

11

OOP - Practical

1) Write a C++ program for:

- a) Sum of 2 numbers
- b) Arithmetic operation by switch
- c) Check even or odd
- d) print nos. 1 to 10 nos using for loop
- e) _____ using while loop
- f) Print below:
* * * * * b) 1 2 3 4 5 c) 1 2 3 4 5
* * * * * 1 2 3 4 5 3 3 3
* * * * * 1 2 3 4 5 4 4 4 4
* * * * * 1 2 3 4 5 5 5 5 5

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

Output:

Enter any 2 numbers

5

6

Sum is 11.

2: Arithmetic Operators using switch.

```
#include <iostream>
using namespace std;
int main()
{
    float a, b;
    char o;
    cout << "+ - * ";
    cin >> ope;
    cout << " Enter 2 numbers";
    cin >> a >> b;
    switch (o)
    {
        case '+': cout << "Sum is " << a+b;
                     break;
        case '-': cout << "Difference
                     is " << a-b;
                     break;
        case '*': cout << "Product is " << a*b;
                     break;
        case '/': if (b!=0)
                     cout << "Division is " << a/b;
                     else
                     cout << "Can't divide";
                     break;
    }
    return 0;
}
```

Output:

+ - * / : *

Enter two numbers = 5
2

Product is 10

3. Check if the no. is even or odd.

```
#include <iostream>
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";
}
```

3
3

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;
}
```

(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
...
10
```

5 a)

* *
* * *
* * *

include <iostream>

using namespace std;

int main()

{

int i, j, s;

for (i=1; i<4; i++)

{

for (s=1; s<4; i++)

{

cout << " ";

}

for (j=1; j<4; j++)

{

cout << "*";

cout << " ";

}

cout << "\n";

}

return 0;

}

O/p:

* *
* * *
* * *

b)

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

```
# include <iostream>
using namespace std;
int main()
{
    for (int i = 1; i < 6; i++)
    {
        for (int j = 1; j < i; j++)
        {
            cout << j;
            cout << "\n";
        }
        return 0;
}
```

O/P:

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

②
12345

Experiment - 1

```
1. #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 << " Roll no. " << roll;
    }
};

int main()
{
    student s1;
    student s2;
    s1.accept();
    s1.display();
}
```

O/p:

Enter student name:

Dhwani

Enter roll no.

43

Student name: Dhwani

Roll no.: 43

b) #include <iostream>
using namespace std;
class book:

{

private:

string name;

float price;

public:

float price;

void accept()

{

cout << "Enter book name\n";

cin >> name;

cout << "Enter price\n";

cin >> price;

cout << "Enter no. of pages\n";

cin >> pages;

}

void display()

{

cout << "Book name : " << name;

cout << "\n Price : " << price;

cout << "\n Pages : " << pages;

}

}

int main()

{

book b1, b2;

b1 = a

b1. accept();

b2. accept();

b1. display();

b2. display();

if (b1.price > b2.price)

cout << "Book 1";

else

cout << "Book 2";

}

O/P:

Enter book name:

abc

Enter price:

123

Enter no. of pages:

100

Enter book name:

xyz

Enter price:

456

Enter no. of pages:

500

Book 2

```

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

```

```

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

```

Output:

Enter hours: 1
 Enter minutes: 1
 Enter seconds: 1

Hours: 1
 Minutes: 1
 Seconds: 1

Total seconds: 3661

Ques
 1378

Experiment - 2

```
#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;
    for(i=0; i<5; i++)
    {
        c[i].accept();
    }
}
```

```
max = c[0].pop;
```

```
for (i= 0; i<5; i++)
{
    if (c[i].pop > max)
    {
        max = i;
    }
}
```

```
c[max].display();
```

O/p: Enter Name of city
a

Population 1

Name of city b

Population 2

Name of city c

Population 3

Name of city d

Population 4

Name of city e

Population 5

Name of city c
Population 5

```
2) #include <iostream>
using namespace std;
class account
{
    int acc;
    float bal, total;
    total = 0;
public:
    void accept()
    {
        cout << "Enter Account No.";
        cin >> acc;
        cout << "Enter balance";
        cin >> bal;
    }
    void calculate()
    {
        if (bal >= 5000)
        {
            total = bal * 10 / 1000;
        }
        else
        {
            total = 0;
        }
    }
    void display()
    {
        if (total > 0)
        {
            cout << "Account no." << acc;
            cout << "Balance" << bal;
            cout << "Total" << total;
        }
    }
}
```

```

}
}
}
int main()
{
    account a[10];
    int i;
    for (i = 0; i < 10; i++)
    {
        a[i].calculate();
    }
    for (i = 0; i < 10; i++)
    {
        a[i].display();
    }
}
```

```

3 #include <iostream>
using namespace std;
class staff
{
    string name, post;
public:
    void accept()
    {
        cout << "Enter staff name";
        cin >> name;
        cout << "Enter post";
        cin >> post;
    }
    void display()
    {
        if(post == "HOD" || post == "hod")
        {
            cout << "Staff name" << name;
            cout << "Post" << post;
            cout << name << " is the HOD";
        }
    }
    int main()
    {
        Staff S[5];
        int i;
        for(i = 0; i < 5; i++)
        {
            st[i].accept();
        }
    }

```

```

for(i = 0; i < 5; i++)
{
    S[i].display();
}
return 0;

```

O/P:

Enter staff name : a
 Enter post: sales
 Enter staff name: b
 Enter post: manager
 Enter staff name:c
 Enter post: hod
 Enter staff name:d
 Enter post : teacher
 Enter staff name: e
 Enter post: principle

Staff name: c

Staff post: hod

c is the head of department.

