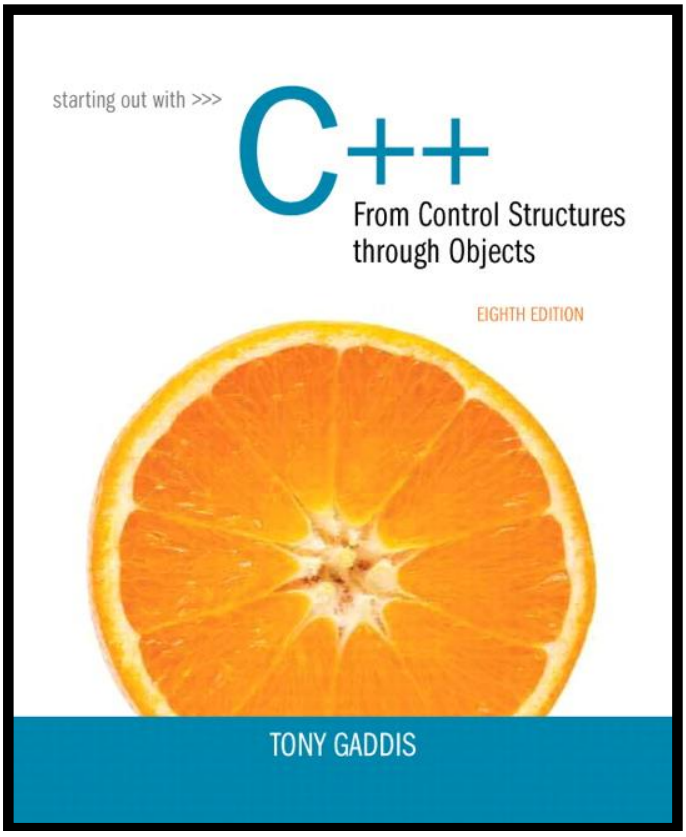
LAB MANUAL

to Accompany



Dean DeFino
Salisbury University

Michael Bardzell
Salisbury University

PEARSON
Addison
Wesley

Boston San Francisco New York
London Toronto Sydney Tokyo Singapore Madrid
Mexico City Munich Paris Cape Town Hong Kong Montreal

Editorial Director	*Marcia Horton*
Acquisitions Editor	*Matt Goldstein*
Program Manager	*Kayla Smith-Tarbox*
Editorial Assistant	*Kelsey Loanes*
Director of Marketing	*Christy Lesko*
Marketing Assistant	*Jon Bryant*

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and Addison-Wesley was aware of a trademark claim, the designations have been printed in initial caps or all caps.

Copyright © 2015 Pearson Education, Inc.

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 the prior written permission of the publisher. Printed in the United States of America. For information on obtaining permission for use of material in this work, please submit a written request to Pearson Education, Inc., Rights and Contracts Department, 75 Arlington Street, Suite 300, Boston, MA 02116, fax your request to 617-848-7047, or e-mail at http://www.pearsoned.com/legal/permissions.htm.

ISBN 0-133-77864-9
1 2 3 4 5 6 7 8 9 10—BB—11 10 09 08

Contents

Lab Manual Introduction vii

LESSON SET 1 ***Introduction to Programming and the Translation Process 1***

Pre-lab Reading Assignment 2
Computer Systems 2
Introduction to Programming 2
Translation Process 3
Integrated Environments 6

Pre-lab Writing Assignment 6
Fill-in-the-Blank Questions 6
Learn the Environment That You Are Working In 7

Lesson 1A 7
LAB 1.1 Opening, Compiling and Running Your First Program 7
LAB 1.2 Compiling a Program with a Syntax Error 7
LAB 1.3 Running a Program with a Run Time Error 8

Lesson 1B 9
LAB 1.4 Working with Logic Errors 9
LAB 1.5 Writing Your First Program (Optional) 11

LESSON SET 2 ***Introduction to the C++ Programming Language 13***

Pre-lab Reading Assignment 14
The C++ Programming Language 14
Memory 15
Variables and Constants 16
Identifiers in C++ 16
Data Types 16
Integer Data Type 16
Floating Point Data Type 17
Character Data Type 17
Boolean Data Type 17
Assignment Operator 17
Fundamental Instructions 17
Arithmetic Operators 19

