


LAB MANUAL - (C++)



Copyright © 2019, Dr. G.D. Kurundkar and Mr. N.A. Naik
All rights reserved.

No part of this publication may be reproduced or transmitted
in any form or by any means, electronic or mechanical,
including photocopy, recording or any information storage and
retrieval system now known or to be invented, without permission in
writing from the publisher, except by a reviewer who wishes to quote
brief passages in connection with a review written for inclusion in a
magazine, newspaper or broadcast.

Published in India by Prowess Publishing,
YRK Towers, Thadikara Swamy Koil St, Alandur,
Chennai, Tamil Nadu 600016

ISBN-10: 1-5457-4356-8
ISBN-13: 978-1-5457-4356-0

Library of Congress Cataloging in Publication

2.	Write a C++ program to check whether the input number is positive number or negative number.	16
3.	Write a C++ program to check whether the input number is an even or an odd number.	17
4.	Write a C++ program to check whether the number is prime or not.	18
5.	Write a C++ program to print prime numbers between 1 to given number n.	19
6.	Write a C++ program to print reverse number.	20
7.	Write a C++ program to check whether the entered character is an alphabet or not.	20
8.	Write a C++ program for swapping two numbers.	21
9.	Write a C++ program to swap two numbers without using third variable.	22
10.	Write a C++ program to check whether the entered character is a vowel or not.	22
11.	Write a C++ program to display the student's grade according to the marks held by a student. The marks scored are taken as input and the class is displayed according to the following range.	23
12.	Write a C++ program to check whether the input year is a leap year or not.	25
13.	Write a C++ program to check whether the entered number is palindrome or not.	26
14.	Write a C++ program to check the given number is Armstrong number or not.	27

INDEX

15.	Write a C++ program to addition of digits of any given number.	28
16.	Write a C++ program to find the largest number of two given numbers.	28
17.	Write a C++ program to find the largest number among the three numbers.	29
3. C++ Programs and Examples on Arrays		31
1.	Introduction to Arrays Declaration of Arrays Advantages of Arrays Disadvantages of Arrays Types of Arrays	31
2.	Write a C++ program to print one dimensional array	33
3.	Write a C++ program to calculate average and percentage of marks for student.	34
4.	Write a C++ program to calculate arithmetic mean of numbers.	35
5.	Write a C++ program to calculate grade of a student on the basis of his/her total marks.	36
6.	Write a C++ program to search any element in an array.	37
7.	Write a C++ program to find the largest element in an array.	38
8.	Write a C++ program to reverse an array.	39
9.	Write a C++ program to convert given Decimal Number to Binary.	40
10.	Write a C++ program to convert given Decimal Number to Octal.	41

Lab Manual - (C++)

11.	Write a C++ program to convert given Decimal Number to Hexadecimal	42
12.	Write a C++ program to print two dimensional array.	42
13.	Write a C++ program to print the addition of two matrices.	44
14.	Write a C++ program to print the subtraction of two matrices.	45
15.	Write a C++ program to print the multiplication of two matrices.	47
16.	Write a C++ program to display the transpose of matrix.	48
17.	Write a C++ program to accept 10 numbers and display the numbers by sorting in descending order.	50
4. C++ Programs using Functions		52
1.	Introduction to Functions in C++ Types of functions in C++: Advantages of Functions Define functions Parameters passing in C++	52
2.	Write a C++ program to calculate area and perimeter of square and rectangle using function.	54
3.	Write a C++ program to calculate area and circumference of circle using function.	55
4.	Write a C++ program to print the Fibonacci series using recursion function.	56
5.	Write a C++ program to swap two numbers using call-by-value method.	57

INDEX

6.	Write a C++ program to swap two numbers using call-by-reference method.	58
7.	Write a C++ program to implement bubble sort algorithm for sorting the numbers in ascending order.	59
8.	Write a C++ program using class to find the greater one of two integer numbers using Inline function, conditional operator and default argument.	60
9.	Write a C++ program to find the sum of digits of a number using recursive function.	61
10.	Write a C++ program to accept records of 'n' employee and store it in an array. Class employee contains eno, ename[20], salary attributes. Overload search() function as follows	62
5. C++ Programs using Strings		66
1.	Introduction to String Define String String Functions	66
2.	Write a C++ program to compare two strings.	67
3.	Write a C++ program to copy one string to another string	68
4.	Write a C++ program to concatenate one string to another string.	69
5.	Write a C++ program to reverse a string.	70
6.	Write a C++ program to delete all the vowels from the string.	70
7.	Write a C++ program to count total number of words used in any sentence.	71

Lab Manual - (C++)

8.	Write a C++ program to sort the strings in alphabetical order.	72
9.	Write a C++ program to convert a character from uppercase to lowercase.	73
10.	Write a C++ program to convert a character from lowercase to uppercase.	74
11.	Write a C++ program to check whether the entered string is palindrome or not.	75
12.	Write C++ program to concatenate the two strings without using library function.	76
6. C++ Programs using Pointers		77
1.	Introduction to Pointers Examples of Declaring pointers Applications of pointers Declaration of Pointers	77
2.	Write a C++ program to accept the set of 5 numbers and print the numbers using pointer.	78
3.	Write a C++ program to display the addition of two numbers using pointer.	79
4.	Write a C++ program to allocate memory dynamically for an integer, initialize it to 10 and free that memory.	80
7. C++ Programs using Structure		81
1.	Introduction to Structure Define Structure	81
2.	Write a C++ program to accept and display the name, roll number and fees of a student using structure.	82