Qn
1318

pointer
to
object

Experiment - 3

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

```
{ String title;
String aName;
int price;
public:
```

```
void info ()
```

```
{ cout << "Enter the book , title,
author name and price ";
cin >> title >> aName >> price;
```

```
}
```

```
void display ()
```

```
{ cout << "Book name " << title;
cout << "Author name " << aName;
cout << "Price " << price;
```

```
}
```

```
int main ()
```

```
{ book * p;
book b1;
p = &b1;
p = info ();
p = display ();
}
```

O/P: Enter book title, author name
and price

ABC

XYZ

100

Book name: ABC

Author name: XYZ

Price : 100

2. This pointer

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

```
{
```

```
int roll; perc
float price;
public:
```

```
void accept (int roll, float perc)
```

```
{
```

this → roll = roll;
this → perc = perc;

```
}
```

```
void display ()
```

```
{
```

```
cout << "Roll no " << roll;
cout << "Marks " << perc;
```

```
}
```

```
int main ()
```

```
{
```

Student s1;

z
S1: accept();
S1: display();

output:

Roll no: 3

Marks: 89.7

Nested class

```
#include <iostream>
using namespace std;
class marks
{
public:
    class percentage
    {
        int m, n;
        float p, i;
        public:
            void accept()
            {
                cout << "Enter the marks you got and marks";
                cin >> m >> n;
            }
            void display()
            {
                i = (float) m / n;
                p = i * 100;
                cout << "Percentage" << p;
            }
    };
}
```

int main()
{
 marks m1;
 marks::percentage p1;
 p1.accept();
 p1.display();
}

z

output: Enter the marks you got and marks

56

60

Percentage: 93.33

Q

1315

Experiment - 4

```
1: #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 display()
    {
        cout << "Value" << value;
    }
    int main()
    {
        Number n1(10), n2(20);
        cout << "Before Swap";
        n1.display();
        n2.display();
        n1.swap(n2);
```

```
cout << "After swap";
n1.display();
n2.display();
return 0;
```

}

Output:

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 accept()
```

```
cout << "Enter 2 numbers";
cin >> a >> b;
```

friend void swap(AB a1);

}

```

    void swap (AB a1)
{
    int temp;
    temp = a1.a;
    a1.a = a1.b;
    a1.b = temp;
    cout << "Values after swapping" << a1.a
        << a1.b;
}

int main ()
{
    AB a1;
    a1.info();
    swap(a1);
}

```

Output:

Enter 2 numbers : 5
4

Values after swapping: 4
5

3. Friend function swap 2 numbers different class.

```

#include <iostream>
using namespace std;
class CB;
class CA;
int numA;
public:
(CA (int val): numA (val) {}}

void disp ()
{
    cout << "Value of Class A" << numA;
}

friend void swap (CA&, CB&);

class CB
{
private:
    int numB;
(CB (int val): numB (val) {}}

void disp ()
{
    cout << "Values in classB" << numB;
}

friend void swap (CA&, CB&)

void swap (CA&a, CB&b)
{
    int temp = a.numA;
    int a.numA = b.numB;
    b.numB = temp;
}

```

```

int main()
{
    CA objA = (10);
    CA objB (20);

    cout << " Before swapping ";
    objA::disp();
    objB::disp();
    swap (objA, objB);
    cout << " After swapping ";
    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 2 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 ca (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 << "Average" << avg;  
}  
int main ()  
{  
    result n;  
    result2 y;  
    n.accept ();  
    y.accept ();  
    cal (n, y);  
}
```

output: Enter marks out of 50: 45
Enter marks out of 50: 46
Average : 45.5

5. Greatest among 2 numbers (Diff class) (friend function)

```
#include <iostream>  
using namespace std;  
class B;  
class A
```

```
{  
    int a;  
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 ()  
{  
    A n;  
    B y;  
    n.accept ();  
    y.acc ();  
    gr (n, y);  
}
```

O/P: Enter value: 10
Enter value: 100
Second value is greater.

Extra friend function:

```
1. #include <iostream>
using namespace std;
class ClassB;
class ClassA
{
    int a;
public:
    Class A (int val) : a(val) {}
    friend int sum (Class A, Class B);
};

class Class B
{
    int b;
public:
    Class B (int val) : b(val) {}
    friend int sum (Class A, Class B);
};

int sum (Class A objA, ClassB objB)
{
    return objA.a + objB.b;
}

int main ()
{
    Class A a(10);
    Class B b(20);
    cout << "sum is" << sum(a,b);
    return 0;
}

