Practical

① To add 2 numbers.

→

```
#include <iostream>
using namespace std;
int main()
{
int a, b, sum = 0;
cout << "Enter 2 numbers\n";
cin >> a >> b;
sum = a + b;
cout << "sum is " << sum;
return 0;
}
```

Output:

```
Enter any 2 numbers
5
6
Sum is 11
```

② Arithmetic operations using switch.

→

```
#include <iostream>
int main() using namespace std;
{
  float a, b;
  char ope;
cout << "+, -, /, *";
  cin >> ope;
  cout << "Enter 2 numbers";
  cin >> a >> b;
  switch (ope)
{
```

```
case '*' : cout<<" Product is "<<a*b;
           break;
case '+': cout << "sum is " <<a+b; break;
           break;
case '-': cout << " Difference is" << a-b;
           break;
case '/': if (b != 0)
           cout<<" Division is " << a/b;
           else
           cout << " cant divide";
           break;
  }
  return 0;
  }
```

output:

+ - * / : *

Enter two numbers: 5
2

Product is 10

Scanned with OKEN Scanner

③ Check if the numbers is even or odd.

→

```
#include<iostream>
using namespace std;
 int main()
{
 int num; even,odd;
cout<<"Enter an integer:";
 cin>>num;

 if (num%.2==0)
{
cout<<num<<" is even";
else
cout<<num<<" is odd";
}
return 0;
}
```

output:

Enter an integer: 6

6 is even.

④ Print 1 to 10 using:

(i) for loop

→
```
# include <iostream>
using namespace std;
 int main()
{
  int i;
for (i=0; i<=10 ; i++)
{
cout << "\n" << i;
}
 return 0;
 }
```

Output : 1 2 3 4 5 6 7 8 9 10 in vertical

(ii) while loop

→
```
#include <iostream>
 using namespace std;
  int main()
{
  int i;
   i=0;
while (i<11)
{
cout << "\n" << i;
        i++;
}
  return 0;
}
```

Output:

1
...
9
10

(5) a)

```
    *
   * *
  * * *
```

→

```
#include <iostream>
int main() → using ~~mainspace~~ namespace std;
{
int i, j, s;
for (i=1; i<=3; i++)
{
for (s=1; s<=3-i; s++)
{
cout << " ";
}
for (j=1; j<=i; j++)
{
cout << "*";
cout << " ";
}
cout << "\n";
}
return 0;
}
```

output:

```
    *
   * *
  * * *
```

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

→
```
# include <iostream>
int main()→ using namespace std;
{
int i, j;
for (i=1; i<=5; j++)
{
for ( j=1; j<=i; j++)
{
cout << j;
}
cout << "\n";
}
return 0;
}
```

output:

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

Scanned with OKEN Scanner

c) 1
2 2
3 3 3
4 4 4 4
5 5 5 5 5

→
```
# include <iostream>
int main () using namespace std;
{
 int i,j;
for (i=1; i<=5; i++)
{
for (j=1; j<=i; j++)
{
cout<<i;
}
cout <<"\n";
}
return 0;
}
```

Output:

1
2 2
3 3 3
4 4 4 4
5 5 5 5 5

Scanned with OKEN Scanner

Experiment-1

1) WAP to declare a class student having data members as roll, name. Accept & display data for one object.

→

```
#include<iostream>
using namespace std;
class student
{
private:
string name;
int roll;
public;
void accept()
{
cout<<"Enter student name\n";
cin >> name;
cout<<" Enter roll no.\n";
cin >> roll;
}
void display()
{
cout<<" Student name:"<<name;
cout<<"\n Roll no.:"<<roll;
}
};
int main()
{
student s1;
s1.accept();
s1.display();
}
```

Scanned with OKEN Scanner

Output:

```
Enter student name
Sandesh
Enter roll no.
52
Student name: Sandesh
Roll no. : 52
```

Scanned with OKEN Scanner

2) WAP to declare a class Book having data members as bid, bname, bprice. Accept data for 2 books & display name of book having greater price.

→

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

int main()
```

Scanned with OKEN Scanner

```
{
    book b1, b2;
    b1.accept();
    b2.accept();
    b1.display();
    b2.display();
    if (b1.price > b2.price)
    cout<<"\n Book 1 price is more";
    else
    cout<<"\n Book 2 price is more";
}
```

Output:

```
Enter book id
123
Enter book name
abc
Enter price
986
Enter book id
456
Enter book name
xyz
Enter price
1065

Book name: abc
Book id: 123
Price: 986
Book name: xyz
Book id: 456
Price: 1065
Book 2 price is more
```

Scanned with OKEN Scanner

3) WAP to declare a class Time. Accept time in HH:MM:SS & display total time in seconds.

→

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

class Time {
 public:
 float hr,min,sec;
 float sum;

 void accept() {
 cout<<"Enter hours:";
 cin>>hr;
 cout<<"Enter minutes:";
 cin>>min;
 cout<<"Enter seconds:";
 cin>>sec;
 }

 void display() {

 cout<<"Hours:"<<hr<<"\n";
 cout<<"Minutes:"<<min<<"\n";
 cout<<"seconds:"<<sec<<"\n";
 }

 void tosecond() {
 sum = hr * 3600 + min * 60 + sec;
 cout<<"Total seconds:"<<sum<<"\n";
 }
};
```

```
int main () {
    Time. t1;
    t1. accept ();
    t1. display ();
    t1. tosecond ();
    return 0;
}
```

Output:

```
Enter hour: 1
Enter minutes: 1
Enter seconds: 1
 Hour : 1
Minutes : 1
seconds : 1
Total seconds: 3661
```

