

friend void find greatest (Num 1, Num 2);

};

void find greatest (Number 1 a, Number 2 b)

{

```

if (a.num1 > b.num2) {
    cout << "greatest number is : " << a.number;
} else if (b.num2 > a.num1)
{
    cout << "greatest number is : " << b.number;
} else {
    cout << "Both are equal " << endl;
}

int main () {
    Number1 obj1;
    Number2 obj2;
    obj1.accept ();
    obj2.accept ();
    find_greatest (obj1, obj2);
    return 0;
}

```

O/P

```

Enter number = 8
Enter number = 7
Greatest number is : 8

```

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

Q.6)

```

→ #include <iostream>
using namespace std;
class class B;
class class A {
    private:
        int valueA
    public:
        void input () {
            cout << "enter value for class A";
            cin >> valueA;
        }
        friend int sum (class A a , class B b);
};

class class B {
    private:
        int valueB;
    Public:
        void input () {
            cout << "Enter value for class B: ";
            cin >> valueB;
        }
        friend int sum (class A a , Class B b);
};

int sum(class A a , class B b)
{
    return a.valueA + b.valueB;
}

```

```

int main () {
    class A objA;
    class B objB;
    objA::Input();
    objB::input();
    cout << "sum:" << sum(objA, objB) << endl;
    return 0;
}
    
```

O/P

Enter value for class A:3

Enter value for class B:5

sum:8

-1)

```

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

```
class Number {
```

```
private:
```

```
    int value;
```

```
Public:
```

```
void accept() {
```

```
    cin >> value;
```

```
}
```

```
void display() {
```

```
    cout << value << endl;
```

```
}
```

```
friend void swap(Numbers n1, Numbers n2);
```

```
}
```

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

```
void swapNumbers (Number n1, Number n2)
```

```
int temp = n1.value;
```

```
n1.value = n2.value;
```

```
n2.value = temp;
```

```
int main () {
```

```
Number num1, num2;
```

```
cout << "Enter value for num1: ";
```

```
num1.accept ();
```

```
cout << "Enter value for num2: ";
```

```
num2.accept ();
```

```
cout << "Before swapping: " <endl;
```

```
cout << "num1 = ";
```

```
num1.display ();
```

```
cout << "num2 = ";
```

```
num2.display ();
```

```
swapNumbers (num1, num2);
```

```
cout << "After swapping: " <endl;
```

```
cout << "num1 = ";
```

```
num1.display ();
```

```
cout << "num2 = ";
```

```
num2.display ();
```

```
}
```

O/P

Enter value for num1 : 5

Enter value for num2 : 3

Before swapping

num1 = 5

num2 = 3

After swapping

num1 = 3

num2 = 5

8) Define 2 classes (a) rectangle (b) square

→ #include <iostream>

using namespace std;

class cube;

class box {

private:

int volume;

Public:

void accept()

{

cout << "Enter the volume of box: ";

cin >> volume;

friend void findgreater (box, cube);

};

class cube {

private:

int volume;

Public:

void accept()

{

cout << "Enter the volume of box: ";

cin >> volume;

}

friend void findgreater (box, cube);

};

class cube {

private:

int volume;

Public:

void accept()

```
{  
    cout << "enter the volume of cube : ";  
    cin >> volume;  
}
```

```
}; friend void findgreater(box, cube);
```

```
{ void findgreater (box b, cube c);
```

```
    if (b.volume > c.volume) {  
        cout << "volume of box is greater";  
    }
```

```
    else if (b.volume == c.volume) {  
        cout << "both volumes are equal";  
    }
```

