

Ane's Student Edition

Trouble Free

C++



Hari Mohan Pandey



Ane Books  
Pvt. Ltd.

# Trouble Free C++

**H.M. Pandey**

*Mukesh Patel School of Technology Management and  
Engineering, Shirpur Campus,  
SVKM's NMIMS University, Mumbai.*



**Ane Books Pvt. Ltd.**

New Delhi ♦ Chennai ♦ Mumbai

Bengaluru ♦ Kolkata ♦ Thiruvananthapuram ♦ Lucknow

### **Trouble Free C++**

H.M. Pandey

© Ane Books Pvt. Ltd., 2015

*Published by*

#### **Ane Books Pvt. Ltd.**

4821, Parwana Bhawan, 1st Floor, 24 Ansari Road,

Darya Ganj, **New Delhi** - 110 002, India

Tel.: +91(011) 23276843-44, Fax: +91(011) 23276863

e-mail: kapoor@anebooks.com, Website: [www.anebooks.com](http://www.anebooks.com)

#### *Branches*

- Avantika Niwas, 1st Floor, 19 Doraiswamy Road, T. Nagar,  
**Chennai** - 600 017, Tel.: +91(044) 28141554, 28141209  
e-mail: anebooks\_tn@airtelmail.in
- Plot No. 59, Sector-1, Shirwane, Nerul, **Navi Mumbai** - 400 706,  
Tel.: +91(022) 27720842, 27720851  
e-mail: anebooksmum@mtnl.net.in
- 38/1, 1st Floor, Model House, First Street, Opp. Shamanna Park,  
Basavannagudi, **Bengaluru** - 560 004, Tel.: +91(080) 41681432, 26620045  
e-mail: anebang@airtelmail.in
- Flat No. 16A, 220 Vivekananda Road, Maniktalla,  
**Kolkata** - 700 006, Tel.: +91(033) 23547119, 23523639  
e-mail: anebooks\_vsn@bsnl.in
- # 6, TC 25/2710, Kohinoor Flats, Lukes Lane, Ambujavilasam Road,  
**Thiruvananthapuram** - 01, Kerala, Tel.: +91(0471) 4068777, 4068333  
e-mail: anebookstvm@airtelmail.in

#### *Representative Office*

- C-26, Sector-A, Mahanagar, **Lucknow** - 226 006  
Mobile - +91 93352 29971

ISBN (13): 978-93-8015-662-0

---

All rights reserved. No part of this book may be reproduced in any form including photocopying, microfilms, photoprints, storage in any retrieval system, transmission in any permanent or temporary form, without the prior written consent of the publisher.

---

Printed at : Gopalji Enterprises, Delhi.

# **Trouble Free C++**



## Preface

This book is a comprehensive, hands-on guide to C++ programming- but one that does not assume you have programmed before. (People familiar with earlier programming or another structured programming language will, of course, have an easier time and can move through the early chapters quickly.)

Soon, you will write sophisticated programs that take full advantages of C++ exciting and powerful object-oriented nature. You will start at the beginning and when you have finished this book, you will have moved far along the road to C++ mastery. I have tried hard to cover at the least the fundamentals of every technique that a professional C++ will need to master.

I have also tried hard to stress the new ways of thinking needed to master C++ programming, so even experts in more traditional programming languages can benefit from this book. I have taken this approach because trying to force C++ into the framework of older programming languages is ultimately self-defeating you can not take advantage of its power if you continue to think within an older paradigm.

To make this book even more useful, there are extensive discussions of important topics left out of most other introductory books. There are whole chapter on objects, including non-trivial examples of building your own objects with C++. When I teach you process of inheritance at that time I have introduced you to develop C++ code on behalf of inherited diagrams. There is a chapter for input-output manipulators which shows you number of functions used in C++ for input/output control. In the same chapter I teach you about manipulator and ways to develop your own manipulator. There is a whole chapter for miscellaneous new features of C++. There is even a whole chapter on exception handling. There are methods of keeping file information through the chapter file handling. There are a chapter for Standard Template Library and for the String Class. There is a whole chapter for showing the hidden secrets of C++. The book also includes lots of examples with step by step explanation. The book also includes an extensive discussion of sorting and searching techniques and lots of tips and tricks. In sum, unlike many of the introductory books out there, I not only want to introduce you to a topic, but I go into it in enough depth that you can actually use the techniques for writing practical programs.