13/6

Scanned with OKEN Scanner

Experiment-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>
using namespace std;
class city
{
    private:
    string name;
    public:
     int pop;
    void accept()
{
cout<<"Name of city?";
 cin>>name;
cout<<"population?";
 cin>>pop;
}

 void display()
{
cout<<"Name of city:"<<name;
cout<<" population:"<<pop;
 }
};

   int main()
{
   city c[5];
  int i, max=0;
for(i=0; i<5; i++)
```

Scanned with OKEN Scanner

```
{
  c[i].accept();
}
max = c[0].pop;

for(i=0; i<5; i++)
{
if(c[i].pop > c[max].pop)
  {
    max = i;
  }, cout<<"\n city with highest population\n";
  c[max].display();
}
}
```

Output:

```
Name of city:  Pune
 Population :  10000
Name of city:  Mumbai
 Population :  20000
Name of city:  Indore
 Population :  30000
Name of city:  Nagpur
 Population :  40000
Name of city:  Kolkata
 Population :  50000


Name of city  Kolkata
 Population   50000
```

Scanned with OKEN Scanner

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

→

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

class Account {
 int accNo;
 float balance;

 public:
 void accept(){
 cout<<"Enter Account Number:";
 cin>>accNo;
 cout<<"Enter Balance:";
 cin>>balance;
 }

 void addinterest()
 {
 if(balance>=5000){
   balance = balance + (balance * 10/100);
   }
 }

  void display()
 {
 cout<<"Account No:"<<accNo<<" Balance:"
 <<balance<<endl;
 }
```

Scanned with OKEN Scanner

```
float getBalance()
 {
 return Balance;
 }
};

int main() {
 Account a[10];
for(int i=0; i<10; i++)
 {
 a[i].accept();
 a[i].addInterest();
 }

cout<<"\n Accounts with Balance >=5000\n";
for( int i=0; i<10; i++)
 {
 if (a[i].getbalance()>=5000)
 {
    a[i].display();
 }
}
 return 0;
}
```

O/P :

```
Enter Account number: 1
Enter Balance : 5000
Enter Account number: 2
 Enter Balance : 6500
Enter Account number: 3
Enter Balance: 4000
```

Scanned with OKEN Scanner

Enter Account Number: 4
Enter Balance: 8900
Enter Account Number: 5
Enter Balance: 3200
Enter Account Number: 6
Enter Balance: 7800
Enter Account Number: 7
Enter Balance: 1200
Enter Account Number: 8
Enter Balance: 10000
Enter Account Number: 9
Enter Balance: 2300
Enter Account Number: 10
Enter Balance: 4999

Accounts with Balance >= 5000:
Account No: 1 Balance: 5500
Account No: 2 Balance: 7150
Account No: 4 Balance: 9790
Account No: 6 Balance: 8580
Account No: 8 Balance: 11000

Scanned with OKEN Scanner

c) WAP to declare a class 'staff' having data members as name and post. Accept this data for 5 staff and display names of staff who are "HOD".

→

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

void display()
{
cout<<"\n staff Name:"<<name<<"\n staff Post:"<<post<<endl;
if(post=="HOD"||post=="hod")
{
cout<<name<<" is Head of Department.\n";
}
else
{
cout<<name<<" is a staff.\n";
}
}
};
```

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

Q1
13/8

Scanned with OKEN Scanner

Experiment-3

a) WAP to declare a class 'book' containing data members as book_title, author_name, and price. Accept and display the information for one object using a pointer to that object.

→
```
#include<iostream>
using namespace std;
class book{
 string title;
 string a_name;
   int price;
 public:
void info() {
cout<<"enter the book title, author name
and price of the book:";
cin>>title>>a_name>>price;
}

void display() {
cout<<"Book name:"<<title<<"\t author
name:"<<a_name<<"\t price:"<<price;
}};
};

int main() {
   Book *n;
   Book b1;
    n = &b1;
    n->info();
    n->display();
```

output:

enter the book, title, author name and price of the book:

Book 1
J.K. Rowling
450

Book name: Book 1 author name: J.K. Rowling
price: 450

2) WAP to declare a class 'student' having data members roll_no and percentage. Using 'this' pointer.

→
```
#include<iostream>
using namespace std;
class student{
  int roll;
  float perc;
 public:
 void info(int roll_no, float percentage){
   this -> roll = roll_no;
   this -> perc = percentage;
  }
   void display (){
 cout<<"roll no.:"<<roll<<"\t mark:"<<perc;
  }
};

 int main (){
  student s;
  s.info(3, 90);
 s.display();
 }
```

O/P:
roll no.: 3 marks: 90

Scanned with OKEN Scanner

Nested class

3)
```
#include<iostream>
using namespace std;
class Marks {
 public:
class Percentage {
 int marks, n;
 float n, i;
 public:
 void info () {
cout<<" Enter the marks you got:";
cin >> marks;
cout<<" Enter total marks:";
 cin >> n;
}
 void display () {
  i = (float) marks/n;
  n = i * 100;
cout<<" percentage:" <<n;
  }
 };
};
 int main () {
 Marks m1;
 Marks :: Percentage n1;
    n1.info();
    n1.display();
  }
```

O/P:

```
Enter the marks you got: 67
Enter total marks: 70
 Percentage: 95.71
```

13/8

Experiment-4

1. Swap 2 numbers from same using object as function argument.

→
```
#include<iostream>
using namespace std;
class number {
 int value;
 public:
number ( int v=0) {
 value = v;
  }
void swap (number & other) {
 int temp = value;
value = other.value;
other.value = temp;
  }
void disp() {
cout <<" value: " << value<<endl;
}};
int main() {
number n1(10), n2(20);
cout <<" Before swap:" <<endl;
 n1.disp();
 n2.disp();
 n1.swap(n2);
cout <<"\n After swap:" <<endl;
 n1.disp();
 n2.disp();
 return 0;
}
```

o/p : Before swap:
value : 10
value : 20

After swap:
value : 20
value : 10

Scanned with OKEN Scanner

2. Swap 2 numbers from same class using friend fuction

→
```
# include<iostream>
using namespace std;
class AB {
int a,b;
public:
void info() {
cout<<"Enter 2 numbers:";
cin>>a>>b;
}
friend void swap (AB al);
};
void swap (AB al) {
int temp;
temp = al.a;
al.a = al.b;
al.b = temp;
cout<<"In Values after swapping:"<<al.a<<
al.b;
}
int main() {
AB al;
al.info;
swap(al);
}
```

O/P :

Enter 2 numbers : 5
4

Value after swapping : 4
5

Scanned with OKEN Scanner

3. Friend function swap 2 numbers different class

→
```
#include <iostream>
using namespace std;
 class CB;
class CA;
int num A;
public:
CA (int val) : num A (val) { }
void disp () { {
cout<<("value in class A :" << num A << end l;
}
friend void swap ( CA &, CB &);
 };
class CB{
private:
 int num B;
CB (int val) : num B (val) {}
void disp () {
cout <<("value in class B :" << num B <<(end l;
}
friend void swap (CA & CB &);
};
Void swap (CA & a, CB & b) {
{
 int temp = a.num A;
 a.num A = b.num B;
  b.num B = temp;
 }

int main () {
CA obj A (10);
CB obj B (20);
```

Scanned with OKEN Scanner

```
cout<<" Before swapping:" <<endl;
obj A.disp();
obj B.disp();
swap(obj.A, obj.B);
cout<<" In After swapping:" <<endl;
obj A.disp();
obj B.disp();
return 0;
}
```

O/P:

```
Before swapping:
value in Class A: 10
value in Class B: 20

After swapping:
value in Class A: 20
value in Class B: 10
```

Scanned with OKEN Scanner

4. Avg of two results.

→
```
#include<iostream>
using namespace std;
 class result2;
 class result{
  int a;
  public:
 void accept() {
 cout<<" Enter marks out of 50:";
 cin>>a;
 }
 friend void cal(result r1, result2 r2);
 };
 class result2 {
 int b;
 public:
 void accept() {
 cout<<" Enter marks out of 50:";
 cin>>b;
 }
 friend void cal (result r1, result r2);
 };
 void cal( result r1, result r2) {
 float avg = (float) (r1.a + r2.b) /2;
 cout<<"\n avg:" <<avg;
 }
 int main (){
  result x;
  result2 y;
  x.accept ();
  y.accept ();
  cal (x,y);
  }
```

Scanned with OKEN Scanner

O/P: Enter marks out of 50: 45
Enter marks out of 50: 46
avg: 45.5

5. Greatest among 2 numbers (Diff class) (Friend functio

→
```
#include<iostream>
using namespace std;
class B;
class A {
 int a;
 public:
void acc() {
cout<<"Enter value:";
cin>>a;
 }
friend void gr(A al, B al);
};
class B {
 int b;
 public:
void accept() {
cout<<"Enter a value:";
cin>>b;
 }
friend void gr(A al, B al);
};
void gr(A al, B bl) {
if(al.a > bl.b) {
cout<<"First value is greater";
 }
else {
cout<<"second value is greater";
 }
```

Scanned with OKEN Scanner

```
int main() {
  A x;
  B y;
x.acc();
y.acc();
gr(x,y);
}
```

O/P:

```
Enter value: 10
Enter value: 100
Second value is greater.
```

14/8

Scanned with OKEN Scanner

Experiment-5

5 a) Write a cpp program to implement types of constructor

Experiment-4

① Passing of object as a function argument (swap)

→

```
#include<iostream>
using namespace std;
 class number
 {
 int value;
 public:
 number (int v=0) {
 value = v;
 }
 void swap ( number & other) {
 int temp = value;
 value = other.value;
 }other.value = temp;
 }
 void disp () {
cout<< "value:" <<value << endl;
 }};
 int main () {
number n1(10), n2(20);
cout << "Before swap:" << endl;
 n1.disp();
 n2.disp();
 n1. swap(n2);
cout<< "In After swap:" <<endl;
 n1. disp();
 n2. disp();
 return 0;
 }
```

Scanned with OKEN Scanner

→ O/P : Before swap:
Value : 10
Value : 20
After swap:
Value : 20
Value : 10

② Friend function (swap same class)

→
```
#include<iostream>
using namespace std;
class AB {
int a,b;
public:
void info() {
cout<<" Enter 2 numbers:";
cin>>a>>b;
}
friend void swap (AB &al);
};
void swap (AB &al) {
int temp;
temp = al.a;
al.a = al.b;
al.b = temp;
cout<<"\n Values of after swapping:"<<al.a<<al.b;
}
int main() {
  AB.al;
al.info;
swap (al);
}
```

→ o/p: Enter 2 numbers: 5
4
Values after swapping: 4 5

③ Friend function (swap different class)

→
```
#include<iostream>
using namespace std;
 class CB;
 class CA {
 int numA;
 public:
CA(int val): numA(val) {}
 void disp() {
cout<<" value in class A:"<<numA<<endl;
 }
 friend void swap (CA&, CB&);
 };
 class CB {
 private:
int numB;
CB(int val): numB(val) {}
 void disp() {
cout<<"value in class B:"<<numB<<endl;
 }
 friend void swap (CA&, CB&);
 };
 void swap (CA&a, CB&b)
 { int temp = a.numA;
      a.numA = b.numB;
      b.numB = temp;
 }
```

Scanned with OKEN Scanner