```
else {  
    cout << "volume of cube is greater";  
}
```

```
int main () {
```

```
    box h;
```

```
    cube c;
```

```
    h.accept ();
```

```
    c.accept ();
```

```
    findgreater (&h, &c);
```

```
    return 0;
```

BIP

Enter volume of box : 67
" " " cube = 45

Volume of box is greater.

9)

" : C++
Arith in complex num " is fun
: complex << n;

→ #include <iostream>

using namespace std;

class complex { .. . d add) unop op bin
private:

float real;

float imaginary;

public: n o add to complex " >> two

void accept () {

cout << " enter real part: " ;

cin >> real;

cout << " enter imaginary part: " ;

cin >> imaginary;

}

void display () {

cout << real << " + " << imaginary << " i "

<< endl;

}

friend complex addcomplex (complex c1,
complex c2);

};

Complex addcomplex (complex c1, complex c2) {
complex temp;

temp.real = c1.real + c2.real;

temp.imaginary = c1.imaginary + c2.imaginary;
return temp;

}

DA : add to complex sum
Z1 = 3+4i Z2 = 5+6i
temp is add to sum

S	M	T	W	T	F	S	S
MONDAY							

Page No.:	YOUVA
Date:	

```

int main () {
    complex num1, num2, sum;
    cout << "enter first complex number: " <<
        num1.accept();
    cout << "enter second complex number: " <<
        num2.accept();
    sum = addComplex (num1, num2);
    cout << "sum of complex number: ";
    sum.display();
    return 0;
}
  
```

O/P

Enter first complex number:

enter real part: 7 + 0i

enter imaginary part: 5

Enter second complex number: A + Bi

enter real part: 3

enter imaginary part: 9i

Sum of complex numbers = 10 + 4i

(112) program solution

QUESTION

(10)

```

→ #include <iostream>
using namespace std;
class student {
private:
    string name;
    float mark1, mark2, mark3;
public:
    void accept() {
        cout << "enter student name: ";
        cin >> name;
        cout << "enter marks in three subjects ";
        cin >> mark1 >> mark2 >> mark3;
    }
    friend void calculate Average (student);
};

void calculate Average (student)
{
    float avg = (5 * mark1 + 5 * mark2 + 5 * mark3) / 3
    cout << "Student Name: " << s.name << endl;
    cout << "Average marks: " << Avg << endl;
}

int main()
{
    student stu;
    stu.accept();
    calculate Average (stu);
    return 0;
}

```

S	E	T	W	T	F	S
AUGUST	1	2	3	4	5	6
	7	8	9	10	11	12

Page No.:	YOUVA
Date:	

O/P

enter student name: shraddha

enter marks in three subjects:

78

92

89

student name: shraddha

Average marks : 86.33

(2) 11)

9/0

→ #include <iostream>

using namespace std;

Class Beta;

Class Gamma;

Class Alpha {

private:

void accept () {

cout << "Enter value for Alpha: ";

cin >> a;

}

friend void sumvalues (Alpha, Beta, Gamma);

};

Class Beta {

private

int b;

public:

void accept () {

cout << "Enter value for beta a: ";

cin >> b;

}

friend void sumvalues (Alpha, Beta, Gamma);

};

Class Gamma {

private:

int c;

public:

void accept () {

cout << "Enter value for gamma: ";

cin >> c;

}

```

friend void swapvalues(Alpha, beta, Gamma);
};

void sumvalues(Alpha, x, Beta y, gamma z) {
    int sum = x; a = y + b = z + c;
    cout << "sum of all values: " << endl;
    sum << endl;
}

int main() {
    Alpha obj A;
    Beta obj B;
    Gamma obj C;
    obj A . accept();
    obj B . accept();
    obj C . accept();
    sum values (obj A, obj B, obj C);
    return 0;
}

```

Q1 P

Enter values for alpha: 7
 Enter value for Beta: 4
 Enter values for Gamma: 0

Sum of all values = 11

(A, B) sum + 2 * C = 11
 7 + 4 + 0 = 11

11 >> reference >> 21

11 >> reference >> 21

12) (Comm) et al., 2001) *entomophaga* bien bientôt

↳ (S. ammophilus) (LBB, 1916) 900 m. 2 hirs.

→ #include <iostream>

```
#include <math>
```

using namespace std;

class point {

private

float a;

~~float y;~~

Public:

```
void accept () {
```

```
cout << "Enter coordinates (x,y) : ";
```

```
cin >> x >> y;
```

1

```
friend float calcDistance (point, point);
```

1

float calodistance (point p1, point p2)

{

~~float dx = p2.x - p1.x;~~ not useful now

~~float dy = p2.y - p1.y;~~ ~~and~~ ~~auto~~ ~~num2~~

return $\sqrt{dx^2 + dy^2}$;

~~int main () {~~

point A, B;

A. accept();

B. accept();

float distance = calcDistance(A, B);

```
cout << "The distance between the two points
```

is <<distance << endl;

return 0;

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

O/P :

Enter coordinates (x, y): 6, 7

Enter coordinates (x, y): 6, 9

The distance b/w the 2 points is: 2.8281

13)

. 910

→ #include <iostream>
using namespace std;

class Audit;

class Bank Account {

private:

double balance;

public:

void getbalance fromuser () {

cout << "Enter account balance: ";

cin >> balance;

}

friend void Audit report (Bank account, audit);
};

class audit {

public :

friend void auditreport (Bank account, audit);

};

void auditReport (Bank account, Audit)
{

cout << "audit Report": Account balance is: " <<

account balance << endl;

}

int main () {

Bank account acc;

audit auditor;

acc.getbalancefromuser ();

audit report (acc, auditor);

return 0;

}

D/P

Page No.:

Enter account balance: ₹ 677.00 at 100
Audit Report : account balance is ₹ 677.00

~~Q/H~~
15/8

< margin should be

: big enough price

MWS 22.00

: MWS . n tri

: margin:

() MWS

: MWS = 0

; " : n to 90000 " >> LWS

; n << mws

(++i ; n=5 ; i=1 tri) rot

; i = MWS

↑ () margin bias

>> MWS >> " : 5 " >> n >> " at 1 more MWS " >> LWS

Exp: 6

- a. WAP to find the sum of numbers between 1 to n using a constructor where the value of n will be passed to the constructor.

→ #include <iostream>
using namespace std;

```

class Sum
{
    int n, sum;
public:
    sum ()
    {
        sum = 0;
        cout << "Enter value of n: ";
        cin >> n;
        for (int i = 1; i <= n; i++)
        {
            sum += i;
        }
    }
    void display ()
    {
        cout << "sum from 1 to " << n << " is: " << sum <<
    }
};
```

```
{ int main()
```

```
    sum s;
```

```
    s. display
```

```
} return 0;
```

O/P

Enter value of n : 5

Sum from 1 to ~~i=~~ 5 is : 15.

input in swi

: 111011

(andor &ri) mub

111011 = 11

(0 = mub

(+i) ; nesj ; i=i +1) rot

; 1 = + mub

} () posqrib bior

>>mub>>" :ei " >> N >>" of 1 mub mub " >> mub

() nesm &ri

-> mub

(f) posqrib -&

-> mub

2) Parameterised Constructor:

→ #include <iostream>

using namespace std;

```
{ class sum
```

```
    int n, sum;
```

```
    public:
```

```
        sum (int value)
```

```
    {
```

```
        n = value;
```

```
        sum = 0;
```

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

```
            sum += i;
```

```
}
```

```
    void display () {
```

```
        cout << " sum from 1 to " << n << " is: " << sum <<
```

```
}
```

```
};
```

```
int main ()
```

```
{
```

```
    sum s,
```

```
    s.display ();
```

```
    return 0;
```

```
}
```

O/P

Output (pgt) (6)

Enter value of n: 5

Comments shown in

sum from 1 to n: 15

Explanation given

```

cout << "enter value of n:";    } must ends
cin >> number;                in tri
sum s (number);               i int tri
s. display ();                : calling
                                } (num + tri) must
                                i num = n
                                ; o = tot+at
return 0;                      }
}

```

O/P

Enter value of n: 5

sum from 1 to 5: 15

: a. do i = n
; o = tot+at

} (i++ ; a=s ; i= i +tri) rot
; i = + tot+at

} () output given

>> " : &i " >> n >> " at 1 mod num " >> sum

; long >> tot+at

} () num + tri

; numnum + tri

; " minimum o num " >> sum

; minimum << min

3) Copy constructor.

