



C++

Programming Step-by-Step

Asadullah Shah



IIUM PRESS

INTERNATIONAL ISLAMIC UNIVERSITY MALAYSIA

C++ PROGRAMMING: STEP BY STEP

Editors

Asadullah Shah



IIUM Press

Published by:
IIUM Press
International Islamic University Malaysia

First Edition, 2011
©IIUM Press, IIUM

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without any prior written permission of the publisher.

Perpustakaan Negara Malaysia

Cataloguing-in-Publication Data

Bibliography p.
Includes Index
ISBN

ISBN: 978-967-418-090-4

Member of Majlis Penerbitan Ilmiah Malaysia · MAPIM
(Malaysian Scholarly Publishing Council)

Printed by :
IIUM PRINTING SDN. BHD.
No. 1, Jalan Industri Batu Caves 1/3
Taman Perindustrian Batu Caves
Batu Caves Centre Point
68100 Batu Caves
Selangor Darul Ehsan

CONTENTS

DEDICATION	iii
PREFACE	viii
ACKNOWLEDGEMENT	ix
1. INTRODUCTION	
<i>Asadullah Shah and Assadullah Shaikh</i>	1
2. ARITHMETIC EXPRESSIONS AND DATA TYPES IN C++	
<i>Asadullah Shah and Assadullah Shaikh</i>	5
3. SENDING THE OUTPUT TO A PRINT FILE	
<i>Asadullah Shah and Assadullah Shaikh</i>	11
4. DECISION MAKING: IF-ELSE STATEMENTS AND RELATIONAL OPERATORS	
<i>Asadullah Shah and Assadullah Shaikh</i>	17
5. LOGICAL OPERATORS AND SWITCH STATEMENTS	
<i>Asadullah Shah and Assadullah Shaikh</i>	25
6. REVIEW, SUMMARY & BUILDING SKILL	
<i>Asadullah Shah and Khamran Khowaza</i>	33
7. ITERATIVE STRUCTURES	
<i>Asadullah Shah and Khamran Khowaza</i>	39

8. THE FOR LOOP	
<i>Asadullah Shah and Khamran Khowaza</i>	49
9. THE DO-WHILE LOOP	
<i>Asadullah Shah and Khamran Khowaza</i>	55
10. REVIEW OF VARIABLES, FORMATTING	
<i>Asadullah Shah and Khamran Khowaza</i>	59
11. REVIEW OF ITERATIVE STRUCTURES	
<i>Asadullah Shah and Sumbul Khowaza</i>	63
12. POST-TEST AND NESTED LOOPS	
<i>Asadullah Shah and Sumbul Khowaza</i>	73
13. FUNCTIONS	
<i>Asadullah Shah and Sumbul Khowaza</i>	83
14. CALL-BY-VALUE AND REFERENCE	
<i>Asadullah Shah and Sumbul Khowaza</i>	91
15. MORE ON FUNCTIONS	
<i>Asadullah Shah and Sumbul Khowaza</i>	99
16. STRUCTURES (STRUCT) AND FILES	
<i>Asadullah Shah and Muniba Shaikh</i>	111
17. ARRAYS	
<i>Asadullah Shah and Muniba Shaikh</i>	119
18. EXERCISE OF ARRAY	
<i>Asadullah Shah and Muniba Shaikh</i>	127

19. READ DATA FROM A FILE	
<i>Asadullah Shah and Muniba Shaikh</i>	137
20. OBJECT ORIENTED PROGRAMMING	
<i>Asadullah Shah and Muniba Shaikh</i>	143
21. SELECTION SORTING	
<i>Asadullah Shah and Syed Ifthar Ali</i>	153
22. BUBBLE SORT ALGORITHM	
<i>Asadullah Shah and Syed Ifthar Ali</i>	161
23. REVIEW OF ARRAYS	
<i>Asadullah Shah and Syed Ifthar Ali</i>	167
24. LINEAR SEARCHING	
<i>Asadullah Shah and Syed Ifthar Ali</i>	179
25. BINARY SEARCH	
<i>Asadullah Shah and Syed Ifthar Ali</i>	189
26. VECTOR CLASS	
<i>Asadullah Shah and Ejaz Ahmed</i>	199
27. POINTERS	
<i>Asadullah Shah and Ejaz Ahmed</i>	203
28. FUNCTION POINTERS	
<i>Asadullah Shah and Ejaz Ahmed</i>	213
29. POLYMORPHISM AND VIRTUAL FUNCTIONS	
<i>Asadullah Shah and Ejaz Ahmed</i>	219

30. C++ REFERENCES	
<i>Asadullah Shah and Ejaz Ahmed</i>	223
31. CONST CORRECTNESS	
<i>Asadullah Shah and Osama Mahfooz</i>	229
32. MORE ON CONST KEYWORDS	
<i>Asadullah Shah and Osama Mahfooz</i>	235
33. GOTO STATEMENT	
<i>Asadullah Shah and Osama Mahfooz</i>	241
34. HANDLING ERRORS IN C++	
<i>Asadullah Shah and Osama Mahfooz</i>	249
35. STATIC: THE MULTIPURPOSE KEYWORD	
<i>Asadullah Shah and Osama Mahfooz</i>	253

28. FUNCTION POINTERS

Asadullah Shah and Ejaz Ahmed

Department of Computer Science, Faculty of Information and
Communication Technology, International Islamic University Malaysia,
Malaysia

Abstract

A function pointer is a variable that stores the address of a function that can later be called through that function pointer. The function pointers are very useful in C++ when a function might be used for a particular behaviors, for instance, sorting a list, the sorting can be ascending as well as descending order, instead of writing out a bunch of code, all you need to do is call the function. But sometimes you would like to choose different behaviors at different times in essentially the same piece of code. In such a situation function pointer plays a role and can be used to adopt the any behavior.

28.1 Functions as Arguments to Other Functions

Assuming that one might want to allow the function's caller to choose the order in which the data is sorted. One way to do this is to provide a flag as an argument to the function any one of these ascending or descending sorting will be based upon the flag conditions and execution of code takes place.

28.2 Callback Functions

Another use for function pointers is setting up "listener" or "callback" functions that are invoked when a particular event happens. The function is called, and this notifies your code that something of interest has taken place.