Lab Number:	1
Student Name:	Shruti Gokhale
Roll No :	40

Title:

To Add Two Numbers, Print Number Entered by User, Swap Two Numbers, Check Whether Number is Even or Odd

1.1 Implement using C++

1.2 Implement using Java

Learning Objective:

- Students will be able to write C++ and java program for simple arithmetic operations and take input from user.

Learning Outcome:

- Ability to execute a simple C++ and Java program with and without any inputs to the program.
- Understanding the constructs in C++ and Java.

Course Outcome:

ECL304.1	Understand object-oriented programming concepts and implement using C++ and Java

Theory:

Difference between procedural and object oriented language

Procedural Programming can be defined as a programming model which is derived from structured programming, based upon the concept of calling procedure. Object oriented programming can be defined as a programming model which is based upon the concept of objects.

Procedural Programming	Object oriented programming
In procedural programming, program is divided into small parts called functions.	In object oriented programming, program is divided into small parts called objects.

Faculty: Ms. Deepali Kayande

Procedural Programming follows top down approach.	Object oriented programming follows bottom up approach.
There is no access specifier in procedural programming.	Object oriented programming have access specifiers like private, public, protected etc.
Procedural programming does not have any proper way for hiding data so it is less *secure*.	Object oriented programming provides data hiding so it is more *secure*.
In procedural programming, function is more important than data.	In object oriented programming, data is more important than function.

Application of object orientation

1. Object-Oriented Databases

These databases try to maintain a direct correspondence between the real-world and database objects in order to let the object retain its identity and integrity. They can then be identified and operated upon.

2. Real-Time System Design

Real-time systems inherent complexities that make it difficult to build them. Object-oriented techniques make it easier to handle those complexities. These techniques present ways of dealing with these complexities by providing an integrated framework, which includes schedulability analysis and behavioral specifications.

3. Client-Server Systems

Object-oriented client-server systems provide the IT infrastructure, creating Object-Oriented Client-Server Internet (OCSI) applications.

Brief introduction to C++ and Java

C++: C++ is platform-dependent. C++ was designed for systems and applications programming. It was an extension of the C programming language. C++ was designed for systems and applications programming. It was an extension of the C programming language. C++ was designed for systems and applications programming. It was an extension of the C programming language. C++ was designed for systems and applications programming. It was an extension of the C programming language.
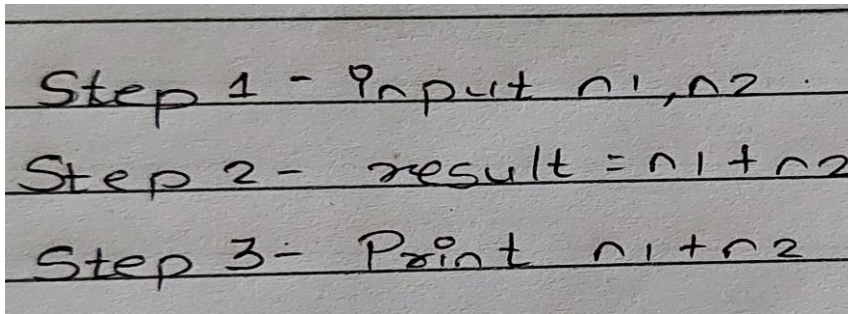
Faculty: Ms. Deepali Kayande

Java: Java is platform-independent. Java was designed and created as an interpreter for printing systems but later extended as a support network computing. It was designed to be easy to use and accessible to a broader audience. Java doesn't support multiple inheritance through class. It can be achieved by using interfaces in java. Java supports pointer internally. However, you can't write the pointer program in java. It means java has restricted pointer support in java.

C++ PROGRAMS

1. TO ADD TWO NUMBERS
ALGORITHM:
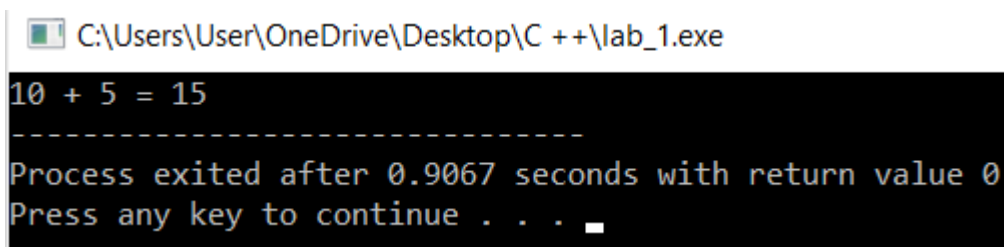
Step 1 - Input n1, n2.
Step 2 - result = n1 + n2
Step 3 - Print n1 + n2

PROGRAM:

```
//To Add Two Numbers
#include<iostream>
using namespace std;
int main()
{
        int n1,n2,result;
        n1=10;
        n2=5;
        result=n1+n2;
        cout << n1 << " + " << n2 << " = " << result;
        return 0;
}
```
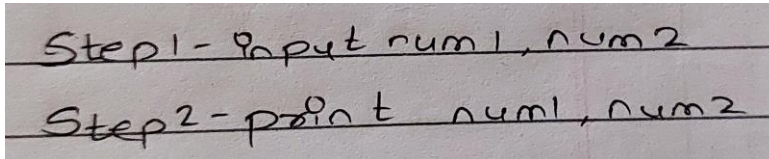
OUTPUT SCREENSHOT:

```
C:\Users\User\OneDrive\Desktop\C ++\lab_1.exe

10 + 5 = 15
--------------------------------
Process exited after 0.9067 seconds with return value 0
Press any key to continue . . .
```

