

Computer Programming with C++

This textbook provides in-depth explanation of C and C++ programming languages along with the fundamentals of object oriented programming paradigm. Essential concepts including functions, arrays, pointers and inheritance are explained in a coherent manner.

The book follows an example-driven approach, to facilitate easy comprehension of theoretical concepts. Common concepts of C programming language are also elaborated wherever necessary. The text provides detailed explanation on complex topics including Dynamic Memory Allocation, Object Slicing, VTABLEs, Up Casting and Down Casting.

The concepts are explained using line diagrams, notes, conversation themes and flow charts. The book offers useful features including error finding exercises, quiz questions and points to remember. Necessary comments to explain the logic used to implement particular functionality are provided for the ease of readers. Plenty of computer programs, review questions and useful case studies are interspersed throughout the text.

The book is intended for undergraduate and graduate students of engineering and computer science.

Kunal Pimarkhede is an adjunct faculty at the Vidyalkar Institute of Technology, Mumbai. As a software professional, he has development experience in technologies such as Adobe Flex, Informatica; database systems like Oracle, Sybase; Java based frameworks like Spring, Struts, etc. His areas of interest include Microprocessors, Automata Theory, Data Structures, Analysis of Algorithms, Artificial Intelligence and Computer Programming.

Cambridge University Press
978-1-316-50680-6 — Computer Programming with C++
Kunal Pimparkhede
Frontmatter
[More Information](#)

Computer Programming with C++

Kunal Pimparkhede



Cambridge University Press
978-1-316-50680-6 — Computer Programming with C++
Kunal Pimparkhede
Frontmatter
[More Information](#)

CAMBRIDGE
UNIVERSITY PRESS

University Printing House, Cambridge CB2 8BS, United Kingdom
One Liberty Plaza, 20th Floor, New York, NY 10006, USA
477 Williamstown Road, Port Melbourne, vic 3207, Australia
4843/24, 2nd Floor, Ansari Road, Daryaganj, Delhi – 110002, India
79 Anson Road, #06–04/06, Singapore 079906

Cambridge University Press is part of the University of Cambridge.

It furthers the University's mission by disseminating knowledge in the pursuit of education, learning and research at the highest international levels of excellence.

www.cambridge.org

Information on this title: www.cambridge.org/9781316506806

© Cambridge University Press 2017

This publication is in copyright. Subject to statutory exception and to the provisions of relevant collective licensing agreements, no reproduction of any part may take place without the written permission of Cambridge University Press.

First published 2017

Printed in India

A catalogue record for this publication is available from the British Library

Library of Congress Cataloging-in-Publication Data

Names: Pimparkhede, Kunal, author.

Title: Computer programming with C++ / Kunal Pimparkhede.

Description: New York : Cambridge University Press, 2016. | Includes index.

Identifiers: LCCN 2015051227 | ISBN 9781316506806 (paperback)

Subjects: LCSH: C++ (Computer program language) | C (Computer program language) | Object-oriented programming (Computer science) | Computer programming. | BISAC: COMPUTERS / Programming Languages / General.

Classification: LCC QA76.73.C153 P469 2016 | DDC 005.13/3--dc23 LC record available at <http://lcn.loc.gov/2015051227>

ISBN 978-1-31-650680-6 Paperback

Additional resources for this publication at www.cambridge.org/9781316506806

Cambridge University Press has no responsibility for the persistence or accuracy of URLs for external or third-party internet websites referred to in this publication, and does not guarantee that any content on such websites is, or will remain, accurate or appropriate.

Contents

<i>Preface</i>	<i>xi</i>
<i>Acknowledgements</i>	<i>xiii</i>

PART-I Structured Programming

1. Introduction	3
1.1 Overview	3
1.2 Computer System Architecture	8
1.3 C/C++ Development Environment	13
1.4 Evolution of Programming Languages	17
2. Fundamentals	23
2.1 Overview	23
2.2 The First C/C++ Program	23
2.3 Writing Comments	27
2.4 Constants or Literals	28
2.5 Variables and Data Types	31
2.6 <code>printf()</code> Statement (Supported by both C and C++)	38
2.7 C++ Style of Printing the Value on Computer Screen	49
2.8 <code>endl</code> Modifier (Supported by C++ not by C)	51
2.9 Accepting User Input Using <code>scanf()</code> Function (Supported by both C and C++)	52
2.10 <code>cin</code> Object in C++ (Only in C++ not in C)	58
2.11 Manipulator <code>setw</code>	63
2.12 Defining Constants using <code>#define</code> —A Pre-processor Directive	65
2.13 Character Specific Input/Output	66
3. Operators and Type Casting	71
3.1 Overview	71
3.2 Arithmetic Operators	77
3.3 Relational Operators	82