```
int main () {
(A objA (10);
(B objB (20);
cout<<" Before swapping: " <<endl;
objA.disp();
objB.disp();
swap (objA, objB);
cout<<"\n After swapping :"<<endl;
objA.disp();
objB.disp();
return 0;
}
```

O/P:

Before swapping:
value in Class A: 10
Value in Class B: 20

After swapping:
Value in Class A: 20
Value in Class B: 10

④ Avg of two results

→

```
# include <iostream>
Using namespace std;
class result2;
class result1{
  int a;
public:
void accept () {
cout<<" Enter marks out of 50: ";
cin>> a;
}
```

Scanned with OKEN Scanner

```
friend void cal(result r1, result2 r2);
};
class result2 {
 int b;
 public:
void accept() {
cout<<"Enter marks out of 50:";
 cin>>b;
 }
friend void cal(result r1, result2r2);
 };
 void cal(result r1, result2 r2) {
 float avg = (float)(r1.a + r2.b)/2;
 float avo
cout<<"\n avg:"<<avg;
 }
 int main() {
 result x;
 result2 y;
 x.accept();
 y.accept();
 cal(x,y);
  }
```

O/P:

```
Enter marks out of 50:45
Enter marks out of 50:46
  avg:45.5
```

Scanned with OKEN Scanner

⑤ Greatest among 2 numbers (Different class) (friend function)

→

```
#include<iostream>
using namespace std;
 class B;
 class A {
 int a;
 public:
 void acc() {
cout<<"Enter value:";
cin>>a;
 }
friend void gr (A a1, B b1);
 };
 class B {
 int b;
public:
void acc () {
cout<<"Enter a value:";
cin>>b;
}
friend void gr (A a1, B b1);
};
void gr ( A a1, B b1) {
if (a1.a > b1.b) {
cout<<"First value is greater";
}
else {
cout<<"second value is greater";
}
int main () {
```

Scanned with OKEN Scanner

```
A x;
B y;
x.acc();
y.acc();
gr(x,y);
}
```

O/P:

```
Enter value: 10
Enter value: 100
Second value is greater
```

12/11

Scanned with OKEN Scanner

Experiment-4

* Write a C++ program to demonstrate passing of a object as function argument.

① Average of two results using 2 class.

→
```
# include <iostream>
using namespace std;
class result2;
class result1 {
 int a;
public:
void acc() {
cout<<" Enter marks out of 50:";
cin>> a;
 }
friend void cal( result1 r1, result2 r2);
 };
class result2 {
 int b;
public:
void acc () {
cout<< Enter marks out of 50 : ";
cin>> b;
 }
friend void cal(result1 r1, result2 r2);
 };
void cal( result1 r1, result2 r2) {
 float avg =(float) (r1.a/r1.b) /2;
cout<<" In avg : " avg <<avg;
 }

int main () {
result1 x;
result2 y;
```

```
x.acc();
y.acc();
cal(x,y);
}
```

O/P: Enter marks out of 50 : 45
Enter marks out of 50 : 46
avg : 95.5

② Find the greater number using 2 classes and friend functio

→
```
#include<iostream>
using namespace std;
class BB;
class AA {
 int a;
 public:
void acc(){
cout<<" enter 'a' value: ";
cin >> a;
}
friend void avg (AA a1, BB b1);
};
class BB {
 int b;
 public:
void acc() {
cout<<" enter 'b' value: ";
cin>> b;
}
friend void avg (AA a1, BB b1);
};
```

Scanned with OKEN Scanner

```
void avg (AA a1, BB b1) {
if (a1.a > b1.b) {
cout<<"\n a is the greater value";
 }
else {
cout<<"\n b is the greater value";
}
int main() {
 AA a;
 BB b;
 a.acc();
 b.acc();
 avg(a, b);
}
```

O/P: Enter 'a' value : 4
Enter 'b' value : 10
b is the greater value

③ Swap values using friend function

→
```
#include <iostream>
using namespace std;
 class AA{
 int a, b;
 public:
 void acc() {
 cout <<"enter 2 values:";
 cin >> a>> b;
 }
 friend void swap (AA a1);
 };
```

```
void swap (AA a1) {
 int temp;
 temp = a1.a;
a1.a = a1.b;
a1.b = temp;
cout<<"\n Values after swapping:"<<a1.a
<<a1.b;
}
int main() {
 AA a1;
a1. acc();
 swap (a1);
 }
```

O/P:

Enter two values: 4
5

Values after swapping: 5
4

12/11

Experiment-5

Q. Write a c++ program to implement types of constructors. a) Find sum of nos betn 1 to n using
① Default constructor consmuctor where value of n will be passed to the constructor.

```
#include<iostream>
using namespace std;
class sum
{
 int n;
 public:
sum(){
 n=5;
 }
void calc()
 {
 int s=0, i;
for(i=0; i<=n; i++)
 {
  s=s+i;
 }
cout<<"sum is:"<<s;
 }
};
int main()
 {
  sum s1;
 s1.calc();
 }
```

O/P:

sum is: 15

(2) Parameterized constructor

```
#include<iostream>
using namespace std;
class sum{
  int no;
  public:
 sum(int n) {
    no=n;
  }
 void calc()
 {
  int s=0,i;
 for(i=0;i<=no;i++)
  {
   s=s+i;
  }
 cout<<"sum is:"<<s;
  }
};

int main()
{
 sum s1(5);
 s1.calc();
 }
```

O/P:

Sum is: 15

③ copy constructor