```
#include <iostream>
```

```
using namespace std;
```

```
class sum {
```

```
    int n;
```

```
    int total;
```

```
public :
```

```
    sum (int num) {
```

```
        n = num;
```

```
        total = 0;
```

```
}
```

```
    sum (const sum & job)
```

```
{
```

```
    n = job.n;
```

```
    total = 0;
```

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

```
        total += i;
```

```
}
```

```
    void display () {
```

```
        cout << "sum from 1 to " << n << " is :" <<
```

```
        total << endl;
```

```
}
```

```
};
```

```
int main () {
```

```
    int number;
```

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

```
    cin >> number;
```

```

sum obj1 (no); sum of marks in Hindi (i) (d)
sum obj2 (no); sum of marks in English (e)
obj2. display (); print marking of student in "Hindi"
return 0;
}
    
```

<mark> abulani <--
late assignment given

O/P

Enter a number = 6

Sum from 1 to 6 = 21

} marking 22.01

; sum prime

; finding last

; finding

} () marking

; "sum marking sum" => two

; sum (< n)

; "sum marking sum" => two

; finding sum {

} () marking two

; "n! ... direct marking ... n!" => two

; two => sum => "sum" => two

>> " " => finding => "finding" => two }

A) alarm tri

; 1. marking

; () marking 12

; 0 marking }

- b) (i) Write a program to declare a class "student" having data members as name and percentage.
 3) Write a constructor to initialize these data members.
 + Accept and display data for one student.

→ #include <iostream>
 using namespace std;

```

class student {
    string name;
    float percentage;
public:
    student () {
        cout << "enter student name : ";
        cin >> name;
        cout << "enter percentage : ";
        cin >> percentage;
    }

    void display () {
        cout << "In.... student Details... \n ";
        cout << "name : " << name << endl;
        cout << "percentage : " << percentage << endl;
    }
};

int main () {
    student s1;
    s1.display ();
    return 0;
}
  
```

O/P

Enter student name: Shraddha
 Enter percentage: 89.5

..... Student details

Name : shraddha

Percentage : 89.5

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2.2012

2) Parameterised

```
#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 of student : " << name <<
        "In percentage : " << percentage
        percentage << endl;
    }
};

int main () {
    student S1 ("Shraddha", 89.5);
    S1.display ();
    return 0;
}
```

OIP

Enter student name : Shraddha
 Enter percentage : 89.5

Student details.

Name : Shraddha

Percentage : 89.5

c) Define a class college members variables as roll number , name, course. WAP using constructor with default value as "computer engineering" for course. Accept this class and display the data;

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

```
class college {
```

```
private:
    int roll_number;
    string name;
    string course;
```

```
public:
```

```
college(int r, string n, string c = "computer engineering")
```

```
{
```

```
    roll_number = r;
    name = n;
    course = c;
```

```
}
```

college

(int r, string n, string c = "computer engineering")

{

roll_number = r;

name = n;

course = c;

}

void display();

cout << "Roll no. = " << rollno << endl;

cout << "Name = " << name << endl;

cout << "Course = " << course << endl;

}

};

```
int main () {
    int roll;
    string name, course;
```

```
cout << " enter roll number for student 1: ";
cin >> roll;
cout << " enter name for student 1: ";
cin >> name;
```

```
college student 1 (roll, name);
```

```
cout << " enter roll number for student 2: ";
cin >> roll;
cin.ignore();
```

```
cout << " enter name for student 2: ";
getline (cin, name);
```

```
cout << " enter course for student 2: ";
getline (cin, course);
```

```
college student 2 (roll, name, course);
cout << " In student Details: ";
student 1 . display ();
student 2 . display ();
return 0;
```