Pre-lab Writing Assignment 19
Fill-in-the-Blank Questions 19

Lesson 2A 20
LAB 2.1 Working with the `cout` Statement 20
LAB 2.2 Working with Constants, Variables and Arithmetic Operators 21

Lesson 2B 22
LAB 2.3 Rectangle Area and Perimeter 22
LAB 2.4 Working with Characters and Strings 22

LESSON SET 3 ***Expressions, Input, Output and Data Type Conversions 25***

Pre-lab Reading Assignment 26
Review of the `cout` Statement 26
Input Instructions 26
Strings 27
Summary of storing and inputting strings 28
Formatted Output 28
Expressions 29
Precedence Rules of Arithmetic Operations 29
Converting Algebraic Expressions to C++ Expressions 30
Data Type Conversions 30
Files 31

Pre-lab Writing Assignment 32
Fill-in-the-Blank Questions 32

Lesson 3A 33
LAB 3.1 Working with the `cin` Statement 33
LAB 3.2 Formatting Output 35
LAB 3.3 Arithmetic Operations and Math Functions 36

Lesson 3B 37
LAB 3.4 Working with Type Casting 37
LAB 3.5 Reading and Writing to a File 38
LAB 3.6 Student Generated Code Assignments 39

LESSON SET 4 ***Conditional Statements 41***

Pre-lab Reading Assignment 42
Relational Operators 42
The `if` Statement 42
The `if/else` Statement 43
The `if/else if` Statement 43
The Trailing `else` 44
Nested `if` Statements 44
Logical Operators 45
The `switch` Statement 46
Character & string comparisons 47

Pre-lab Writing Assignment 48
Fill-in-the-Blank Questions 48

Lesson 4A 48
LAB 4.1 Working with Relational Operators and the `if` Statement 48
LAB 4.2 `if/else` and Nested `if` Statements 49
LAB 4.3 Logical Operators 50

Lesson 4B 51
LAB 4.4 The `switch` Statement 51
LAB 4.5 Student Generated Code Assignments 52

LESSON SET 5 *Loops and Files 55*

Pre-lab Reading Assignment 56
Increment and Decrement Operator 56
The `while` Loop 56
Counters 58
Sentinel Values 59
Data Validation 60
The `do-while` Loop 60
The `for` Loop 61
Nested Loops 63
Files 64

Pre-lab Writing Assignment 65
Fill-in-the-Blank Questions 65

Lesson 5A 65
LAB 5.1 Working with the `while` Loop 65
LAB 5.2 Working with the `do-while` Loop 67

Lesson 5B 69
LAB 5.3 Working with the `for` Loop 69
LAB 5.4 Nested Loops 70
LAB 5.5 Reading and Writing to a File 72
LAB 5.5 Student Generated Code Assignments 73

LESSON SET 6.1 *Introduction to Void Functions (Procedures) 75*

Pre-lab Reading Assignment 76
Modules 76
Pass by Value 78
Pass by Reference 81

Pre-lab Writing Assignment 83
Fill-in-the-Blank Questions 83

Lesson 6.1A 84
LAB 6.1 Functions with No Parameters 84
LAB 6.2 Introduction to Pass by Value 84

Lesson 6.1B 86
LAB 6.3 Introduction to Pass by Reference 86
LAB 6.4 Student Generated Code Assignments 89

LESSON SET 6.2 *Functions that Return a Value 91*

Pre-lab Reading Assignment 92
Scope 92
Scope Rules 93
Static Local Variables 94
Default Arguments 94
Functions that Return a Value 96
Overloading Functions 99
Stubs and Drivers 99

Pre-lab Writing Assignment 101
Fill-in-the-Blank Questions 101

Lesson 6.2A 101
LAB 6.5 Scope of Variables 101
LAB 6.6 Parameters and Local Variables 104

Lesson 6.2B 106
LAB 6.7 Value Returning and Overloading Functions 106
LAB 6.8 Student Generated Code Assignments 110