```
#include<iostream>
using namespace std;
 class sum{
   int no;
   public:
  sum(int n) {
    no=n;
   }
  sum(sum &su)
  {
   no=su.no;
  }
  void calc()
  {
   int s=0,i;
 for(i=0;i<=no;i++)
  {
   s=s+i;
  }
 cout<<"sum is:"<<s;
  }
};

 int main()
 {
 sum s1(5);
 sum s2(s1);
 s1.calc();
  }
```

O/P:

sum is: 15

Scanned with OKEN Scanner

b) WAP to declare a class "student" having data members as name and percentage. write a constructor to initialize these data members. Accept and display data for one student.

→
```
#include<iostream>
using namespace std;
class student {
 string name;
  int m1,m2;
  int total;
  public:
 student() {
 name = "ABC";
  m1 = 72;
  m2 = 90;
  total = 200;
  }
  void calc() {
 float perc = (float) (m1+m2) / total * 100;
 cout<< "Student Name:" <<name <<endl;
 cout<< "Percentage:" <<perc << endl;
   }
 };
 int main() {
  student s1;
  s1. calc();
  }
```

① Default constructor

O/P;

Student Name: ABC
Percentage : 81

Scanned with OKEN Scanner

② Parameterized constructor

```
#include<iostream>
using namespace std;
class student{
 string name;
 int m1, m2;
 int total;
 public:
student(string n, int mu1, int mu2, int t){
 name=n;
 m1=mu1;
 m2=mu2;
 total=t;
 }
 void calc()
 {
float perc=(float)(m1+m2)/total*100;
cout<<"Student Name:"<<name<<endl;
cout<<"Percentage:"<<perc<<endl;
 }
};
 int main()
 {
 student s1("ABC",90,90,200);
  s1.calc();
 }
```

O/P:

Student Name: ABC
Percentage: 90

Scanned with OKEN Scanner

② Copy constructor

```
#include<iostream>
using namespace std;
class student {
 string name;
 int m1,m2;
 int total;
 public:
student(string n, int mu1, int mu2, int t) {
  name = n;
  m1 = mu1;
  m2 = mu2;
  total = t;
 }
 student(student & st) {
 name = st.name;
 m1 = st.m1;
 m2 = st.m2;
 total = st.total;
 }
 void calc()
 {
float perc = (float) (m1+m2) / total * 100;
cout<<" student Name:"<<name<<endl;
cout<<" Percentage:" << perc << endl;
 }
};
int main() {
student s1("ABC", 90,90,200);
 student s2(s1);
  s1.calc();
  s2.calc();
 }
```

O/P: Student Name: ABC
Percentage: 90
Student Name: ABC
Percentage: 90

(2|1)

Scanned with OKEN Scanner

Experiment-6

1. Single inheritance

→

```
#include<iostream>
using namespace std;
 class person {
 protected:
 string name;
  int age;
 };
class student : protected person
  {
 int roll_no;
 public:
 void setdata (string n, int a, int r)
  {
 name = n;
  age = a;
 roll_no = r;
  }
 void display () {
 cout<<"Name:"<<name<<endl;
 cout<<"Age:"<<age<<endl;
 cout<<"Roll No:"<<roll_no<<endl;
  }
};
 int main () {
 student s;
s.setdata ("shlok", 20, 101);
s. display ();
 return 0;
}
```

O/P :

Name: shlok
Age : 20
Roll No: 101

Scanned with OKEN Scanner

2. Multiple inheritance

→
```
#include <iostream>
using namespace std;
class academic {
    protected:
  string name;
   int marks;
};
class sports {
    protected:
   int score;
};
class student : protected academic, protected sports {
 int total;
 public:
 void accept () {
 cout<<" Enter name:";
 cin>> name;
cout<<" Enter marks:";
cin >> marks;
cout<<" Enter score:";
cin >> score;
total = marks + score;
 }
 void display () {
cout<<" Name:" << name <<endl;
cout<<" Marks:" << marks <<endl;
cout<<" Score:" << score << endl;
cout<<" Total :" << total << endl;
 }
};
```

Scanned with OKEN Scanner

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

O/P:

```
Enter name: shlok
Enter marks: 89
Enter score : 7
Name : shlok
Marks: 89
score : 7
Total : 96
```

Scanned with OKEN Scanner

3. Multilevel Inheritance

→
```
#include<iostream>
using namespace std;
 class vehicle {
   public:
 string model;
 string brand;
 };
class car : public vehicle {
   protected:
  int type;
 };
class electriccar : public car {
 public:
int battery_capacity;
void accept() {
cout<<" Enter Model:";
cin >> model;
cout<<" Enter Brand:";
 cin >> brand;
 cout<<" Enter Type (1 for Sedan, 2 for SUV):";
 cin >> type;
 cout<<" Enter Battery Capacity (in kwh):";
 cin>> battery_capacity;
}
 void display () {
cout<<" Model:" << model << endl;
cout<<" Brand:" << brand << endl;
cout<<" Type:" << (type == 1 ? " Sedan : SUV")
  << endl;
cout<<" Battery Capacity:" << battery_capacity
  << " kWh" << endl;
```

Scanned with OKEN Scanner

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

O/P:

```
Enter Model: X7
Enter Brand: BMW
Enter Type (1 for Sedan, 2 for SUV): 2
Enter Battery capacity (in kwh): 1000
Model: X7
Brand: BMW
 Type: SUV
Battery capacity: 1000 kwh
```

Scanned with OKEN Scanner

④ Hierarchical

→
```
#include<iostream>
using namespace std;
 class emp {
 protected:
 string name;
  int id;
 };
class man : public emp { private:
  string dep;
 };
class dev : public emp {
 public :
string lang;
string dep;
void acc(){
cout<< "Emp name:";
cin >> name;
cout<< "Emp ID";
 cin>> id;
Cout<< "Enter programming Lang:";
 cin>> Lang;
cout<<"Enter dept:";
 cin >> dep;
 }
void disp (){
cout<<"Name:" <<name <<endl<<"ID" <<id<<
endl;
cout<<" Programming Lang:"<<Lang <<endl<<
"Dept:" << dep;
 }
};
```

Scanned with OKEN Scanner

