

# NEERAJ<sup>®</sup>

---

## PROGRAMMING IN C++

---

By:

*Varsha Bhatia* B.E., M.C.A.

*Reference Book*  
*Including*  
*Solved Question Papers*

---

*New Edition*

---



### NEERAJ PUBLICATIONS

*( Publishers of Educational Books )*

*( An ISO 9001 : 2008 Certified Company )*

**1507, 1st Floor, NAI SARA, DELHI - 110006**

Ph: 011-23260329, 45704411, 23244362, 23285501 Off. Mob. : 8510009878

*E-mail: info@neerajignoubooks.com*

*Website: www.neerajignoubooks.com*

*Price*

**₹ 240/-**

**Published by:**

**NEERAJ PUBLICATIONS**

Admn. Office : **Delhi-110 007**

Sales Office : **1507, 1st Floor, Nai Sarak, Delhi-110 006**

E-mail: [info@neerajignoubooks.com](mailto:info@neerajignoubooks.com) Website: [www.neerajignoubooks.com](http://www.neerajignoubooks.com)

**Reprint Edition** | Typesetting by: **Competent Computers** | Printed at: **Novelty Printer**

**Notes:**

1. For the best & upto-date study & results, please prefer the recommended textbooks/study material only.
2. This book is just a Guide Book/Reference Book published by NEERAJ PUBLICATIONS based on the suggested syllabus by a particular Board /University.
3. The information and data etc. given in this Book are from the best of the data arranged by the Author, but for the complete and upto-date information and data etc. see the Govt. of India Publications/textbooks recommended by the Board/University.
4. Publisher is not responsible for any omission or error though every care has been taken while preparing, printing, composing and proof reading of the Book. As all the Composing, Printing, Publishing and Proof Reading etc. are done by Human only and chances of Human Error could not be denied. If any reader is not satisfied, then he is requested not to buy this book.
5. In case of any dispute whatsoever the maximum anybody can claim against NEERAJ PUBLICATIONS is just for the price of the Book.
6. If anyone finds any mistake or error in this Book, he is requested to inform the Publisher, so that the same could be rectified and he would be provided the rectified Book free of cost.
7. The number of questions in NEERAJ study materials are indicative of general scope and design of the question paper.
8. Question Paper and their answers given in this Book provide you just the approximate pattern of the actual paper and is prepared based on the memory only. However, the actual Question Paper might somewhat vary in its contents, distribution of marks and their level of difficulty.
9. Any type of ONLINE Sale/Resale of "NEERAJ BOOKS/NEERAJ IGNOU BOOKS" published by "NEERAJ PUBLICATIONS" on Websites, Web Portals, Online Shopping Sites, like Amazon, Flipkart, Ebay, Snapdeal, etc. is strictly not permitted without prior written permission from Neeraj Publications. Any such online sale activity by an Individual, Company, Dealer, Bookseller, Book Trader or Distributor will be termed as ILLEGAL SALE of Neeraj IGNOU BOOKS/NEERAJ BOOKS and will invite a legal action against the offenders.
10. Subject to Delhi Jurisdiction only.

**© Reserved with the Publishers only.**

**Spl. Note:** This book or part thereof cannot be translated or reproduced in any form (except for review or criticism) without the written permission of the publishers.

## **How to get Books by Post (V.P.P.)?**

If you want to Buy NEERAJ IGNOU BOOKS by Post (V.P.P.), then please order your complete requirement at our Website [www.neerajignoubooks.com](http://www.neerajignoubooks.com). You may also avail the 'Special Discount Offers' prevailing at that Particular Time (Time of Your Order).

To have a look at the Details of the Course, Name of the Books, Printed Price & the Cover Pages (Titles) of our NEERAJ IGNOU BOOKS You may Visit/Surf our website [www.neerajignoubooks.com](http://www.neerajignoubooks.com).

No Need To Pay In Advance, the Books Shall be Sent to you Through V.P.P. Post Parcel. All The Payment including the Price of the Books & the Postal Charges etc. are to be Paid to the Postman or to your Post Office at the time when You take the Delivery of the Books & they shall Pass the Value of the Goods to us by Charging some extra M.O. Charges.

We usually dispatch the books nearly within 4-5 days after we receive your order and it takes Nearly 5 days in the postal service to reach your Destination (In total it take atleast 10 days).