Now a confession: My original goal was to make this book a “**one-stop resource**”, but, realistically, C++ has gotten far too big and far too powerful for any one book to do this. Nonetheless, if you finish this book, I truly believe that you will be in a position to begin writing

commercial-quality C++ programs! True mastery will take longer: I have tried to give suggestions that can take you to next level.

## ➔ HOW THIS BOOK IS ORGANISED?

The subject-matter of this book is divided into **18 chapters**. Each chapter has been written and developed with immense simplified programs (**except chapter-1, which is foundation chapter for various programming methodology**) which will clear the core concepts of the C++ language.

The book “**TROUBLE FREE C++**” has been written specially for those students who are tyro in the field of programming. Inside the book you will find numerous programs instead of just code snippet to illustrate even the basic concept.

The book assumes no previous exposure to the C++ programming language. It also contains some good programming examples which might be useful for experienced programmers. All the programming examples given in the book have been tested on **VC++ compiler, Turbo C++ 3.0 and Turbo C++ 4.5 compilers** under **windows and DOS**.

Each chapter contains a number of **examples** to explain the theoretical as well as practical concepts. **Every chapter is followed by questions to test the student performance and receptivity.**

**Here are short descriptions of the chapters:**

**Chapter 1** covers the topics of basic introduction of programming methodology and introduction of OOP like **structured programming** by which one can understand the elementary elements (**sequence structure, Loop or iteration and Decision structure**) of any programming language. The chapter also explains the basic approaches (**Bottom-up and Top-down**) which compiler applied for the compilation of programs. This chapter explains the basic concepts of object-oriented programming too. This chapter gives idea to programmer to categorise any programming language into **object-oriented or object based language**.

**Chapter 2** In this chapter, I have explained the historical development of C++ language. The chapter also gives introductory idea of tokens, variables, data types and basic structure of C++ program. In the same chapter, I have explained the method of **compiling and executing** the C++ program on **Turbo C++3.0, Turbo C++ 4.5 and VC++**.

**Chapter 3** This chapter introduces programmers about the behavior of operators used in C++. In this chapter, I have explained the most of the common features applied in C and C++ both, because as we say C++ is super set of C then **operator and expression** used in C must be implemented with C++ too.

**Chapter 4** covers the **operators used only with C++ not with C**. In this chapter, I have covered the operators like **scope resolution operator, reference variables, bool data type**. This chapter gives idea of **dynamic memory allocation** and operators **new and delete** for dynamic memory allocation in C++.

**Chapter 5** gives the idea of **declaring function (prototyping)**, function of **main ()** function, introduction of **recursion**. It also gives the meaning of **call by reference** and **call by address** and difference between **call by reference** and **call by address**. Here I have explained the functionality of **inline function** and **function overloading** too.

**Chapter 6** gives the introduction of **class and objects** used in C++. Here I have put the comparison of **structure and class**, way of **accessing private data** and gives idea to **passing and returning objects**. In this chapter I have given some very crucial elements of C++ like **array of objects**, **friend function**, **Static class members** and **constant member function**. All these concepts play very important role in software development.

**Chapter 7** covers the behaviors of **constructor**. Here I show the role of different types of constructor like **default constructor**, **constructor with parameters**, **copy constructor**. This chapter also gives the ideas of dynamic constructor and destructor.

**Chapter 8** gives ideas to programmer to **overload different types of operators** used in C++ like, **binary operators**, **assignment operators**, **unary operators**. Overloading with the help of **friend function** and rules of overloading any operator and way for **type conversion** too.

**Chapter 9** In this chapter we will deal the concept of **inheritance** different types of inheritance *i.e.* **single level**, **multilevel**, **multiple**, **hierarchical** and **hybrid**. Here I have also defined the different visibility modifier with respect to inheritance. Application of **constructor and destructor** in inheritance and concept of **containership** is also defined in the same chapter.

**Chapter 10** gives the way of implementing the concepts like **pointers to objects**, **this pointer**, and way of **binding**, what is **virtual function** and how to work with **virtual function**, rules for **virtual function**? This chapter also gives the comparison of **virtual function**, **pure virtual function** the fundamental concepts of **object slicing** and **virtual destructor**.

**Chapter 11** explains the concepts of C++ **stream classes** and **formatted** and **unformatted** input and output operation applied in C++. In the same chapter I have also given the concept of **manipulator**.