3.4	Shorthand Operators	84
3.5	Bitwise Operators in C/C++	85
3.6	Increment/Decrement Operators	89
3.7	Order of Operations Evaluated by the <code>printf()</code> Statement	94
3.8	Implicit Type Casting/System Casting	97
3.9	Explicit Type Casting	97
3.10	<code>sizeof</code> Operator in C/C++	101
3.11	Scope Resolution Operator(<code>::</code>) Only in C++ not in C	102
4.	Decision Making Control Statements	110
4.1	Overview	110
4.2	<code>if else</code> Statement	110
4.3	Logical Operators	123
4.4	<code>else if</code> Ladder	125
4.5	<code>switch</code> Statement	132
4.6	Ternary Operator/Conditional Operator	135
4.7	<code>goto</code> Statement	139
5.	Iterative Control Statements: Loops	146
5.1	Introduction	146
5.2	<code>while</code> Loop in C/C++	149
5.3	<code>for</code> Loop in C/C++	164
5.4	<code>do..while</code> Loop	180
5.5	<code>break</code> and <code>continue</code> Statements	183
5.6	Infinite Loops	190
5.7	Comma Operator with <code>for</code> Loop	192
5.8	Creating Variables Local to Loops (Possible in C++ but not in C)	192
5.9	Empty Loops	193
6.	Arrays	216
6.1	Overview	216
6.2	Creating an Array	218
6.3	Array of Characters	238
6.4	2D Arrays	252
6.5	2D Array of Characters	261
6.6	String-Specific Input and Output Operations: <code>gets()</code> / <code>puts()</code>	264
7.	Functions	299
7.1	Overview	299
7.2	Creating Functions	303
7.3	Local Variables of the Function	309
7.4	Functions with Arguments	313
7.5	Functions with Return Values	318

7.6	Passing Array as an Argument to the Function	325
7.7	Recursion	329
7.8	Activation frames: How Function Calls and Returns are Internally Handled in C/C++	337
7.9	Storage Classes in C/C++	342
7.10	Inline Functions in C/C++	345
7.11	Function with Default Arguments (Only in C++ not in C)	347
7.12	Command Line Arguments	349
7.13	Some Built-in Functions	353
8.	Pointers	371
8.1	Overview	371
8.2	Creating Pointers	377
8.3	Data Type of Pointers	380
8.4	Types of Function Calls	383
8.5	Arithmetic Operations with Pointers	393
8.6	Accessing Array Elements using a Pointer	400
8.7	Initialization of an Array: Revisited	407
8.8	Self-addressability of Character Variables	413
8.9	Array of Pointers	415
8.10	Pointer to a Pointer	418
8.11	Pointers and 2D Arrays	420
8.12	void Pointers	426
8.13	Pointer to a Function	427
8.14	Reference Variables (Only Available in C++ not in C)	429
8.15	Lvalue and Rvalue	433
9.	Structures and Unions	445
9.1	Overview	445
9.2	Creating Structures	446
9.3	Array of Structure Objects	457
9.4	Nesting of Structures	469
9.5	Structures and Pointers	472
9.6	Accessing Array of Objects using a Pointer	476
9.7	Passing Object as an Argument to a Function	479
9.8	Difference between Structure and Union	481
10.	Dynamic Memory Allocation in C++	497
10.1	Overview	497
10.2	Dynamic Memory Management in C++	499
10.3	Linked List	509
10.4	delete Keyword in C++	527

PART-II Object Oriented Programming

11. Classes and Objects	533
11.1 Overview	533
11.2 Creating Classes	537
11.3 Creating Objects of a Class	538
11.4 Access Specifiers in C++	542
11.5 Data Hiding and Encapsulation	545
11.6 Employee Management System: An Example	548
11.7 Account Management System: An Example	554
11.8 Calculating Slope of the Line: An Example	559
11.9 Addition of Complex Numbers: An Example	563
11.10 Addition of Points in Cartesian Coordinate System: An Example	567
11.11 Array of Objects	569
11.12 Employee Management System: Revisited	571
11.13 <code>friend</code> Functions	573
11.14 Addition of <code>Point</code> Objects using <code>friend</code> Function: An Example	574
11.15 Pointer to Objects	580
11.16 Binding of Pointers with Individual Members of the Class	584
11.17 <code>this</code> Pointer	589
11.18 Resolving Ambiguity using <code>this</code> Pointer	592
11.19 Cloning Objects using <code>this</code> Pointer: An Example	594
11.20 Dynamic Memory Allocation of Objects	596
11.21 Linked List to Maintain Data about Employees	600
11.22 Composition and Aggregation between Classes	609
11.23 Converting the Relationship to Aggregation	615
11.24 Defining the Member Functions Outside Class using Scope Resolution Operator	620
11.25 Function Overloading and Compile Time Binding	622
11.26 Local Classes	625
11.27 Nested Classes	627
12. Constructors and Destructors	652
12.1 Overview	652
12.2 Creating Constructors	654
12.3 Constructor Overloading	659
12.4 Program to Perform Addition of <code>Point</code> Objects using Constructors: An Example	665
12.5 Constructor with Default Arguments	668
12.6 Cloning Objects using Constructor/Copy Constructor	669
12.7 Allocating Dynamic Memory Inside Constructor	674
12.8 Destructors in C++	680
12.9 Static Members and Static Member Functions	684