LESSON SET 7 *Arrays 113*

Pre-lab Reading Assignment 114
One-Dimensional Arrays 114
Array Initialization 115
Array Processing 115
Arrays as Arguments 116
Two-Dimensional Arrays 121
Multi-Dimensional Arrays 122
Arrays of Strings 122

Pre-lab Writing Assignment 122
Fill-in-the-Blank Questions 122

Lesson 7A 123
LAB 7.1 Working with One-Dimensional Arrays 123
LAB 7.2 Strings as Arrays of Characters 126

Lesson 7B 129
LAB 7.3 Working with Two-Dimensional Arrays 129
LAB 7.4 Student Generated Code Assignments 134

LESSON SET 8 ***Searching and Sorting Arrays 137***

Pre-lab Reading Assignment 138
Search Algorithms 138
Linear Search 138
The Binary Search 140
Sorting Algorithms 142
The Bubble Sort 143
The Selection Sort 145

Pre-lab Writing Assignment 148
Fill-in-the-Blank Questions 148

Lesson 8 149
LAB 8.1 Working with the Linear Search 149
LAB 8.2 Working with the Binary Search 150
LAB 8.3 Working with Sorts 152
LAB 8.4 Student Generated Code Assignments 156

LESSON SET 9 ***Pointers 157***

Pre-lab Reading Assignment 158
Pointer Variables 158
Using the & Symbol 158
Using the * Symbol 159
Using * and & Together 160
Arrays and Pointers 161
Dynamic Variables 162
Review of * and & 166

Pre-lab Writing Assignment 167
Fill-in-the-Blank Questions 167

Lesson 9A 167
LAB 9.1 Introduction to Pointer Variables 167
LAB 9.2 Dynamic Memory 168

Lesson 9B 170
LAB 9.3 Dynamic Arrays 170
LAB 9.4 Student Generated Code Assignments 171

LESSON SET 10 ***Characters and Strings 175***

Pre-lab Reading Assignment 176
Character Functions 176
Character Case Conversion 177
String Constants 178
Storing Strings in Arrays 179
Library Functions for Strings 179
The `get` and `ignore` functions 181
Pointers and Strings 184

Pre-lab Writing Assignment 186
Fill-in-the-Blank Questions 186

Lesson 10 187
LAB 10.1 Character Testing and String Validation 187
LAB 10.2 Case Conversion 190
LAB 10.3 Using `getline()` 192
LAB 10.4 String Functions—`strcat` 193
LAB 10.5 Student Generated Code Assignments 193

LESSON SET 11 ***Structures and Abstract Data Types 195***

Pre-lab Reading Assignment 196
Access to Structure Members 197
Arrays of Structures 200
Initializing Structures 201
Hierarchical (Nested) Structures 202
Structures and Functions 204

Pre-lab Writing Assignment 205
Fill-in-the-Blank Questions 205

LESSON 11 A 205
LAB 11.1 Working with Basic Structures 205
LAB 11.2 Initializing Structures 206
LAB 11.3 Arrays of Structures 208

LESSON 11 B 209
LAB 11.4 Nested Structures 209
LAB 11.5 Student Generated Code Assignments 211

LESSON SET 12 ***Advanced File Operations 213***

Pre-lab Reading Assignment 214
Review of Text Files 214
Opening Files 214
Reading from a File 215
Output Files 218
Files Used for Both Input and Output 219
Closing a File 220
Passing Files as Parameters to Functions 220
Review of Character Input 221
Binary Files 224
Files and Records 226
Random Access Files 228

Pre-lab Writing Assignment 231
Fill-in-the-Blank Questions 231

Lesson 12A 231
LAB 12.1 Introduction to Files (Optional) 231
LAB 12.2 Files as Parameters and Character Data 233

Lesson 12B 235
LAB 12.3 Binary Files and the `write` Function 235
LAB 12.4 Random Access Files 238
LAB 12.5 Student Generated Code Assignments 240