**Chapter 12** gives the idea of how to handle **file** in C++ programming language? This chapter introduces programmer about the fundamental concepts of **file streams**, **way of opening and closing file**, **different modes of opening** a text file in C++. This chapter also gives the approaches to **check end of any file**, **Random access in file**. In the same chapter I have put the introductory ideas of **command line argument** and way of working with **binary mode** and **error handling mechanism** with file handling.

**Chapter 13** In this chapter, I have put miscellaneous new features of C++ like **explicit keyword**, **mutable keyword**, **constant object**, **namespace**, **typed operator**, etc.

**Chapter 14** covers the introductory idea of **template programming** *i.e.* **function template** and **class template**.



**Chapter 15** This chapter deals with basic concepts of **exception handling mechanism**, **how to handle exception with the help of class?** How to **re-throw** an exception, how to **catch** all exceptions.

**Chapter 16** gives the introduction of **standard template library**. Gives the behavior of **container classes**, introduces approaches for working with **vector** and working with **list**.

**Chapter 17** covers the topics like declaring and using **strings**. Introduces the principle by which one can differentiate **C++ strings with C- style strings**. Introduces operators for **string objects**, **methods for string classes**, **arrays of string** and **iterating string** etc.

**Chapter 18** gives the fundamental ideas of very basic elements used in C++. This chapter also clarifies some of the critical questions generated in students/faculty/professionals mind.

**Appendix 1** gives the explanations of some critical questions and answer.

**Appendix 2** presents some language-technical elements.

**Appendix 3** discusses the technical questions which are generally asked in technical interviews.

All efforts have been made to keep errors to barest minimum. However, there is possibility that some errors might have crept in inadvertently. I would, therefore, be grateful if such oversights are pointed out by the readers. Suggestions for improvement of the text from readers are welcome.

—Author

# Contents

<b>1. Introduction to Oops</b>	<b>1</b>
1.1 Structured Programming	1
1.2 Procedural Programming	3
1.3 Programming Methodology	4
1.4 Object Oriented Programming	5
1.5 Basic Concepts of Oops	6
1.6 Characteristics of Oops	9
1.7 Advantages of Oops	10
1.8 Object Oriented Programming Languages	10
1.9 Object-Based Languages	10
<b>2. Beginning with C++</b>	<b>11</b>
2.1 C++ Overview	11
2.2 C++ Character Set	12
2.3 C++ Tokens	12
2.4 Variables	16
2.5 Counting Tokens	18
2.6 Data Types	18
2.7 Qualifiers	19
2.8 Range of Data Types	20
2.9 Your First C++ Program	20
2.10 Structure of C++ Program	24
2.11 Style of Writing C++ Program	26
2.12 Programming Examples	27
<b>3. Introduction of C Features of C++</b>	<b>39</b>
3.1 Introduction	39
3.2 Operators and Expressions	39
3.3 Declaring Constants	65
3.4 Type Conversion	69
3.5 Decision Making: An Introduction	70

3.6	Unconditional Branching using Goto	84
3.7	Introduction of Looping	86
<b>4.</b>	<b>Introduction of Operators and References in C++</b>	<b>109</b>
4.1	Introduction	109
4.2	Scope Resolution Operator	110
4.3	Reference Variables	115
4.4	The Bool Data Type	120
4.5	The Operator New And Delete	122
4.6	Malloc Vs New	133
4.7	Pointer Member Operators	133
<b>5.</b>	<b>Introduction of Function in C++</b>	<b>143</b>
5.1	Introduction	143
5.2	Function Declaration /Prototyping	144
5.3	The Main Function in C++	148
5.4	Recursion	148
5.5	Call by Reference	151
5.6	Call by Reference Vs Call by Address	159
5.7	Return by Reference	159
5.8	Inline Function	163
5.9	Function Overloading	169
5.10	Function With Default Arguments	177
<b>6.</b>	<b>Introduction of Class and Objects</b>	<b>187</b>
6.1	Working with Class	187
6.2	Programming Examples (Part-1)	191
6.3	Structure in C++	209
6.4	Accessing Private Data	210
6.5	Programming Examples (Part-2)	214
6.6	Passing and Returning Object	235
6.7	Array of Object	243
6.8	Friend Function	249
6.9	Static Class Members	261
6.10	Constant Member Function	269
<b>7.</b>	<b>Constructor and Destructor</b>	<b>275</b>
7.1	Introduction	275
7.2	Constructor with Parameters	277
7.3	Implicit and Explicit Call to Constructor	279