INDEX

3.	Write a C++ program to accept and display the name, marks in three subjects and roll number of 'n' students using structure. Display the output in tabular form.	83
4.	Write a C++ program to read roll number, name of 'n' students and display the data in descending order of roll numbers using structure.	85
5.	Write a C++ program to accept bill no, customer code, name and amount of 100 bills and print the details of the customer with highest billed amount.	87
8. C++ Programs on Patterns		91
1.	Write a C++ program to print the half pyramid. * * * * * * * * * * * * * * *	91
2.	Write a C++ program to draw the following pattern: *	92

3.	Write a C++ program to print the following pattern of natural numbers: 1 2 3 4 5 6 7 8 9 10	93
4.	Write a C++ program to print the following pattern: 1 1 2 1 2 3 1 2 3 4 1 2 3 4 5	94
5.	Write a C++ program to print the following pattern: 1 3 2 6 5 4 10 9 8 7	95
6.	Write a C++ program to print the following pattern: A BC DEF GHIJ KLMNO	96

INDEX

7.	Write a C++ program to print the following pattern: A BB CCC DDDD EEEEEE	97
9. C++ Programs on Conversions		98
1.	Write a C++ program to convert decimal number to binary number.	98
2.	Write a C++ program to convert decimal number to octal number.	99
3.	Write a C++ program to convert decimal number to hexadecimal number.	100
4.	Write a C++ program to convert binary number to decimal number.	101
5.	Write a C++ program to convert octal number to decimal number.	102
10. C++ Programs using Classes and Objects		103
1.	Introduction to Classes and Objects Classes and Objects Syntax for objects in C++	103
2.	Write a C++ program to find the area of circle using class circle which have following details: Accept radius from the user, Calculate the area, Display the result.	105

3.	Write a C++ program to define a class employee having members Emp-id, Emp-name, basic salary and functions accept() and display(). Calculate DA=25% of basic salary, HRA=800, I-tax=15% of basic salary. Display the pay slip.	106
4.	Write a C++ program to find the factorial of a number using copy constructor.	108
5.	Create a C++ class for participant object with the following attributes participant no., name, number of matches and number of goals done in each match. The number of matches varies for each participant. Write parameterized constructor which initializes participant no., name, number of subjects and creates array for number of goals and number of matches dynamically.	109
6.	Write a program to demonstrate the destructor.	111
11. C++ Programs using Operator Overloading		113
1.	Introduction to Overloading Function overloading Operator overloading	113
2.	Write a C++ program to overload unary operators that is increment and decrement. i.e. ++ and -- (Operator overloading is a concept of polymorphism)	115
3.	Write a C++ program to overload binary operator '+' to add two complex numbers.	116
4.	Write a C++ program to add two complex numbers using operator overloaded by a friend function.	118

INDEX

5.	Write a C++ program to overload '+' operator to concatenate two strings.	119
12. C++ Programs using concept of Inheritance		121
1.	Introduction to inheritance Types of inheritances	121
2.	Write a C++ program to add two numbers using single inheritance. Accept these two numbers from the user in base class and display the sum of these two numbers in derived class.	122
3.	Write a C++ program to calculate the percentage of a student using multi-level inheritance. Accept the marks of three subjects in base class. A class will derived from the above mentioned class which includes a function to find the total marks obtained and another class derived from this class which calculates and displays the percentage of student.	124
13. C++ Programs using Templates		126
1.	Introduction to Templates Template can be used in two ways	126
2.	Write a C++ program to demonstrate the addition of multiple types of data using generic function or template.	127
3.	Write a C++ program to find largest of three elements using template.	128

14. C++ Programs on Exception Handling		130
1.	Introduction to Exception Handling	130
2.	Write a C++ program to demonstrate the use of try, catch block with the argument as an integer and string using multiple catch blocks.	132
3.	Write a C++ program to demonstrate try, throw and catch statement.	133

CHAPTER

1

Basic C++ Programs

1. Introduction

The object-oriented paradigm is initial concept of a new programming strategy. The first object-oriented language was Simula (Simulation of real systems) that was developed in 1960 by researchers at the Norwegian Computing Center.

- ✓ In 1970, Alan Kay and his team mates at Xerox PARK created a personal computer named Dynabook and the first pure object-oriented programming language (OOPL) Smalltalk, for programming the Dynabook.
- ✓ In the 1980s, Grady Booch published a paper titled Object Oriented Design that presented a design for the programming language, Ada. In the ensuing editions, he extended his ideas to a complete object-oriented design method.
- ✓ In the 1990s, Coad incorporated behavioral ideas to object-oriented methods.

Other significant innovations were Object Modelling Techniques (OMT) by James Rumbaugh and Object-Oriented Software Engineering (OOSE) by Ivar Jacobson.