```
{1109 >> 0109 >> " = 01108 " >> 109
{1109 >> 0109 >> " = 01107 " >> 109
{1109 >> 0109 >> " = 01106 " >> 109}
```

S	S	T	W	T	F	S
Avg	90	85	80	75	70	65
Page No.:	1	2	3	4	5	6

M	T	W	T	F	S	S
Page No.:	1	2	3	4	5	6
Date:	YOUVA					

OIP

enter roll number for student 1 : 20

enter name for student 1 : shraddha

enter course for student 1 : computer
<option> engineering.

enter roll number for student 2 : 13

enter name for student 2 : sakshi

enter course for student 2 : computer engineering.

Student details:

Roll Number : 20

Name : shraddha

Course : Computer engineering.

Roll number : 13

Name : sakshi

Course : Computer engineering.

Pen

11/11

ENCLOSING OR SCOPING .

910

Ques 1) Explain protected members

Ans 1) #include <iostream> // header file for i/o

#include <string>

using namespace std;

class person { // protected members of class person

protected:
string name;
int age;

}

class student : protected person { // protected members of class student

int roll; // public members of class student

public:

void acc () { // member function of class student

cout << "Enter name and roll no: ";

cin >> name >> age >> roll;

void display () {

cout << "Name: " << name << endl <<
"age: " << age << endl <<
"roll no: " << roll;

}

int main () {

student s1;

s1.acc();

s1.display();

return 0;

}

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

==> O/P

Enter name, age and roll : Shraddha 17 20

Name: Shraddha

Age: 17

Roll no: 20

2) Multiple inheritance:

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

```
class Academics {
protected:
    int marks;
```

```
class sports {
protected:
    int sports-score;
```

```
class result : private Academics, private sports
{
public:
    result (int m, int ss) {
        marks = m;
        sports-score = ss;
    }
}
```

```
void add () {
    score = marks + sports-score;
    cout >> endl << "Marks :" << marks << endl
    << "Sports Score :" << sports-score <<
    endl << "Result :" << score;
}
```

S	S	S	T	W	F	S
A	V	U	D	V	A	M
AVUVY						
STUDY						

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

int main () {

 result = r1(80, 90);

 r1.add();

 return 0;

}

=> O/P

Marks : 80

sports_score : 90 int total = 80 + 20

Result : 170

3) Multilevel

```
#include <iostream>
```

```
using namespace std;
```

```
class vehical {
protected:
    string brand, model;
};
```

```
class car : Protected vehical {
protected:
    string type;
};
```

```
class electriccar : protected car {
int bat cap;
public:
electric car () {
cout << "Enter Brand, Model, Type
+ battery capacity of car: ";
cin >> brand >> model >> type >> bat cap;
}
```

```
void disp () {
cout << endl << "Brand: " << brand <<
endl << "Model: " << model << endl <<
"Type: " << type << endl << "Battery capacity:
" << Batcap << endl;
}
```

```
};
```

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

```

int main () {
    electric car el;
    el.disp();
    return 0;
}

```

=> O/P

Enter brand ,Model ,Type & battery capacity :

Mahindra

XUV700

SUV

69,00,000

Brand : Mahindra .

Model : XUV700 .

Type : SUV.

Price : 6900000.

4) Hierarchical

```
#include <iostream>
```

```
using namespace std;
```

```
class Employee {
```

```
protected:
```

```
    int emp ID;
```

```
    string name;
```

```
}
```

```
class Manager : private employee {
```

```
    string dept;
```

```
public:
```

```
    Manager () {
```

```
        cout << "Enter emp ID ,name &
```

```
        dept : "
```

```
        cin >> emp ID >> name >> dept ;
```

```
}
```

```
void disp () {
```

```
    cout << endl << "emp ID: " << emp ID
```

```
    << "Name: " << name << " Department: "
```

```
    << dept ;
```

```
}
```

```
};
```

```
class dev : private employee {
```

```
    string prog L;
```

```
public:
```

```
dev () {
```

```
    cout << "Enter emp ID ,name & prog
```

```
language : ";
```

```

} cin >> empID >> name >> progL;

