LAB MANUAL - (C++)

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

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 Definations
- ✓ Class Defination // 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