Op: sum is 30
```

```
2. #include <iostream>
using namespace std;
class Number
{
    int value;
public:
    Number (int val) : value (val) {}
    void display()
    {
        cout << value;
    }
    friend void swapNo (N1, N2);
};

void swapNo (N1 n1, N2 n2)
{
    int temp = n1.value;
    n1.value = n2.value;
    n2.value = temp;
}

int main ()
{
    Number n1(5), n2(15);
    cout << "Before swap" << n1.display();
    cout << n2.display();
    swapNo (n1, n2);
    cout << "After swap" << n1.display();
    cout << n2.display();
}

Op: Before swap 5
      After swap: 15
```

```

3. #include <iostream>
using namespace std;
class cube;
class box
{
    int volume;
public:
    box (int v): volume(v) {}
    friend void findGreater (Box, Cube);
};

class cube
{
    int volume;
public:
    cube (int v): volume(v) {}
    friend void findGreater (Box, Cube);
};

void findGreater (Box b, cube c)
{
    cout << "Greater value" <<
        ->(b.volume > c.volume)?
            b.volume : (volume) << endl;
}

int main ()
{
    box box (118);
    cube cube (90);
    findGreater (box, cube);
    return 0;
}

```

o/p: Greater value : 118

```

4. #include <iostream>
using namespace std;
class complex
{
    int real, imag;
public:
    complex (int r=0, int i=0): real(r),
        imag(i) {}

    void display () { cout << real << " + "
        imag << "i"; }

    complex add (complex c1, complex c2)
    {
        return complex (c1.real + c2.real,
            c1.imag + c2.imag);
    }

    int main ()
    {
        complex c1(4,5), c2(2,3);
        complex result = add (c1, c2);
        cout << "Sum of complex number";
        result.display ();
        return 0;
    }

```

o/p:
Sum of complex numbers: 6 + 8;

Experiment - 5

1. Sum- default

```
# include <iostream>
using namespace std;
class sum {
    int n, s=0;
public:
    sum ()
```

{

n=10;

}

void display ()

{

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

{

s = s + i;

cout << "Sum is" << s;

}

}

int main ()

{

sum s1;

s1.display ();

}

O/P: Sum is 55

2. Sum-parametrized.

```
#include <iostream>
using namespace std;
class sum {
    int no, s=0;
public:
    sum (int n)
    {
        no = n;
    }
    void display ()
    {
        for (int i=0; i<=no; i++)
            s = s+i;
        cout << "Sum is" << s;
    }
};
int main()
{
    sum s1 (10);
    s1.display();
}
```

O/p:

Sum is 55

3. Sum-copy constructor.

```
#include <iostream>
using namespace std;
class sum {
    int no, s=0;
public:
    sum (int n)
    {
        no = n;
    }
    sum (sum &s1)
    {
        no = s1.no;
    }
    void display ()
    {
        for (int i=0; i<=no; i++)
            s = s+i;
        cout << "Sum is" << s;
    }
};
```

u. WAP to declare class student having data members as percentage. Initialize using constructor.

* Default:

```
#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 display()

```
{
```

~~float perc = (float) (m1 + m2) / total * 100;
cout << "Student Name" << name;
cout << "Percentage" << perc;~~

int main()

```
{ Student s1;
s1.display();
```

Student Name ABC
Percentage 81

2. Student - Parameterized.

```
#include <iostream>
using namespace std;
class Student
{
    string name;
    int m1, m2;
    int total;
public:
    Student(string n, int m1, int m2, int total)
    {
        name = n;
        m1 = m1;
        m2 = m2;
        total = total;
    }
}
```

name = n,
m1 = m1,
m2 = m2;
total = total;

void display()

```
{ float perc = (float) (m1 + m2) / total * 100;
```

```
cout << "Student Name" << name;
cout << "Percentage" << perc;
```

Ques

12/11

Experiment - 6

1) Single Inheritance

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

```
{ protected:
```

```
    string name;
    int age;
```

```
}
```

```
class student : Protected Person
```

```
{
```

```
private:
```

```
    int roll;
```

```
public:
```

```
void accept()
```

```
{
```

```
    cout << "Enter name";
    cin >> name;
    cout << "Enter age";
    cin >> age;
    cout << "Enter roll no.";
    cin >> roll;
```

```
}
```

```
void display
```

```
{
```

```
    cout << "Name" << name;
    cout << "Age" << age
    cout << "Roll No." << roll;
```

```
}
```

```
int main ()
```

```
{
```

```
    Student s1;
```

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

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

```
    return 0;
```

```
}
```

Op: 1. Enter Name Dhwani
 Enter Age 17
 Enter Roll no 37

Name Dhwani

Age 17

Roll no 37

2) Multiple Inheritance

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

```
{
```

```
protected:
```

```
    string name;
    int marks;
```

```
{
```

```
class sports
```

```
{
```

```
protected:
```

```
    int score;
```

```
{
```