13. Operator Overloading	699
13.1 Overview	699
13.2 Overloading Operators	700
13.3 Overloading One's Complement ~ and Minus - Operators: An Example	703
13.4 Overloading Binary Operators Plus + and Minus -	707
13.5 Overloading Shorthand Operators: An Example	713
13.6 Overloading Relational Operators: An Example	716
13.7 Overloading Increment/Decrement Operators: An Example	720
13.8 Function Object: Overloading Function Call Operator ()	726
13.9 Overloading Subscript Operator []	728
13.10 Overloading Assignment Operator =	731
13.11 Overloading Type Cast Operator	735
13.12 Conversion of One User-defined Type to Another	738
13.13 Creating Global Operator Functions	741
13.14 Overloading Insertion and Extraction Operator for Student Objects: An Example	745
13.15 Overloading Operators new and delete	749
13.16 Overloading operator ->	754
14. Inheritance	770
14.1 Overview	770
14.2 Creating a Parent–Child Relationship between Classes	773
14.3 Access Specifiers in C++: Revisited	776
14.4 Types of Inheritance	788
14.5 IS-A and HAS-A Relationship: An Example	793
14.6 Multi-level Inheritance: Calculator	801
14.7 Resolving Ambiguity in Multiple Inheritance	808
14.8 Virtual Base Class	813
14.9 Function Overriding	821
14.10 Pointers and Inheritance	824
14.11 Overriding a Function with Different Return Type	831
14.12 Virtual Functions and Runtime Polymorphism	833
14.13 Virtual Tables	843
14.14 Pure Virtual Functions and Abstract Classes	851
14.15 <code>static_cast</code> and <code>dynamic_cast</code>	854
14.16 Constructors and Inheritance	858
14.17 Working of Constructors with Multiple Inheritance	863
14.18 Destructors and Inheritance	864
14.19 Virtual Destructors	867
15. Input and Output Streams in C++	881
15.1 Overview	881
15.2 Types of I/O Streams	882

x ◆ Contents

15.3	Console Input and Output in C++	883
15.4	Formatted v/s Unformatted I/O Operations	884
15.5	Formatting the Output using Member Functions of Class <code>ios</code>	885
15.6	Formatted I/O using <code>ios</code> Flags	889
15.7	Formatted I/O using Manipulators	893
15.8	Creating your Own Manipulator	894
15.9	Passing Arguments to the Custom Manipulator	895
15.10	Character by Character Unformatted I/O Operations	897
15.11	Line by Line Unformatted I/O Operations	900
15.12	File I/O Operations	902
15.13	Performing Operations on File	904
15.14	Closing the File	914
15.15	File Pointer Manipulation Functions	915
16.	Templates in C++	928
16.1	Overview	928
16.2	Function Templates	928
16.3	Class Templates	933
16.4	Standard Template Library: One of the Applications of Class Templates	937
16.5	Implementation of Stack using Linked List: An Example	959
16.6	Queue using List	964
17.	Exception Handling in C++	969
17.1	Overview	969
17.2	Exception Handler in C++	970
17.3	<code>throw</code> Keyword	971
17.4	Examples	971
17.5	Order of writing Catch Blocks	981
17.6	Catching and Throwing User-defined Objects	982
17.7	Program Specifying Throw List	986
17.8	C++ Built-in Exception Classes	988
	<i>ASCII Values</i>	994
	<i>List of Keywords in C++</i>	998
	<i>Software Development Life Cycle</i>	999
	<i>Bibliography</i>	1002

Preface

A computer program is a set of instructions which is followed by a machine to generate the required output. The language in which a computer program is written is called a computer programming language. Several computer programming languages are in use in the IT industry today, for developing diverse software applications.

The study of C and C++ is considered an important step towards mastering computer programming fundamentals. Hence, C and C++ are included in the syllabus of any computer science course.