void disp () {
    cout << endl << "Emp ID: " << empID <<
    endl << "Name: " << name << endl <<
    "Programming lang: " << progL << endl;
}

int main () {
    dev d1;
    manage m1;
    d1.disp ();
    m1.disp ();
    return 0;
}

```

=> O/P

Enter empID ,name , Prog language : 01
Shraddha

JS

Enter empID , name & dept : 02

Ziya

CS

emp ID : 11

name : Shraddha

Prog lang : JS

emp ID : 2

Name : Ziya

Dept : CS.

5) Hybrid

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

```
class person {
protected:
    string name;
    int age;
};
```

```
class stud : public person {
protected:
    int roll;
};
```

```
class Academics {
protected:
    int marks;
};
```

```
class sports {
protected:
    int sport_score;
};
```

```
class score : public stud, public Academics, public
sports {
public:
    score() {
        cout << "Enter name, roll, sports score,
marks: ";
    }
};
```

Exp: 7

1) WAP using function overloading to calculate the area of laboratory (rectangle) and area of class - room (square)

```
#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;
```

```
float length, breadth, side;
```

```
cout << "Enter length & breadth for laboratory : ";
cin >> length >> breadth;
cout << "Area of laboratory (rectangle) : " <<
a.calculate (length, breadth) << endl;
cout << "Area of classroom (square) : " <<
a.calculate (side) << endl;
return 0;
```

2) WAP using function overloading to calculate sum of 5 float values & 10 integer values.

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

class sumcalculator {
public:
    float sum (float a ,float b , float c, float d, float e);
    int sum (int a,intb, intc, intd, inte, intf, intg,
             int h, int i, int j);
};

float sum (float a ,float b , float c, float d, float e) {
    return a+b+c+d+e;
}

int sum (int a,intb, intc, intd, inte, intf, intg,
         int h, int i, int j) {
    return a+b+c+d+e+f+g+h+i+j;
}

int main () {
    sumcalculator s;
    float fsum = s.sum(1.1f, 2.2f, 3.3f, 4.4f, 5.5f);
    int sum = s.sum(1, 2, 3, 4, 5, 6, 7, 8, 9, 10);
    cout << "sum of 5 float nos: " << fsum << endl;
    cout << "sum of 10 integer nos: " << sum << endl;
    return 0;
}
```

3) WAP to implement Unary (-) operator when used with the object so that numeric data member of the class is negated.

```
#include <iostream>
using namespace std;
class number
{
    int a;
public:
    void accept()
    {
        cout << "a = ";
        cin >> a;
    }
}
```

```
void display()
{
    cout << "a = " << a << endl;
}
```

```
void operator -()
{
    a = -a;
}
```

```
int main()
```

```
number n1;
```

```
n1.accept();
```

```
- n1;
```

```
n1.display();
```

```
return 0;
```

```
}
```

P

111

Exp 8:

a) Way to overload the '+' operator so that two strings can be concatenated, eg. "xyz" + "pqr" then o/p will be "xypyqr".

→ class mystring
 def __init__(self, text):
 self.text = text.

def __add__(self, other):
 return mystring(self.text + other.text)

def __str__(self):
 return self.text

s1 = mystring("xyz")
 s2 = mystring("pqr")

result = s1 + s2
 print(result)

O/P

xyzpqr

2) WAP to create a base class Ilogin having data members names and password. Declare accept function virtual. Derive email login and membership login classes from Ilogin. Display email login details and membership login details of the employee.

```
# include <iostream>
# include <string>
```

```
class Ilogin {
```

```
protected:
```

```
std::string name;
```

```
std::string password;
```

```
public:
```

```
virtual void accept () = 0;
```

```
virtual void display () = 0;
```

```
}
```

O/P

Enter name = shraddha

Enter password = 0912

Name = shraddha

Password = 0912

Ques

P/I/I

Exp 9:

i) WAP to copy the contents of one file into another.