Faculty: Ms. Deepali Kayande

2. TO PRINT NUMBERS ENTERED BY USER
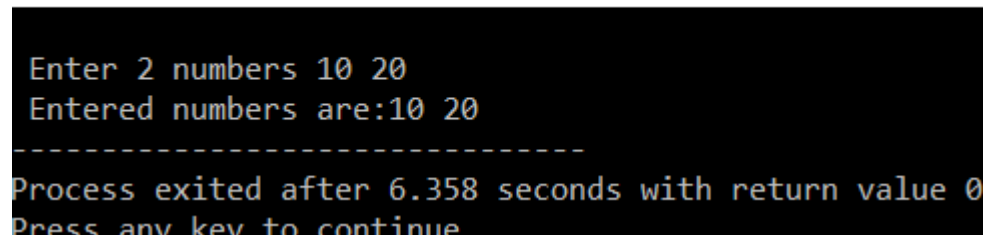**ALGORITHM:**

Step1 - Input num1, num2

Step2 - print num1, num2

PROGRAM:
```cpp
// Print Number Entered by User
#include<iostream>
using namespace std;
int main()
{
int num1,num2;
        cout<<"\n Enter 2 numbers";
        cin>>num1>>num2;
        cout<< " Entered numbers are:" << num1<< " " << num2;
        return 0;
}
```

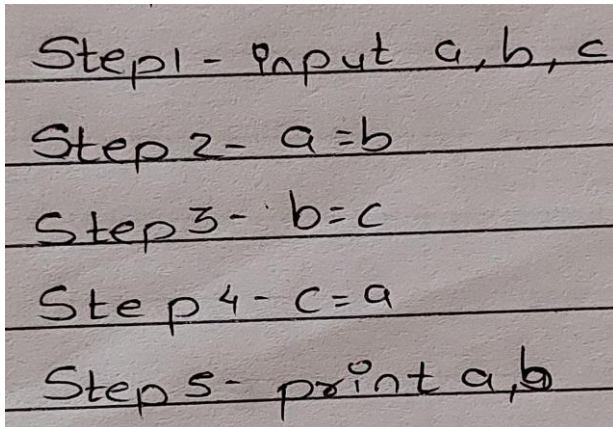**OUTPUT SCREENSHOT:**
C:\Users\User\OneDrive\Desktop\C ++\lab_1.exe

```
Enter 2 numbers 10 20
Entered numbers are:10 20
------------------------------
Process exited after 6.358 seconds with return value 0
Press any key to continue
```

3. TO SWAP TWO NUMBERS
ALGORITHM:

Step 1 - Input a, b, c

Step 2 - a = b

Step 3 - b = c

Step 4 - C = a

Step 5 - print a, b

PROGRAM:

```
//Swap Two Numbers

#include<iostream>
using namespace std;
int main()
{
        int a,b,c;

        cout<<"\n Enter two no to swap:";

        cin>>a>>b;

        c=a;

        a=b;

        b=c;

        cout<<"\n Swapping the numbers: \n";

        cout<<" a= "<<a<<" b= "<<b;

        return 0;

}
```
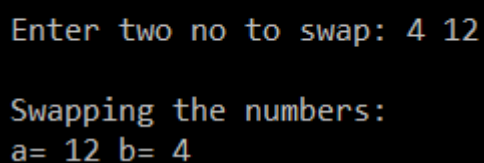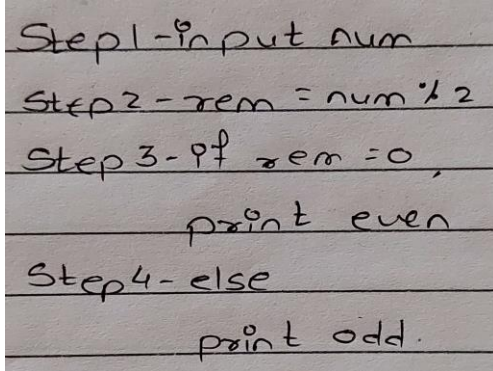
OUTPUT SCREENSHOT:

C:\Users\User\OneDrive\Desktop\C ·

```
Enter two no to swap: 4 12

Swapping the numbers:
a= 12 b= 4
```

Faculty: Ms. Deepali Kayande

4. TO CHECK WHETHER NUMBER IS EVEN OR ODD
ALGORITHM:


```
Step1 - input num
Step2 - rem = num % 2
Step3 - if rem = 0,
            print even
Step4 - else
            print odd.
```

PROGRAM:

```cpp
//To check whether no is even or odd

#include<iostream>
using namespace std;
int main()
{
        int num=5;

        cout<<"\n Number is";

        cin>>num;


        if ( num % 2 == 0)

        cout<<num<<" is even";

        else

        cout<<num<<" is odd";

}
```
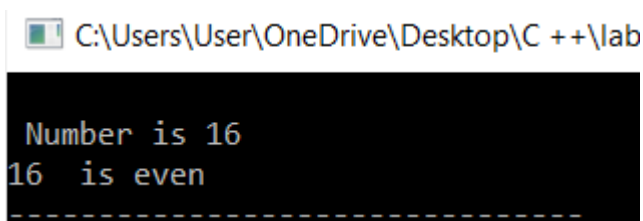
OUTPUT SCREENSHOT:


```
C:\Users\User\OneDrive\Desktop\C ++\lab

Number is 16
16  is even
---------------------------------
```

Faculty: Ms. Deepali Kayande