LESSON SET 13 ***Introduction to Classes 243***

Pre-lab Reading Assignment 244
Introduction to Object-Oriented Programming 244
Client and Implementation Files 246
Types of Objects 247
Implementations of Classes in C++ 247
Creation and Use of Objects 247
Implementation of Member Functions 248
Complete Program 251
Inline Member Functions 254
Introduction to Constructors 255
Constructor Definitions 256
Invoking a Constructor 256
Destructors 256
Arrays of Objects 258

Pre-lab Writing Assignment 260
Fill-in-the-Blank-Questions 260

Lesson 13A 261
LAB 13.1 Squares as a Class 261
LAB 13.2 The Circles as Class 263

Lesson 13.B 265
LAB 13.3 Arrays as Data Members of Classes 265
LAB 13.4 Arrays of Objects 267
LAB 13.5 Student Generated Code Assignment 269

APPENDIX A ***Visual C++ Environment 271***

APPENDIX B ***UNIX 273***

Index 277

Lab Manual Introduction

To the Student...

A closed laboratory in computer programming is a vital activity for helping you gain valuable programming skills. Programming cannot be learned by "spectators". In other words, you cannot become a skilled programmer simply by watching others do it. You must spend numerous hours working on programs yourself. A closed laboratory experience gives you the opportunity to edit, write, compile, build, and execute programs of varying length and complexity under the guidance of your instructor. You will be able to reinforce concepts learned in class with a "hands on" approach. Throughout the course, your programming skills should steadily progress by applying knowledge learned in class to the laboratory setting.

This lab manual is divided into chapters called "Lesson Sets". At the beginning of each lesson set you will see a Purpose section which outlines the goals and expected outcomes of the lesson. This is immediately followed by a Procedure section. The first two steps of this section ask you to complete the Pre-lab Reading and Pre-lab Writing Assignments as a prerequisite to attempting the labs. It is imperative that you do both assignments before coming to your lab session. The laboratory exercises assume you have read and understood the key points of the corresponding lesson. The Pre-lab Writing Assignment usually consists of 8 – 10 very simple fill in the blank questions. Once the Pre-lab Reading is complete, you should have no trouble completing these questions. Your instructor may choose to collect this assignment at the beginning of your lab session. Although each Pre-lab Reading Assignment gives a concise overview of key concepts from the corresponding chapter in the text, it is not a substitute for reading your text. The text develops ideas in much more detail and also covers certain topics that cannot be included in a closed lab due to time constraints. Hence, this lab manual should be used as a supplement, not a replacement, for the text.

Your instructor will tell you which lab assignments should be completed during the lab session and which should be completed outside of class for homework. Although a hard copy of all code used for the lab assignments is included at the end of each lesson set, the code is also included in electronic form on the Web at www.aw.com/cssupport, under author "Gaddis." You should use this code rather than re-typing it from scratch.

To the Instructor...

A closed laboratory in computer programming is a vital activity for helping students gain valuable programming skills under your guidance. Many different opinions concerning the content of such labs have been generated over the past few years, ranging from programming assignments to scheduled exercises using prepared materials. Although this manual emphasizes the latter approach and has pre-developed code for students to complete or edit, there are assignments

that ask each student to independently create small programs which may be assigned as lab activities or as post-lab homework. These student generated code assignments are not intended as a substitute for larger programming assignments. Rather, they are small programs designed to test students on the material given in the lessons. The length of the lab activities vary from fifty minutes to two hours, depending on the particular institution. For this reason, the manual is divided into "Lesson Sets", each consisting of two fifty to sixty minute lessons of lab work. A fifty minute lab session should be able to complete an individual lesson and a one and a half to two hour session should be enough time for an entire lesson set. These times refer to "average classes". It is of course impossible to set a time frame for each student in a given lab. It is natural that some advanced students may finish a little early, while others will need more than the suggested time frame. Each Lesson Set corresponds to a chapter from Starting Out with C++: From Control Structures through Objects, Eighth Edition, by Tony Gaddis. The one exception, however, is Chapter 6. This chapter deals with functions and the corresponding laboratory exercises are broken into two lesson sets.

