



# 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

---

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

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

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

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

# 30. C++ REFERENCES

---

Asadullah Shah and Ejaz Ahmed

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

## Abstract

A C++ *reference* is an alternative name for an object. All operations applied to a reference act on the object to which the reference refers. The address of a reference is the address of the alternative object.

A reference type is defined by placing the reference modifier `&` after the type specifier. All references must be initialized except function parameters when they are defined. Once defined, a reference cannot be reassigned because it is an alternative to its target. If a function needs to modify the actual value of an argument or needs to return more than one value, the argument must be passed by reference. Passing arguments by reference can be done using either references or pointers. Unlike other languages, C++ does not force to use pointers if someone wants to pass arguments by reference.

## 30.1 Introduction

C++ references allow creating another name for a variable what this means is that when you declare a reference and assign it a variable, it will allow to treat the reference exactly as though it were the original variable for the purpose of accessing and modifying the value of the original variable—even if the reference is located within a different scope. This is quite different from how C++ normally works, where you have arguments to a function copied into new variables.