→ #include <iostream>

#include <fstream>

#include <string>

int main () {

std::ifstream input_file("first.txt", std::ios::in);

std::ofstream output_file("second.txt", std::ios::in | std::ios::out);

if (!inputfile.is_open() || !outputfile.is_open()) {

std::cerr << "error opening files." << std::endl;

return 1;

}

std::string line;

std::cout << "copying contents from first.txt to
second.txt\n\n";

while (std::getline(inputfile, line)) {

 outputfile << line << std::endl;

 std::cout << line << std::endl;

 Inputfile.close();

 Outputfile.close();

BASIC INFORMATION	
NAME	ROLL NO.
AVUVYA	9102

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

std:: cout << "In copy complete. content written to second. text\n";

```
    return 0;
}
```

2) WAP to count digits and spaces using file handling.

→ # include <iostream>

include <fstream>

int main () {

std :: fstream file ("input . text");

if (!file . is - open ()) {

std :: cerr << "error : could not open
the file ." << std :: endl;

return 1;

}

char ch;

int digit count = 0;

int space count = 0;

while (file . get (ch)) {

if (ch >= '0' + d ch <= '9') {

} digit count ++;

else if (ch == ' ') {

space count ++;

}

}

```

file.close();

std::cout << "Number of digits: "
<< digitcount << std::endl;

std::cout << "Number of spaces: "
<< spacecount << std::endl;

return 0;
}

```

O/P

Number of digit = 5

Number of space = 8

3) Write a C++ program to count words using file handling?

```
#include <iostream>
#include <fstream>
#include <string>
int main() {
    std::ifstream file ("input.txt");
    if (!file.is_open ()) {
        std::cerr << "error : could not open the
        file.", << std::endl;
    }
    return 1;
}
```

```
std::string word;
int wordcount = 0
while (file >> word) {
```

```
    wordcount++;
}
```

```
file.close();
```

```
std::cout << "total number of words : " << word
count << std::endl;
```

```
return 0;
}
```

O/P

Total number of words = nine (9).

Q
nlu

Exep 10: Write a program to compute sum of elements of array using function template (eg. pass integer, float and double array of 10 elements).

- Q) WAP to find sum of array elements using function template (eg. pass integer, float and double array of 10 elements).

#<iostream>

```
using namespace std;
template <typename t>
t findsum (t arr[], int size){
```

```
    t sum = 0;
    for (int i = 0; i < size; i++) {
        sum += arr[i];
    }
```

```
    return sum;
}
```

```
int main()
```

```
{ const int size = 10;
```

```
int intarr [size];
```

```
cout << "enter 10 integer element:";
```

```
for (int i = 0; i < size; i++) {
```

```
    cin >> intarr[i];
}
```

```
int intsum = findsum (intarr, size);
```

```
cout << "sum of integer array element = "
     << intsum << endl;
```

~~float floatarr [5] = {1.5, 2.2, 3.5, 4.4, 5.5};~~

```
float floatsum = findsum (floatarr);
```

```
cout << "sum of float array elements : " <<
float sum << endl;
```

```
return 0;
}
```

b) Write C++ program of square function using template specialisation.

- calculate the square of integer no. and a string.
(square of string is nothing but concatenation of a string with itself.)
- Write a specialised func for the square of a string.

```
# include <string>
template <typename T>
T square (T value) {
    return value * value;
}

std :: string square <std :: string>
(std :: string values) {
    return values + values;
}

int main () {
    int Num;
    std :: string text;
    std :: cout << "enter an integer: ";
    std :: cin >> Num;

    std :: cout << "enter a string: ";
    std :: cin >> text;
}
```

std :: cout << "square of integer: " <<
square (num) << std :: endl;

std :: cout << "square of string: " <<
square (text) << std :: endl;

return 0;

Pn

1111

Exp - 11

a)