```
int main(){
 dev d;
d.acc();
d. disp();
}
```

O/P:

```
Emp name: ABC
 Emp ID: 101
Enter Programming Lang: CPP
Enter dept: HOD
Name: ABC
  ID : 101
Programming  lang: CPP
 Dept: HOD
```

⑤ ~~Hybrid~~

Scanned with OKEN Scanner

⑤ Hybrid

→
```
#include<iostream>
using namespace std;
class teacher {
protected:
string name;
int age;
public:
void acc()
cout<<" Enter teacher name:";
cin>>name;
cout<<"Teacher age:";
cin>>age;
}
void disp(){
cout<<" Teacher name :"<<name<<endl;
cout<<"Age:"<<age<<endl;
};
class student : public teacher {
protected:
int roll;
string sname;
void acc(){
cout<<" Enter student name:";
cin>>sname;
cout<<"student roll no:";
cin>>roll;
}
void disp(){
cout<<"student name:"<<sname<<endl;
cout<<" Roll no:"<<roll<<endl;
};};
```

Scanned with OKEN Scanner

```
class marks: public student {
protected:
int m1, m2, m3
public:
void accept() {
cout<<"Enter marks of 3 subj:";
cin>>m1>>m2>>m3;
}
void dispm() {
cout<<" Marks in subj 1,2,3: "<<m1<<endl<<m2
<<endl<<m3<<endl;
};
class acad: public marks {
public:
int total() {
return m1+m2+m3;
};
int main() {
acad a;
a.acc();
a.accs();
a.accm();
a.disp();
a.disps();
a.dispm();
int total = a.total();
cout<<" Total marks: "<<total<<endl;
}
```

Scanned with OKEN Scanner

O/P:

```
Enter teacher name: ABC
Teacher age: 40
Enter student name: XYZ
Student roll no: 35
Enter marks of 3 subj: 90
90
90
Teacher name: ABC
 Age: 40
student name: XYZ
 Roll no: 35
Marks in subj 1, 2, 3: 90
90
90
Total marks: 270
```

12/11

Practical-7

6) Area of laboratory (rectangle) and area of classroom (square). (constructor overloading)

→

```
#include (iostream)
using namespace std;
class xyz {
public:
int l,b,s,a;
void calc(int len, int br) {
   l = len;
   b = br;
a = l*b;
}
void calc( int si) {
 s=si;
 a=s*s;
 }
};
int main() {
xyz a;
cout<<"Enter length and breadth of rectangl
:";
cin>>a.l>>a.b;
a.calc(a.l,a.b);
cout<<"Area of rectangle:"<<a.a<<endl;
 xyz b;
cout<<"Enter side of square:";
cin>>b.s;
b.calc(b.s);
cout<<"Area of square:"<<b.a<<endl;
return 0;
}
```

Scanned with OKEN Scanner

→ O/P: Enter length and breadth of rectangle: 10
5
Area of rectangle: 50
Enter side of square: 5
Area of square: 25

② Constructor overloading (sum of 10 integer and 5 float)

→
```
#include<iostream>
using namespace std;
 class sum {
  public:
float fsum = 0;
 int isum = 0;
 void calc (float a) {
  fsum = fsum + a;
 }
 void calc (int a) {
 isum = isum + a;
 }
 };
 int main () {
  sum s;
 float f;
   int i;
 cout << " Enter 5 float values :";
  for (int j=0; j<5; j++);
  cin >> f;
  s.calc ();
  }
 cout <<"Sum of float values: " << s.fsum << endl;
 cout << " Enter 10 integer values:";
```

```
for (int j=0; j<10; j++) {
cin >> i;
s.calc(i);
}
cout<<"sum of integer values:"<<s.isum<<endl;
return 0;
}
```

③ Unary-operator

→
```
#include <iostream>
using namespace std;
class numb {
  public:
int val;
void acc(){
cout<<"Enter a numb:";
cin >> val;
 }
void disp(){
cout<<"value:"<<val;
}
void operator-(){
val = -val;
};
int main(){
numb obj;
obj.acc();
-obj;
obj.disp();
}
```

O/P:

Enter a numb: 5

Value: -5

Scanned with OKEN Scanner

(5) Unary ++ operator (Pre & post increment)

→
```
#include <iostream>
using namespace std;
class ab {
 public:
 int num;
void acc() {
cout<<"Enter number:";
cin >>num;
}
void disp() {
cout<<" Number:" <<num <<endl;
}
void operator++() {
++num;
}
~~void operator++()~~ → void operator ++(int) {
                          num++;
                        };
int main() {
ab obj;
obj.acc();
++obj;
obj.disp();
 return 0;
 }
```

O/P:

Enter number: 5
Number: 7

12/11

Scanned with OKEN Scanner

Practical-8

① Overload '+' operator so that two strings can be concatenated.

→
```
#include <iostream>
#include <string>
using namespace std;
class abc{
public:
string str;
void acc(){
cout<<"Enter string:";
cin>>str;
}
abc operator+(abc s){
abc temp;
temp.str = str+s.str;
return temp;
}
void disp(){
cout<<"concatenated string:"<<str<<endl;
}};
int main(){
abc s1,s2,r;
s1.acc();
s2.acc();
r=s1+s2;
r.disp();
return 0;
}
```

O/P:

```
Enter string : Hello
Enter string : World
Concatenated string : Hello World
```

Scanned with OKEN Scanner

(2)

→
```
#include (iostream)
using namespace std;
class ILogin {
protected:
string name, password;
public:
void accept() {
cout<<"Name:";
cin >> name;
cout << "password:";
cin>> password;
 }
};
class EmailLogin: virtual public ILogin {
public:
void showEmail () {
cout<< name <<" " << password <<endl;
 }
};
class membershiplogin :virtual public ILogin {
public:
void showMembership () {
cout<< name<<" " << password<< endl;
 }
};
class Employee: public EmailLogin, public
MembershipLogin {
public:
 void input () {
   accept();
 }
```

Scanned with OKEN Scanner

```
    void display(){
    showEmail();
    showMembership();
     }
};
int main(){
 Employee e;
 e.input();
 e.display();
 return 0;
}
```

O/P:

Name: ABC
Password: 123

ABC 123
ABC 123

12/11

Scanned with OKEN Scanner

Practical-9

①

→

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

int main() {
fstream f1, f2;
  char ch;
f1.open ("First.txt", ios::in);
f2.open ("Second.txt", ios::out);

while (f1.get(ch)) {
      f2.put(ch);
}

cout <<" File copied successfully. " <<endl;

f1.close();
f2.close();
  return 0;
}
```

O/P:

File copied successfully.

Scanned with OKEN Scanner

(2)

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

int main(){
fstream File;
File.open("File.txt",ios::in);

if(!File){
cout<<"Unable to open file";
return 1;
}

char ch;
int digit_count=0, space_count=0;

while (File.get(ch)) {
if (ch>='0' && ch<='9') {
    digit_count++;
  }
else if (ch==' ') {
    space_count++;
   }
}
cout<<"Digits: "<<digit_count<<endl;
cout<<"Spaces: "<<space_count<<endl;
File.close();
return 0;
}
```

O/P:

Unable to open file

Scanned with OKEN Scanner

③

→

```
#include <iostream>
#include <fstream>
using namespace std;
int main () {
fstream File;
file.open ("File.txt", ios::in);
if (! File) {
cout << "Unable to open file";
return 1;
}
char ch;
int word_count = 0;
int prev_type = 0;
while (File.get (ch)) {
int current_type = ((ch >= 'a' && ch <= 'z') ||
                    (ch >= 'A' && ch <= 'Z') ||
                    (ch >= '0' && ch <= '9')) ? 1 : 0;
if current_type == 1 && prev_type == 0) {
   word_count++;
}
prev_type = current_type;
}
cout << "Word count :" << word_count << endl;
File.close ();
return 0;
}
```

O/P:

Unable to open file

12/11

Experiment-10

```
#include (iostream)
using namespace std;
template (typename T)
T sum (T a[] , int n) {
    T s = 0;
for( int i=0; i(n; i++)
s = s + a[i];
return s;
}
int main () {
int i[10] = {1,2,3,4,5,6,7,8,9,10};
float f[10] = {1.1,2.2,3.3,4.4,5.5,6.6,7.7,8.8,9.9,10.10};
cout (( sum (i,10) (( endl;
cout (( sum (f,10) (( endl;
}
```

o/p:

55

59.6

Scanned with OKEN Scanner

(2)

```
#include <iostream>
#include <string>
using namespace std;
template <typename T>
 T square (T x) {
   return x * x;
}
template <>
string square <string> (string s) {
   return s+s;
 }
int main() {
cout << square (5) <<endl;
cout<< square (string ("Aaa")) <<endl;
 return 0;
 }
```

O/P:

25

AaaAaa

(2/11

(3)

```
#include<iostream>
using namespace std;
template <class T>
class Calculator {
public:
T add(T a, T b) { return a+b; }
T sub(T a, T b) { return a-b; }
T mul(T a, T b) { return a*b; }
T div(T a, T b) { return (b==0)? 0: a/b; }
T mod(T a, T b) { return (b==0)? 0: a%b; }
T max(T a, T b) { return (a>b) ? a: b; }
T min(T a, T b) { return (a<b) ? a: b; }
T square(T a) { return a*a; }
T avg(T a, T b) { return (a+b)/2; }
T cube(T a) { return a*a*a; }
};

int main(){
Calculator<int> c;
 int ch;
while (1) {
cout<<"\n Enter choice:\n";
cout<<"1. Add\n 2. subtract\n 3. Multiply\n
    4. Divide\n";
cout<<"5. Modulus\n 6. Maximum\n 7. minimum\n
     8. Square\n" ;
cout<<" 9. Average\n 10. Cube\n 11. Exit\n";
cout<<"choice:";
cin>>ch;

switch(ch) {
```

```
case 1 : {
int a,b;
cout<<"Enter two numbers:";
cin>>a>>b;
cout<<"Result:"<<c.add(a,b)<<endl;
break;
}
case 2 : {
int a,b;
cout<<"Enter two numbers:";
cin>>a>>b; break;
cout<<"Result:"<<c.sub(a,b)<<endl;
break;
}
case 3 : {
int a,b;
cout<<"Enter two numbers:";
cin>>a>>b;
cout<<"Result:"<<c.mul(a,b)<<endl;
break;
}
case 4 : {
Int a,b;
cout<<"Enter two numbers:";
cin>>a>>b;
cout<<"Result:"<<c.div(a,b)<<endl;
case 5 : { break;
}
Int a,b;
cout<<"Enter two numbers:";
cin>>a>>b;
cout<<"Result:"<<c.mod(a,b)<<endl;
break; }
case 6 : {
int a,b;
cout<<"Enter two numbers:";
cin>>a>>b;
```

Scanned with OKEN Scanner