**NEERAJ PUBLICATIONS**

(Publishers of Educational Books)

(An ISO 9001 : 2008 Certified Company)

**1507, 1st Floor, NAI SARAK, DELHI - 110 006**

**Ph. 011-23260329, 45704411, 23244362, 23285501, Off. Mob. : 8510009878**

E-mail: [info@neerajignoubooks.com](mailto:info@neerajignoubooks.com) Website: [www.neerajignoubooks.com](http://www.neerajignoubooks.com)

# CONTENTS

## PROGRAMMING IN C++

<i>Question Paper—June, 2017 (Solved)</i>	1-8
<i>Question Paper—June, 2016 (Solved)</i>	1-7
<i>Question Paper—June, 2015 (Solved)</i>	1-2
<i>Question Paper—June, 2014 (Solved)</i>	1-10
<i>Question Paper—December, 2013 (Solved)</i>	1-11
<i>Question Paper—December, 2012 (Solved)</i>	1-14

<i>S.No.</i>	<i>Chapter</i>	<i>Page</i>
--------------	----------------	-------------

### **BASICS OF OBJECT ORIENTED PROGRAMMING & C++**

1. Object Oriented Programming	1
2. Introduction to C++	11
3. Objects and Classes	37
4. Constructors and Destructors	54

### **INHERITANCE AND POLYMORPHISM IN C++**

5. Inheritance	68
6. Operator Overloading	82
7. Polymorphism and Virtual Function	92

### **ADVANCED FEATURES OF C++**

8. Streams and Files	103
----------------------	-----

<i>S.No.</i>	<i>Chapter</i>	<i>Page</i>
9.	Templates and STL	123
10.	Exception Handling	138
11.	Case Study	148
		■ ■

**Sample Preview  
of the  
Solved  
Sample Question  
Papers**

*Published by:*



**NEERAJ  
PUBLICATIONS**

[www.neerajbooks.com](http://www.neerajbooks.com)

# QUESTION PAPER

(June – 2017)

(Solved)

## PROGRAMMING IN C++

Time: 3 hours ]

[ Maximum Marks: 100

(Weightage : 75%)

Note: Question No. 1 is compulsory and carries 40 marks. Attempt any three questions from the rest.

**Q.1. (a) Why are object oriented programming languages more popular than structured programming languages? Differentiate between structured and object oriented programming languages.**

**Ans. Ref.:** See Chapter-1, Page No. 2, 'Structured vs. Object Oriented Programming' and Page No. 3, 'Difference between Structured and Object Oriented Programming'.

**(b) Explain ambiguity resolution in multiple inheritance. What happens if we don't use a virtual function in the inheritance?**

**Ans.** In multiple inheritance, there may be possibility that a class may inherit member functions with same name from two or more base classes and the derived class may not have functions with same name as those of its base classes. If the object of the derived class need to access one of the same named member function of the base classes then it result in ambiguity as it is not clear to the compiler which base's class member function should be invoked. The ambiguity simply means the state when the compiler confused.

Syntax:

```
class a
{
private:
.....
public:
void abc() {}
.....
.....
};
```

```
class b
{
private:
.....
public:
void abc() {}
.....
};
class c : public a , public b
{
private:
.....
public:
.....
};
void main()
{
c obj;
obj.abc();//Error
.....
}
```

In this example, the two base classes a and b have function of same name abc() which are inherited to the derived class c. When the object of the class c is created and call the function abc() then the compiler is confused which base's class function is called by the compiler.

The ambiguity can be resolved by using the scope resolution operator to specify the class in which the

member function lies as given below:

```
obj.a :: abc();
```

This statement invoke the function name abc() which are lies in base class a.

Virtual keyword can be used with classes as well as methods. When used with classes, it avoids ambiguity in inheritance. When it is used with methods, it allows method overriding and thus supports run-time polymorphism.

**(c) Write a C++ program to create a matrix class. Define constructor and destructor for this class. Also define a method to find the sum of two matrices.**

**Ans.** Create a class called MATRIX using two-dimensional array of integers. Implement the following operations by overloading the operator ++ which checks the compatibility of two matrices to be added and subtracted. Perform the addition and subtraction by overloading the + and - operators respectively. Display the results by overloading the operator <<.

If (m1 == m2) then m3 = m1 + m2 and m4 = m1 - m2 else display error.

Constructors are special class functions which performs initialization of every object. The compiler calls the Constructor whenever an object is created. Constructors initialize values to object members after storage is allocated to the object.