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

int main () {
    char choice;
    do {
        cout << "enter two number: ";
        double a, b;
        cin >> a >> b;

        calculator <double> calc(a, b);

        cout << "addition:" << calc.add() << endl;
        cout << "subtraction:" << calc.subtract()
            << endl;
        cout << "multiplication:" << calc.multiply()
            << endl;
        cout << "Division:" << calc.divide() << endl;

        cout << "Do you want to perform another
        calculation ? (y/n): ";
        cin >> choice;
    } while (choice == 'y' || choice == 'Y');

    return 0;
}
```

b) Write to implement generic vectors. include following members functions with iterator.

a) To modify the value at a given element.

b) To multiply by a scalar value.

c) To display the vector in the form (10, 20, 30).

```
#include <iostream> : " " >> two
```

```
#include <vector>
```

```
using namespace std;
```

```
template <typename T>
```

```
class myvector {
```

```
private:
```

```
vector<T> data;
```

```
public:
```

```
void addelement (const T & value) {
```

```
data.push_back (value);
```

```
}
```

```
~void modify_element (size_t index,
```

```
const T & Newvalue) {
```

```
if (index >= data.size ()) {
```

```
cout << "index out of range: " << endl;
```

```
return 0;
```

```
}
```

```
auto it = data.begin () + index;
```

```
* it = Newvalue;
```

```
}
```

```

void multiply scalar (T scalar) {
    for (auto it = data.begin(); it != data.end(); ++it)
        *it = (*it) * scalar;
}
    
```

```

void display () const {
    for (size_t i=0; i < data.size(); ++i)
        cout << data[i];
    if (i != data.size() - 1)
        cout << ",";
    cout << endl;
}
}; 
```

```

int main() {
    my_vector<int> vec;
} 
```

```

    vec.add_element(10);
    vec.add_element(20);
    vec.add_element(30); 
```

~~cout << "initial vector :";
 vec.display();~~

~~vec.modify_element(1, 25);
 cout << "after modification :";
 vec.display();~~

~~vec.multiply_by_scalar(2);
 cout << "after multiplying by 2 :";
 vec.display();
 return;~~

Exp: 12

1) WAP using STL:

- Implement stack
- Implement Queue

```
#include <iostream>
```

```
#include <stack>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
stack<int> s;
```

```
queue<int> q;
```

```
int choice, value;
```

```
do {
```

```
cout << "1...main\n";
```

```
cout << "1. push (stack)\n";
```

```
cout << "2. pop (stack)\n";
```

```
cout << "3. display top (stack)\n";
```

```
cout << "4. enqueue (queue)\n";
```

```
cout << "5. dequeue (queue)\n";
```

```
cout << "6. display front (queue)\n";
```

```
cout << "7. exit\n";
```

```
cout << "enter your choice:\n";
```

```
cin >> choice;
```

```
switch (choice) {
```

case 1 :

```
cout << " enter value to push : ";
cin >> value;
s.push (value);
```

cout << value << " pushed onto stack. \n ";

Break;

case 2 :

```
if (!s.empty ()) {
```

cout << s.top () << " popped from stack \n ";

```
s.pop ();
} else {
```

cout << " stack is empty \n ";

}
Break;

case 3 ;

```
if (!s.empty ()) {
```

cout << " top element : " << s.top () << endl;

```
} else {
```

cout << " stack is empty . \n ";

```
}
```

Break;

case 4 :

```

cout << "enter value to enqueue: ";
cin >> value;
q.push(value);
cout < value << "enqueued into queue\n";
Break;
    
```

case 5 ;

```

if (!q.empty())
{
    cout << q.front() << "dequeued from queue
    \n";
    q.pop();
}
else {
    cout << "Queue is empty \n";
}
    
```

Break;

case 6 ;

```

if (!q.empty())
{
    cout << "front element: " << q.front() << endl;
}
else {
    cout << "Queue is empty \n";
}
Break;
    
```

Case 7:

```
cout << "program terminated \n";
Break;
```

```
default:
    cout << "invalid choice! \n";
}
while (choice != ?);
```

```
return;
```