The lab exercises in each lesson set are generally very simple to start and then increase in difficulty. Consequently, the student generated code assignments, which ask students to write complete programs, are given at the end of the second lesson. Most lesson sets contain three such assignments, so you have some flexibility as to how many of these programs are written during the laboratory session. A few lessons do have one somewhat sophisticated (to a beginning programmer) student generated code assignment. Other programming assignments may also be found at the end of each chapter of the text.

Each lesson set consists of the following:

Pre-lab Reading Assignment. This will prepare the students for material presented in the lab. This section gives a good, but brief, review of the corresponding chapter of the text. Examples and sample code are provided throughout this section, some of which are used in the subsequent labs. Students should thus be required to read this section before coming to lab.

Pre-lab Writing Assignment. These consist of short and easy questions on the reading material so that you may make sure students completed the pre-lab reading.

Two Lessons of Lab Assignments. These are done during the lab time, one lesson per hour (or fifty minute period).

Supplements: The following items are available at Addison-Wesley's Instructor Resource Center. Visit the Instructor Resource Center at www.aw.com/irc to register for access.

- Solutions to the lab exercises
- Teacher's Notes which consist of the following:

  Objectives for Students. These are similar to the Purpose section given at the beginning of each lesson set in the lab manual. However, the objectives listed are geared more for the lab work whereas the Purpose section in the manual refers to the Pre-lab Reading material as well. In some lessons they are the same.

  Assumptions. This section gives a brief list of what students should already know before attempting the corresponding lab assignment. It is generally assumed that the students have completed and understood the previous lessons (although some of the later lessons can be skipped) and that they have read and understood the Pre-lab Reading Assignment for the current lesson.

Pre-lab Writing Assignment Solutions. This section contains the answers to the Pre-lab Writing Assignment.

Lab Assignments. This section first lists the labs and then gives a more detailed description of each lab. Labs are broken into the lessons in which they are assigned.

Each instructor is encouraged to pick and choose labs based on the needs of their individual classes. The following is a suggested outline for a 14 week course that meets in a closed lab once a week for 50-60 minutes. This allows you to still cover one chapter a week for most weeks. As a general rule, a one hour lab session is enough time to complete section A of each lesson set. Assignments from section B, including the student generated code assignments, could be given as homework assignments.

Week 1	Lesson Set 1	Lab 1.1	Lab 1.2	Lab 1.3	Lab 1.4 (Optional Homework)
Week 2	Lesson Set 2	Lab 2.1	Lab 2.2		Lab 2.4 (Optional Homework)
Week 3	Lesson Set 3	Lab 3.1	Lab 3.2	Lab 3.3	Lab 3.4 (Optional Homework)
Week 4	Lesson Set 4	Lab 4.1	Lab 4.2	Lab 4.3	Lab 4.4 (Optional Homework)
Week 5	Lesson Set 5	Lab 5.1	Lab 5.2		Lab 5.5 (Optional Homework)
Week 6	Lesson Set 6.1	Lab 6.1	Lab 6.2		Lab 6.3 (Optional Homework)
Week 7	Lesson Set 6.2	Lab 6.5	Lab 6.6		Lab 6.7 (Optional Homework)
Week 8	Lesson Set 7	Lab 7.1	Lab 7.2		Lab 7.3 (Optional Homework)
Week 9	Lesson Set 8	Lab 8.1	Lab 8.2		Lab 8.3 (Optional Homework)
Week 10	Lesson Set 9	Lab 9.1	Lab 9.2		Lab 9.3 (Optional Homework)
Week 11	Lesson Set 10	Lab 10.1	Lab 10.2	Lab 10.3	Lab 10.4 (Optional Homework)
Week 12	Lesson Set 11	Lab 11.1	Lab 11.2	Lab 11.3	Lab 11.4 (Optional Homework)
Week 13	Lesson Set 12	Lab 12.2	Lab 12.3		Lab 12.4 (Optional Homework)
Week 14	Lesson Set 13	Lab 13.1	Lab 13.3	Lab 13.4	Lab 13.2 (Optional Homework)

For a one semester course that meets 2 hours a week in a closed lab, one lesson set per week will cover the manual in a fourteen week semester.