This textbook provides in-depth explanations of C and C++ programming languages along with the fundamentals of the object oriented programming paradigm.

About the Book

This book will be of use to anyone who is a beginner and aspires to learn the fundamentals of computer programming using C and C++. It has been primarily written for students of academic courses which include the study of C, C++ and object oriented programming paradigm. Simple and lucid language has been used to facilitate easy comprehension of complex topics.

Salient Features

- Example-driven approach illustrates application of theoretical concepts
- Theme of a conversation interspersed in the text, elucidate essential themes of the subject
- Each program includes necessary comments to explain the logic used to implement a particular functionality
- Several line diagrams and flow charts facilitate easy comprehension of theoretical concepts
- Student-friendly pedagogical features include:
 - ✓ Error Finding Exercise
 - ✓ Solved Problems

xii ◆ Preface

- ✓ Objective Questions
- ✓ Review Questions

Chapter Organization

This book comprises 17 chapters. **Chapter 1** gives an overview of computer organization and architecture. It also explains the C/C++ development environment. **Chapters 2 to 5** discuss the basic features of C/C++ including data types, variables and different control statements which are supported by the language. **Chapter 6** describes the creation of multivalued data types (also referred to as collection types) using arrays in C/C++. **Chapter 7** explains modular programming using functions. **Chapter 8** elucidates the fundamentals of memory management using pointers in C/C++. **Chapter 9** discusses the creation of composite data types using structures and unions in C/C++. Chapter 10 explains the principles of memory management and Dynamic memory allocations in C++ style. **Chapters 11 to 17** provide in-depth coverage of object oriented features supported by C++.

NOTES

Chapters 1 to 9 cover features which are common to C as well as C++. Hence programs written in these chapters will work with C as well as C++ compilers unless specified otherwise. Whereas Chapters 11 to 17 cover object oriented features which are supported only by C++ and not by C. Chapter 10 explains dynamic memory allocation in C++ style. Hence programs written from Chapter 10 to 17 will work with C++ compilers only.

Does this book also explain the underlying systems which are involved in the execution of a computer program ? **?**



Chapter 1 gives an understanding of computer organization, operating system and other system programs which make up the underlying platform required to execute any C/C++ program. Chapter 1 also gives an overview of many areas which are relevant for understanding computer programming fundamentals using C/C++. The specific features of individual topics have been explained in detail in the later chapters of this book.

I have put my best efforts to make this book as illustrative and interactive as possible. Any suggestions to further improve this book are always welcome. You can write to me at kunalp84@rediffmail.com.

Acknowledgements

Contribution and support of many people in my life has made this book possible. I am indebted to Vishwas Deshpande, Chairman, Vidyalkar Institute of Technology for giving me a wonderful platform to showcase my learnings. I am grateful to Professor N. H. Dubey for his constructive feedback, which added significant value to this project. I owe special thanks to my dear dearest daughter Swara for allowing me to work long hours while I was writing this book.

I am thankful to Professors V. S. Padmakumar, Sanjeev Dwivedi, Sachin Bhojewar, Pankaj Vanwari from Vidyalkar Institute of Technology; Professors Kalpana Sagvekar, Sunil Surve and Brijmohan Daga from Fr. Conceicao Rodrigues College of Engineering, Mumbai; Professor Asawari Dudwadkar, Vivekanand Institute of Technology, Mumbai; Professors Prasad Kulkarni, Vinayak Shinde, Leena Thakur, Surbhi Crasto and Ajit Parab from Babasaheb Gawde Institute of Technology, Mumbai; Professor Yogesh Prabhu, Dr A. K. Pathak, Professor Sameer Velankar and Professor Yogesh Rajadhyaksha for their excellent support at different stages which helped me to present my learnings so well in the form of this book.

I am thankful to Ruhi Bajaj, Thadomal Shahani Engineering College, Mumbai and Shweta Loonkar, D. J. Sanghavi College of Engineering, Mumbai for their valuable suggestions which helped mould this script to meet the needs of a wide range of audience.

I sincerely acknowledge the contribution of all my teachers, who have played a vital role in developing my understanding of the subject and broadening my perspectives. They have strongly influenced me in building a positive attitude towards creative learning.

I would also like to thank every member of the team at Cambridge University Press, including Rachna Sehgal, who supported me throughout the publishing process to actuate timely release of this book.

I express heartfelt gratitude to my family for their moral support and patience. I am thankful to Pradeep Pimparkhede, Shalaka Pimparkhede, Dipti Pimparkhede, Swara Pimparkhede, Prabhavati Pimparkhede, Nivedita Bakre, Rajgopal Pai and Jayanti Pai for their patience while I was writing this script.