```
cout<<"Result:"<<c.max(a,b)<<endl;
break;
}
case 7: {
int a,b;
cout<<"Enter two numbers:";
cin>>a>>b;
cout<<"Result:"<<c.min(a,b)<<endl;
break;
}
case 8: {
int a;
cout<<"Enter number:";
cin>>a;
cout<<"Result:"<<c.square(a)<<endl;
break;
}
case 9: {
int a,b;
cout<<"Enter two numbers:";
cin>>a>>b;
cout<<"Result:"<<c.avg(a,b)<<endl;
break;
}
case 10: {
int a;
cout<<"Enter number:";
cin>>a;
cout<<"Result:"<<c.cube(a)<<endl;
break;
}
case 11: {
  return 0;
}
```

Scanned with OKEN Scanner

```
default:
cout << "Invalid choice or exit requested.";
return 0;
}
}
return 0;
}
```

O/P:

```
Choice : 1
Enter two numbers : 4
5
Result : 9
```

12/11

(4)

→

```
#include<iostream>
using namespace std;
template<class T>
class stack {
  T arr[10];
   int top;
public:
stack() { top=-1;}

void push (T val) {
  if (top<9)
  arr[++top]=val;
  else
cout<<"stack overflow!" <<endl;
}
T pop() {
if (top==-1) {
cout<<"stack underflow!" <<endl;
   return 0;
} else {
return arr[top--];
  }
}
};
int main() {
stack<int> s;
  int ch;
while (1) {
cout<<"\n1. Push\n2. Pop\n3. Exit\n Enter
  choice:";
cin>>ch;
```

```
switch (ch) {
case 1: {
int val;
cout << "Enter value to push:";
 cin >> val;
s.push (val);
 break;
 }
case 2:
int popped = s.pop();
if ( popped != 0)
cout << "Popped:" << popped << endl;
break;
}
case 3:
cout << "Exiting...." << endl;
return 0;
default:
 cout << "Invalid choice!" << endl;
   }
}
return 0;
}
```

```
O/P:
Enter choice: 1
Enter value to push: 23
Enter choice: 2
Popped: 23
Enter choice: 3
 Exiting....
```

Scanned with OKEN Scanner

Page No.:
Date:
YOUVA

Experiment - 11

①

→

```
#include<iostream>
using namespace std;
template <class T>
class vector {
  T a[100];
 int size;
public:
 vector (int s) {
   size = s;
  }
 void set(int i, T val) {
  if (i>=0 && i<size) {
    a[i] = val;
  }
  else {
 cout << "Invalid";
  }
 }
 T get (int i) {
 if (i>=0 && i<size) {
  return a[i];
 }
 else {
  cout << "Invalid";
  return T();
  }
}
 void disp () {
  for (int i=0; i<size; i++) {
  cout << a[i] << " ";
  }
```

Scanned with OKEN Scanner

```
        cout<<endl;
    }
};

int main(){
vector<int> v(5);
for(int i=0; i<5; i++){
  v.set(i,(i+1)*10);
  }
v.disp();
v.set(10,99);
v.set(3, 99);
cout<<"\n After modification:";
  v.disp();
return 0;
}
```

O/P:

10 20 30 40 50

Invalid

After modification: 10 20 30 99 50

Scanned with OKEN Scanner

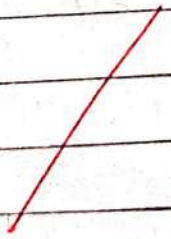
(2)

→

```
#include<iostream>
#include<vector>
using namespace std;
int main() {
vector<int> vec = {1,2,3,4,5};
 int scalar = 3;
 for(int &value : vec){
 value = value * scalar;
 }
 for(int val : vec) {
 cout<<val<<" ";
 }
cout<<endl;
 return 0;
}
```

O/P:

3 6 9 12 15



Scanned with OKEN Scanner

(3)

→

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

 int main() {
vector <int> vec = {10,20,30,40,50};
 cout <<"(";
for (int i=0; i< vec.size(); i++) {
 cout << vec[i];
 if( i != vec.size() - 1) {
 cout <<", ";
    }
 }
 cout <<")";
  return 0;
}
```

O/P:

(10,20,30,40,50)

12/11

Scanned with OKEN Scanner

Experiment-12

① →

```
#include <iostream>
#include <stack>
using namespace std;
int main() {
stack <int> s;
s.push(10);
s.push(20);
s.push(30);
cout<<"Top element:"<<s.top()<<endl;
s.pop();
cout<<"Top element after pop:"<<s.top()
<<endl;
cout<<"stack size:"<<s.size()<<endl;

if (s.empty()) {
cout<<"stack is empty"<<endl;
 }
 return 0;
}
```

O/P:

```
Top element: 30
Top element after pop: 20
Stack size: 2
```

Scanned with OKEN Scanner

(2)

```
#include (iostream)
#include < queue>
 using namespace std;
  int main () {
  queue (int) q;
  q.push(10);
  q.push(20);
  q.push(30);
 cout<< "Front element:" <<q.front() <<endl;
 cout<<"Back element:" << q.back() <<endl;

 q.pop();
cout<<"After pop operations"<<endl;
cout<<"Front element:" << q.front() <<endl;
cout<<" Queue size:" << q.size() <<endl;

 if (q.empty()) {
  cout<<" Queue is empty"<<endl;
    }
 return 0;
 }
```

o/p:

```
Front element: 10
Back element: 30
After pop operations
Front element: 20
 queue size: 2
```

12/11