Destructor is a special class function which destroys the object as soon as the scope of object ends. The destructor is called automatically by the compiler when the object goes out of scope.

The syntax for destructor is same as that for the constructor, the class name is used for the name of destructor, with a tilde ~ sign as prefix to it.

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

class matrix
{
int **p, m, n;
public:
matrix(int row = 2, int col = 2)
{
m = row;
n = col;
p = new(int *); m;
for (int i = 0; i < m; i++)
```

```
p[i] = new int[n];
}
~matrix()
{
for (int i = 0; i < m; i++)
delete p[i];
delete p;
}
void accept()
{
cout << "Enter matrix elements:";
for (int i = 0; i < m; i++)
{
for (int j = 0; j < n; j++)
{
cin >> p[i][j];
}
}
}
void display()
{
cout << "The matrix is:";
for (int i = 0; i < m; i++)
{
cout << endl;
for (int j = 0; j < n; j++)
{
cout << p[i][j] << " ";
}
}
}
matrix operator +(matrix m2)
{
matrix T(m, n);
for (int i = 0; i < m; i++)
{
for (int j = 0; j < n; j++)
{
T.p[i][j] = p[i][j] + m2.p[i][j];
}
}
return T;
}
friend matrix operator * (matrix, matrix);
};
matrix operator * (matrix a, matrix b)
{
if(a.n == b.m)
```

# **Sample Preview of The Chapter**

*Published by:*



**NEERAJ  
PUBLICATIONS**

[www.neerajbooks.com](http://www.neerajbooks.com)



# PROGRAMMING IN C++

## BASICS OF OBJECT ORIENTED PROGRAMMING & C++

1

### Object Oriented Programming

#### INTRODUCTION

Object Oriented Programming is the modern programming approach, it offers a new and popular way to minimize software errors and reduce complexity of programs. Since the invention of the computer different programming approaches have been tried such as procedural programming, modular programming, structural programming etc.

The basic question that comes to our mind was “what is the need and importance for approaches?”

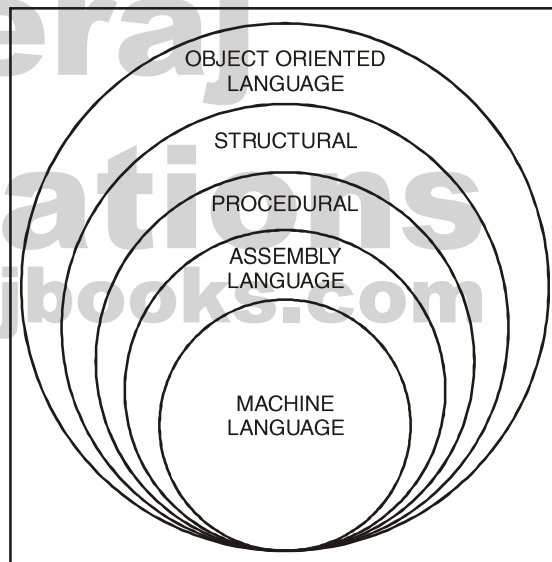
With the evolution of software industry, new tools and technique are announced in quick succession. The main requirement today is secure, reliable, efficient, reusable maintainable software. To build today’s complex software it is just not enough to put together a sequence of programming statement and set of procedures and modules, we need to use techniques that make our software easy to implement, maintain and modify. Based on these techniques different models are build, which specify the guidelines to design and write software programs .These models are known as Programming Paradigms.

**Definition:** Paradigm means organizing principle of a program.

**Paradigms of Programming Languages :** The Paradigms of Programming Language gives the model to the programmer to write the programs. The different Paradigms of programming languages are:

1. Unstructured Programming or Monolithic Programming

2. Procedural Programming
3. Structural Programming
4. Object Oriented Programming



#### Monolithic Programming:

- In this whole problem is solved as a single block.
- All the data are global (common for all) and there is no security.
- To share the codes jumps are allowed and so contain lot of go to statements.
- This is suitable only for small problem.
- Difficult to follow and correct errors
- E.g. Assembly Language, BASIC.

2 / NEERAJ : PROGRAMMING IN C++

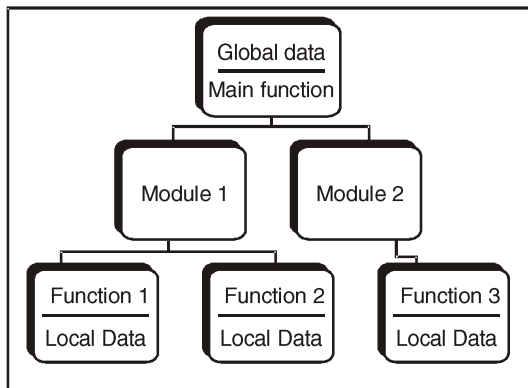
**Procedural Programming:**

- The given problem is divided into a number of sub-problems depending upon its functionality.
- The sub-problems are called procedures or methods.
- Any procedure can be called at any point during the program execution.
- The program has global and local variables.

**Structural Programming:**

- The program is divided into modules and the modules are then divided into functions.
- The usage of goto statement is removed or reduced.
- Each module can work independent of one another.

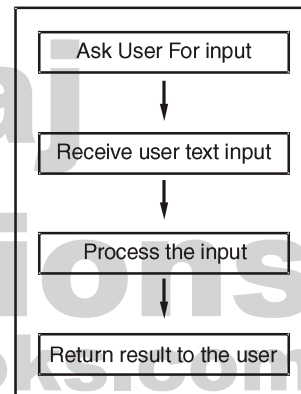
**Example:** ADA, C.



by the programmer's tendency to divide his program source code into logically structured blocks which would normally consist of conditional statements, loops and logic blocks. This style of programming has the implementation of the source code being processed in the order in which bits of the code have been typed in.