2. Object-Oriented Analysis

Object-Oriented Analysis (OOA) is the mechanism of identifying software engineering requirements and developing software specifications in terms of a software system's object model.

The main difference between object-oriented analysis and other forms of analysis is that in object oriented model it interacts with both the data and functions. So in object oriented model function and data are treated separately.

The primary tasks in object-oriented analysis (OOA) are –

- ✓ Identifying objects
- ✓ Organizing the objects by creating object model diagram
- ✓ Defining the object attributes
- ✓ Defining the behavior of the objects, i.e., object actions
- ✓ Describing how the objects interact

The common models used in OOA are use cases and object models.

3. Object-Oriented Programming

Object-oriented programming (OOP) is a programming paradigm based upon objects (having both data and functions) that aims to incorporate the advantages of modularity and reusability. Objects, which are usually instances of classes, are used to interact with one another to design applications and computer programs.

The important features of object-oriented programming are –

- ✓ Bottom-up approach of programming
- ✓ Programs are grouped in classes and objects
- ✓ Focus on data with functions to operate upon object's data
- ✓ Functions are used to interact with objects.

Some examples of object-oriented programming languages are C++, Java, Smalltalk, Delphi, C#, Perl, Python, Ruby, and PHP.

4. Basic Concept of OOPs

The core of the pure object-oriented programming is to create an object, in code, that has certain data and methods. C++ is in the form of objects. For example a dog is an object which has certain properties

such as color, legs, and the sound. It also has certain methods such as jump, bark, and so on.

The few principle concepts of OOP are as follows

Object

This is the basic concept of object oriented programming. That is the user can access data and functions using object.

Class

Class is blueprint of object that is when you define a class it contains data and functions all together that to provide information.

Abstraction

Data abstraction refers to, providing only essential information to the outside world and hiding their background details, i.e., to represent the needed information in program without presenting the details.

For example, a database system hides certain details of how data is stored and created and maintained. Similar way, C++ classes provides different methods to the outside world without giving internal detail about those methods and data.

Encapsulation

Encapsulation means encapsulates the things in a capsule manner. That is C++ provides the facility to use the functions and data all together in an systematic manner.

Inheritance

Code reusability is one of the most useful aspects of object-oriented programming. As the name indicates it is an procedure or approach to design an new class from the other one where the new class is called as derived class and the class from where it is created is called as base class.

Polymorphism

The ability to use an operator or function in different ways in other words giving different meaning or functions to the operators or functions is called polymorphism. Poly refers to many. That is a single

function or an operator functioning in many ways different upon the usage is called polymorphism.

Overloading

The concept of overloading means using same thing for different purpose. C++ provides the concept of Function overloading and operator overloading where same operators and functions are used for different operations.

5. Benefits of OOPs

The benefits of OOP are as follows:

- ✓ OOP provides a clear modular structure for programs.
- ✓ It is good for defining abstract data types.
- ✓ The visibility modes are provide security for data.
- ✓ It is easy to maintain and modify existing code as new objects can be created with small differences to existing ones.
- ✓ objects, methods, inheritance are some important things provided by these particular languages.

Applications of OOPs

- ✓ Now days many software engineers are using oops as their development tool.
- ✓ The most popular application of object-oriented programming, up to now, has been in the area of user interface design such as window.
- ✓ Hundreds of windowing systems have been developed, using the OOP techniques.
- ✓ Real-business system are often much more complex and contain many more objects with complicated attributes and method. OOP is useful in these types of application because it can simplify a complex problem. The promising areas of application of OOP include:

A Simple C++ Program

```
#include<iostream.h> // for basic statements
#include<conio.h> // for clrscr() or getch()
like functions

void main() // main function

{

cout <<"Hello College Student"; // statement
to display Hello Student

}
```

Output of above program is

Hello College Student.

6. Structure of C++ Program

Programs are a sequence of instructions or statements. These statements form the structure of a C++ program. C++ program structure is divided into various sections, namely, *headers*, *class definition*, *member functions definitions* and *main function*.

- ✓ C++ Headers // Include functions or Definitions
- ✓ Class Definition // Classes which are used in C++ Program
 - Declaration of variables and function
- ✓ main function // program

Note that it is possible to write a C++ program without using a class you can use only main functions to write a program.

Comments

You can use single line comment with // and multiline comment with /* and */ in C++

Write a C++ program to display “Welcome to C++ Programs” on the screen.

```
#include<iostream.h>
void main()
{
    cout<<“\n Welcome to C++ Programs !!!”;
}
```

Output:

Welcome to C++ Programs !!!

Write a C++ program to accept a string from the user and display it on the screen.

```
#include<iostream.h>
#include<conio.h>
void main()
{
    char str[100];
    cout<<“\n Enter Your Name : ”;
    cin>>str;
    cout<<“\n Hello ”<<str<<“... \n Welcome
to C++ Programs ”;
    getch();
}
```

Output:

Enter Your Name : Sachin

Hello Sachin

Welcome to C++ Programs

You've Just Finished your Free Sample

Enjoyed the preview?

Buy: <https://store.prowesspub.com>