---

7.4	Copy Constructor	303
7.5	Dynamic Initialization of Objects	308
7.6	Dynamic Constructor	313
7.7	Destructor	325
<b>8.</b>	<b>Introduction of Operator Overloading</b>	<b>333</b>
8.1	Introduction	333
8.2	Operator Overloading with Binary Operator	335
8.3	Overloading Assignment (=) Operator	345
8.4	Overloading Unary Operators	347
8.5	Overloading using Friend Function	356
8.6	Rules of Operator Overloading	368
8.7	Type Conversion	368
<b>9.</b>	<b>Introduction of Inheritance</b>	<b>391</b>
9.1	Introduction	391
9.2	Types of Inheritance	391
9.3	Public, Private and Protected Inheritance	396
9.4	Multiple Inheritance	422
9.5	Hierarchical Inheritance	429
9.6	Virtual Base Class	433
9.7	Constructor and Destructor in Inheritance	440
9.8	Containership	450
<b>10.</b>	<b>Pointers to Objects and Virtual Functions</b>	<b>457</b>
10.1	Pointer to Objects	457
10.2	The This Pointer	464
10.3	What is Binding in C++?	469
10.4	Virtual Functions	470
10.5	Working of a Virtual Function	475
10.6	Rules for Virtual Function	484
10.7	Pure Virtual Function and Abstract Class	484
10.8	Object Slicing	497
10.9	Some Facts about Virtual Function	500
10.10	Virtual Destructor	504
<b>11.</b>	<b>Introduction of Input-Output and Manipulators</b>	<b>511</b>
11.1	Introduction	511
11.2	C++ Stream Classes	511
11.3	Unformatted Input/Output	513

11.4	Formatted Input/ Output Operations	524
11.5	Manipulators	545
<b>12.</b>	<b>Introduction of File Handling in C++</b>	<b>563</b>
12.1	Introduction	563
12.2	File Streams	564
12.3	Opening and Closing a File	564
12.4	File Opening Modes	570
12.6	Checking End of File	575
12.7	Random Access in File	581
12.9	Command Line Arguments	588
12.10	Working with Binary Mode	593
12.11	Error Handling	604
<b>13.</b>	<b>Miscellaneous New Features</b>	<b>611</b>
13.1	Introduction	611
13.2	The Explicit Keyword	611
13.3	The Mutable Keyword	613
13.4	Constant Object	615
13.5	Namespace	616
13.6	The Typoid Operator	627
13.7	The Dynamic_Cast Operator	633
13.8	Static_Cast Operator	635
13.9	The Const_Cast Operator	637
13.10	Smart Pointer	640
<b>14.</b>	<b>Introduction of Template Programming</b>	<b>643</b>
14.1	Introduction	643
14.2	Function Template	644
14.3	Class Template	658
<b>15.</b>	<b>Exception Handling in C++</b>	<b>677</b>
15.1	Introduction	677
15.2	Basics of Exception Handling	677
15.3	Exception Handling Mechanism	679
15.4	Programming Examples	680
15.5	Exception Handling with Class	686
15.7	Catching All Exceptions	695
15.8	Specifying Exception for a Function	698

<b>16. Standard Template Library</b>	<b>703</b>
16.1 Introduction	703
16.2 Stl Overview	704
16.3 Container Classes	707
16.4 Working with Vector	708
16.5 Working with List	716
16.6 Deque	729
16.7 Map	732
16.8 Algorithms	738
16.9 Function Object	747
<b>17. The String Class</b>	<b>757</b>
17.1 Introduction	757
17.2 Declaring and Using String	757
17.3 Programming Example	759
17.4 How String is Different from C-Style Strings?	763
17.5 Operators for String Objects	763
17.6 Methods of String Class	767
17.7 Array of String	777
17.8 Iterating Strings	780
<b>18. Hidden Truth of C++</b>	<b>783</b>
18.1 Copy Constructor	783
18.2 Default Constructor	783
18.3 Destructor and Pointer	784
18.4 Constructor Member Function	785
18.5 Memory Allocation	785
18.6 Class Constructor Notation	786
18.7 Assigning New Name to cin	786
18.8 Assigning New Name to cout	786
18.9 Operator that cannot be Overloaded	787
18.10 Static Constructor	787
18.11 Bad_Alloc Exception	787
18.12 Calling of Copy Constructor	787
18.13 Operator Function as Virtual	788
18.14 Assignment Operator Function	788
18.15 Function that can not be Inherited	788
18.16 Features Of Templates	788