A structured program is made up of simple program flow structures, which are hierarchically organized. They are sequence, selection and repetition. Sequence is an order of statements. Selection means selecting a statement from a set of statements based on the current state of the program (for e.g. using if statements) and repetition means executing a statement until a certain state is reached (for e.g. using for or while statements). ALGOL, Pascal, Ada and PL/I are some of the structured programming languages used today.

**Structured Programming**



**CHAPTER AT A GLANCE**

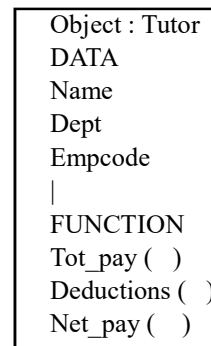
**STRUCTURED VS. OBJECT-ORIENTED PROGRAMMING**

Object Oriented Programming (OOP) and Structured Programming are two programming paradigms. A programming paradigm is a fundamental style of computer programming. Programming paradigms differ in how each element of the programs is represented and how steps are defined for solving problems. As the name suggests, OOP focuses on representing problems using real-world objects and their behaviour, while Structured Programming deals with organizing the program in a logical structure.

Structured programming can be defined as a Software application programming technique that follows a top down design approach with block oriented structures. This style of programming is characterized

**OBJECT ORIENTED PROGRAMMING**

An object oriented language combines the data to its function or code in such a way that access to data is allowed only through its function or code. Such combination of data and code is called an object. For example, an object called Tutor may contain data and function or code as shown below:



The data part contains the Name, Dept. and Employee code. The function part consists of three functions: To\_pay ( ), Deductions ( ) and Net\_pay ( ). Thus, the various objects in the object-oriented language interact with each other through their respective codes or functions.

It may be noted here that the data of an object can be accessed only by the functions associated with that object. However, functions of one object can access the functions of other objects.

**Difference Between Structural and Object Oriented Programming**

Structural Programming	Object Oriented Programming
Program is divided into sub-modules.	Program is organized as classes and their objects.
Data and function are two separate entities.	Data and function are single entity.
Algorithm is given importance.	Data is given importance.
Functions are dependent on each other.	Classes are related in hierarchical manner.
Maintenance is costly.	Maintenance is relatively cheaper.
Software reuse is difficult.	Software reuse is easily possible.

**OBJECT ORIENTED PROGRAMMING CONCEPTS**

Following are some of the concepts related to Object Oriented Programming:

**Class:** The class is known as blueprint of data and member functions. It defines what an object will consist of, and what operations can be performed on such object. In simple words class is a group of objects that share common properties and responsibility.

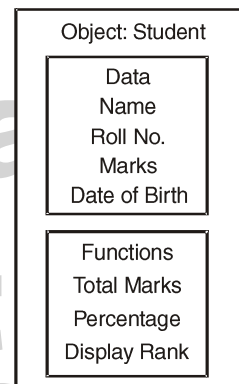
**Object:** An object is an instance of class. Every data member of object contains its own values, where as class only declare data members and functions. Object is an identifiable entity with characteristics and behaviour. Program objects should be resemble closely to the real world objects.

**For Example**

If our problem is to calculate marks of a class of 40 students:

First for a student we know the characteristics of a student is his Name, Roll No., Marks, Date of birth and we need to calculate its total marks in all subjects percentage and want to display its result.

Now based on our problem a generalized class student is declared and for each student present in the class we have to create one object. So if we want to make result of a class of 40 students we will have to create 40 objects of class student.



**Data Abstraction:** The word abstraction means providing only the relevant information & hiding background details. Data abstraction is a programming technique that separates program and interface. Let us consider real life example.

We switch on T.V. change the channel. Apply different settings, but we don't know its internal or background details, that is how the signal is received, translated and displayed on our T.V. screen.

