C++ Programming: Step by Step

Editors

Asadullah Shah

IIUM Press
Published by:
IIUM Press
International Islamic University Malaysia

© 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
Asadullah Shah and Assadullah Shaikh ................................. 1

2. ARITHMETIC EXPRESSIONS AND DATA TYPES IN C++
Asadullah Shah and Assadullah Shaikh .................................. 5

3. SENDING THE OUTPUT TO A PRINT FILE
Asadullah Shah and Assadullah Shaikh .................................. 11

4. DECISION MAKING: IF-ELSE STATEMENTS AND
RELATIONAL OPERATORS
Asadullah Shah and Assadullah Shaikh .................................. 17

5. LOGICAL OPERATORS AND SWITCH STATEMENTS
Asadullah Shah and Assadullah Shaikh .................................. 25

6. REVIEW, SUMMARY & BUILDING SKILL
Asadullah Shah and Khamran Khowaza .................................. 33

7. ITERATIVE STRUCTURES
Asadullah Shah and Khamran Khowaza .................................. 39
8. THE FOR LOOP
Asadullah Shah and Khamran Khowaza ........................................ 49

9. THE DO-WHILE LOOP
Asadullah Shah and Khamran Khowaza ........................................ 55

10. REVIEW OF VARIABLES, FORMATTING
Asadullah Shah and Khamran Khowaza ........................................ 59

11. REVIEW OF ITERATIVE STRUCTURES
Asadullah Shah and Sumbul Khowaza .......................................... 63

12. POST-TEST AND NESTED LOOPS
Asadullah Shah and Sumbul Khowaza .......................................... 73

13. FUNCTIONS
Asadullah Shah and Sumbul Khowaza .......................................... 83

14. CALL-BY-VALUE AND REFERENCE
Asadullah Shah and Sumbul Khowaza .......................................... 91

15. MORE ON FUNCTIONS
Asadullah Shah and Sumbul Khowaza .......................................... 99

16. STRUCTURES (STRUCT) AND FILES
Asadullah Shah and Muniba Shaikh ........................................... 111

17. ARRAYS
Asadullah Shah and Muniba Shaikh ........................................... 119

18. EXERCISE OF ARRAY
Asadullah Shah and Muniba Shaikh ........................................... 127
19. READ DATA FROM A FILE  
*Asadullah Shah and Muniba Shaikh* ......................................................... 137

20. OBJECT ORIENTED PROGRAMMING  
*Asadullah Shah and Muniba Shaikh* .............................................................. 143

21. SELECTION SORTING  
*Asadullah Shah and Syed Ifihar Ali* ............................................................... 153

22. BUBBLE SORT ALGORITHM  
*Asadullah Shah and Syed Ifihar Ali* ............................................................... 161

23. REVIEW OF ARRAYS  
*Asadullah Shah and Syed Ifihar Ali* ............................................................... 167

24. LINEAR SEARCHING  
*Asadullah Shah and Syed Ifihar Ali* ............................................................... 179

25. BINARY SEARCH  
*Asadullah Shah and Syed Ifihar Ali* ............................................................... 189

26. VECTOR CLASS  
*Asadullah Shah and Ejaz Ahmed* ................................................................. 199

27. POINTERS  
*Asadullah Shah and Ejaz Ahmed* ................................................................. 203

28. FUNCTION POINTERS  
*Asadullah Shah and Ejaz Ahmed* ................................................................. 213

29. POLYMORPHISM AND VIRTUAL FUNCTIONS  
*Asadullah Shah and Ejaz Ahmed* ................................................................. 219
30. C++ REFERENCES
Asadullah Shah and Ejaz Ahmed ................................................. 223

31. CONST CORRECTNESS
Asadullah Shah and Osama Mahfooz ........................................... 229

32. MORE ON CONST KEYWORDS
Asadullah Shah and Osama Mahfooz ........................................... 235

33. GOTO STATEMENT
Asadullah Shah and Osama Mahfooz ........................................... 241

34. HANDLING ERRORS IN C++
Asadullah Shah and Osama Mahfooz ........................................... 249

35. STATIC: THE MULTIPURPOSE KEYWORD
Asadullah Shah and Osama Mahfooz ........................................... 253
35. Static: The Multipurpose Keyword

Asadullah Shah and Osama Mahfooz
Department of Computer Science, Faculty of Information and Communication Technology, International Islamic University Malaysia, Malaysia

Abstract
The keywords are the fixed set of reserved terms that are used by programming languages to be used for developing programs. These keywords cannot be used as variables or data items. The key words are same as symbol set of a set of characters used by natural languages.

35.1 Introduction
Most C++ keywords allow you to do one thing. For example "int" is used to declare an integer, or that a function returns an integer or expects an integer as an argument. To allocate memory "new" is used and "delete" is used to free memory. The keyword static has multiple uses. The keyword static can be used in three major contexts: inside a function, inside a class definition, and in front of a global variable inside a file making up a multiple file program.

35.2 Use of Static keyword
The simplest of all uses of static is its use inside a function. It simply means that once the variable has been initialized, it remains in memory until the end of the program. For instance, you can use a static variable to record the number of times a function has been called simply by including the lines static int count = 0; and count++; inside the function.