18.17	The Mutable Qualifier	788
18.18	The Dominance Rule	789
18.19	Inline Function	789
18.20	Name Mangling	789
18.21	One Definition Rule	789
18.22	Static Function	789
18.23	Memory Allocation for Namespace	790
18.24	Arrays and References	790
18.25	Pointer Members in Class	790
18.26	External and Internal Linkage	791
18.27	This Pointer as Hidden Argument	791
18.28	Private Class	791
18.29	Object Creation and Constructor	791
18.30	Destructor and Inheritance	792
18.31	Reference is a Constant Pointer	792
18.32	Vtable and Objects	792
18.33	Static Variables	793
18.34	Usage of Virtual Destructor	793
18.35	Reference for Reference	794
18.36	Pointer to Reference	794
18.37	Overloaded Form of << Operator	795
18.38	Modifier and Mutators	797
18.39	Global Objects	797
18.40	Put and Get Functions	798
18.41	Anonymous Union	799
18.42	Comma Operator with Scope	799
18.43	Static Object	800
18.44	Overriding Main Function	801
18.45	Overloading Main Function as Non-Member Function	802
18.46	Overloading Main Function as Member Function	802
18.47	Calling Constructor and Destructor Explicitly	803
18.48	Accessor	805
18.49	Object Cloning	805
18.50	Aggregation and Containment	805
18.51	SBI for an Object	805
18.52	Active and Passive Objects	805

18.53	Reflexive Association	805
18.54	Refined Name Mangling	806
18.55	Mixin Class	807
18.56	Overloading << as Virtual Member Function	807
18.57	Initializing Data Member in Constant Functions	808
18.58	Defining Constant Member Function Outside the Class	809
18.59	Pure Virtual Function can have Body	810
18.60	Size of an Empty Class	810
18.61	Calling Virtual Functions from Constructor	812
18.62	Calling Virtual Function Explicitly	813
18.63	Memory Alignment	813
18.64	The Pragma Directive	814
18.65	The Pragma Pack Directive	815
18.66	Translation Unit	816
18.67	Deleting This Pointer	816
18.68	C-Style Functions in C++	817
18.69	Initialization Static Members	817
18.70	Template and Memory	817
18.71	Smart Pointer Operator	817
18.72	Anonymous Class	819
18.73	Calling Virtual Function within Constructor	820
18.74	Applying Delete on Statically Allocated Pointers	821
18.75	Preventing Copy Constructor	822
18.76	Reuse Object Code and Source Code	823
18.77	Vptr and Vtable	823
18.78	Pure Abstract Class	823
18.79	Creating Anonymous Object	823
18.80	Private Constructor and Destructor	825
18.81	Using Main as a Class Name	826
18.82	Accessing Private Members in Main	827
18.83	Function Declaration and Exe File	828
18.84	Return Value of Main	828
18.85	The Set_Terminator Function in Exception Handling	828
18.86	Type of Enumeration	830
18.87	Size of Byte	830
18.88	Rules About Bytes and Characters	830



18.89	Pod	830
18.90	Argument to Operator Function	831
18.91	Dreaded Diamond	831
18.92	Another Way For Creating Default Constructor	831
18.93	Efficiency of Initialization in Constructor Initialization list	832
18.94	Static Method as Named Constructors	832
18.95	Running C in C++	833
18.96	Address of Variable as Operands	834
18.97	No Initialization in the Class	834
18.98	Garbage or Memory Leak	834
18.99	Dangling Pointer	835
18.100	This Pointer Internals	835
18.101	Returning Value from Constructor	835
18.102	Destructor cannot be Overloaded	836
18.103	Destructors for the Member Objects	836
18.104	Virtual Friend Function Idiom	838
18.105	The Friendship is not Inherited, Transitive and Reciprocal	840
18.106	Clear and Ignore	840
18.107	'In' Versus 'endl'	842
18.108	Default Constructor is not Granted	842
18.109	Static Data Objects as Composition	842
18.110	Call of Destructor within Catch	843
18.111	Initialization of Static Members	845
18.112	Default Constructors by C++	845
18.113	Constant Members and Function	845
18.114	Singleton Class	845
18.115	Changing the Visibility of Inherited Members	846
18.116	Early and Late Binding	847
18.117	Static with Virtual Function	847
18.118	Parts of Iostream	847
18.119	Deleting a Pointer Twice	848
18.120	Mixing Malloc and Delete	848
18.121	New is Better Than Malloc	848
18.122	Memory Allocation for References	848
18.123	Garbage Collection in C++	848
18.124	Finding Number of Objects to be Deleted	849