**Data Encapsulation:** It means combining data and function together into single entity is called encapsulation. Data can be accessed only with the help of functions. It cannot be accessed directly. The data is hidden so it is safe from accidental alteration. The function provide the interface between the object's data and program. As we have declared a class student in which we kept together the characteristics as well

4 / NEERAJ : PROGRAMMING IN C++

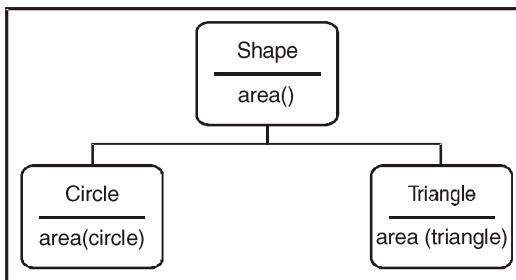
functions/operations which we want to calculate for student together (in the single unit).

**Inheritance:** It is the process of deriving new classes from existing classes. The existing class is called the base class and the new class is called sub-class. Inheritance helps in reducing code size.

**Polymorphism:** It means ability of different object to respond (each in its own way) to the identical message is called polymorphism. Based on the type of data different functions (with same name/signature) are called. It simplifies the programming interface. Instead of reinventing the new name to each function, you add to program, you can reuse the same name.

**For example:** If we declare a class of named shape It can have two sub-classes such as circle and triangle. This shows the property of inheritance both circle and triangle are shapes but have different features. That means they inherit properties of class shapes as well they have few distinct feature of their own.

Here we are using a function named area. which is declared in base or parent class as well as in both the sub-classes with the same name. Based on the type of object. compiler decides which area function will be called. If the object is of Class Circle the function area (circle) will be called and if the object is of Class Triangle the function area (Triangle) will be called. Hence, instead of remembering the names of three different functions, we have to remember the name of one function.



**Modularity:** The modularity of object-oriented programs means that the valid components of big program can be implemented individually. Different people can work on different components. Each execution part is isolated from each other. When this principle is applied it organizes the program and help to detect errors and their location.

**BENEFITS OF Object Oriented Programming**

Large programs are probably the most complicated entities ever created by humans. This leads to complexity that can cause errors which are difficult to locate and remove. Object Oriented Programming(OOP) eliminates many flaws associated with the conventional programming approach and offers a new and powerful way to cope up with the complexity. Its goal is to make clearer, more reliable and good quality software. OOP promises to make applications that can stay around for a longer time without problem and can easily be maintained and extended. There are many advantages of OOP approach which are explained as follows:

- Object Oriented Programming(OOP) lays more emphasis on data rather than functions and does not allow data of an object to be accessed freely in the system. This concept is supported by data hiding which provides greater system security and avoids unintended data corruption.
- OOP provides a clear modular structure for programs which makes it good for defining Abstract Data Type (ADT) using Data Abstraction. Data Abstraction provides benefits both to the user as well as programmer of the class. The user benefits as it interacts with the class using well defined interfaces(function names) without knowing the complex internal implementation details of the class. The programmer takes advantage by modifying the internal logic of the function without changing the interface.
- OOP is well suited for modelling real world problems as close as possible to the user's perspective. The concept of objects helps in trapping the human's thoughts in programming. Object communicates with each through well-defined interfaces using the concept of message passing. It provides proper integration of the system that helps the programmer to solve complex problems very easily. It also maximize re-usability, reduce data dependency and minimize debugging time.
- Object oriented systems are easily extensible as new classes easily be added with little or no modification in the existing system.
- OOPs provide the facility to co-exist the multiple objects without interference.