

# Instructor's Summary for *Murach's C++ Programming* (2<sup>nd</sup> Edition)

Welcome to the instructor's materials for *Murach's C++ Programming (2nd Edition)*. Their purpose is to help any college instructor or corporate trainer run an effective course based on the book. This summary introduces the materials we offer and helps you get started using them.

At the least, we recommend that you read the topics under *What's included in the instructor's materials* because they not only describe the components but also our underlying instructional philosophy. Then, you can find charts that summarize the components at a glance in the section on *How to get started with our materials*.

But first, we offer some thoughts about the structure of this book. We hope its logical progression and emphasis on professional skills will make it a great guide for you and your students.

<b>About the structure of the book .....</b>	<b>2</b>
Section 1: Essential skills for modern C++ .....	2
Section 2: More skills as you need them .....	2
Section 3: Object-oriented programming .....	2
Section 4: Data structures and algorithms .....	2
What to do with all this content .....	2
<b>What's included in the student download .....</b>	<b>3</b>
Book applications .....	3
Exercise starts .....	3
Exercise solutions .....	4
<b>What's included in the instructor's materials .....</b>	<b>4</b>
Book applications, exercise starts, and solutions .....	4
Objectives .....	4
Test banks .....	5
Projects and solutions .....	5
Case studies and solutions .....	5
PowerPoint slides .....	5
<b>How to get started with our materials .....</b>	<b>6</b>
How to use the zip file .....	6
The student download files that get installed.....	7
The instructor materials that get installed .....	7
<b>Any comments? .....</b>	<b>8</b>



**Mike Murach & Associates, Inc.**

1-800-221-5528 • (559) 440-9071 • Fax: (559) 440-0963

[murachbooks@murach.com](mailto:murachbooks@murach.com) • [www.MurachForInstructors.com](http://www.MurachForInstructors.com)

Copyright © 2022 Mike Murach & Assoc. Inc. All rights reserved.

## About the structure of the book

---

To present the C++ skills that your students need in a manageable progression, *Murach's C++ Programming (2<sup>nd</sup> Edition)* is divided into four sections.

### Section 1: Essential skills for modern C++

---

Section 1 presents an eight-chapter course in C++ programming that gets your students off to a great start. By the time they finish chapter 8, they'll be able to develop, test, debug, and deploy C++ procedural programs. In these chapters, your students will learn how to use strings and vectors, how to use files for persistent data storage, and how to code functions. By itself, this section is an excellent first course in programming.

### Section 2: More skills as you need them

---

The three chapters in this section build on the essentials in section 1 by presenting more skills that every C++ programmer should have. That includes how to work with structures, enumerations, built-in arrays, C strings, and exceptions.

The chapters in this section are written as independent modules, which means you can assign them in whatever sequence you prefer, and you don't have to assign all of them. In short, you can adapt the content to the time constraints and requirements of your course.

### Section 3: Object-oriented programming

---

The three chapters in this section present the most important concepts of object-oriented programming (OOP): encapsulation, object composition, inheritance, and polymorphism. These concepts are the same in all modern programming languages, so once your students master them, they'll be able to apply them in any other language they need to learn.

### Section 4: Data structures and algorithms

---

The five chapters in this section present skills that you can use to introduce your students to data structures and algorithms. The first two chapters show how to use the containers, iterators, and algorithms that are available from the Standard Template Library (STL). Then, the next three chapters present the skills you need to work with legacy code that doesn't use the STL or with embedded systems that don't support the STL. These skills include using pointers to work with memory at a low level and using templates to allow a class to support multiple data types.

## What to do with all this content

---

For most curricula, *Murach's C++ Programming (2<sup>nd</sup> Edition)* presents more concepts and skills than you can cover in a single course. But because the sections and chapters are clearly delineated by function, you have some choice in creating the course you want.

For example, when your students complete section 1, you'll have already taught an excellent first course in C++ programming. So if your students are completely new to programming, you can focus on this section, teaching it at a pace that won't leave anyone behind.

On the other hand, if your students are able to move more quickly through section 1, you have some options as to what to cover next, depending on your own interests and those of your students. For example, if OOP is an important part of your course, you can teach chapters 9 and 11 next and then go on to section 3. Or if you want your students to know about pointers, you can teach chapters 9-12 after section 1, and then skip ahead to chapter 17.

But beyond that, instructors often tell us that their students keep our books for reference on the job later on. So don't worry if you don't have the time to teach all the chapters. Your students will still get their money's worth out of our book!

## What's included in the student download

---

To help your students get the most from our book, our retail website at [www.murach.com](http://www.murach.com) lets them download (1) the book applications, (2) the starting code for the exercises in the book, and (3) *the solutions to the book exercises*. Appendixes A and B in the book show your students how to download and set up these materials on their own systems.

Chapter 1 in the book introduces your students to using Visual Studio on Windows or Xcode on macOS to develop C++ programs. So the downloadable code is stored in folders named *vs* and *xcode*. There's also a folder named *dist* that contains the files needed to deploy an application as described in chapter 8.

### Book applications

---

All of the programs in this book are included and stored in a folder named *book\_apps* within the *vs* and *xcode* folders. Once your students have set up these book applications on their own systems, they can run them to see how they work. They can review all of the code in any application when the book only presents the coding highlights. And they can copy and paste code from the book applications into their own C++ programs.

### Exercise starts

---

Each chapter in the book ends with exercises to help your students master the skills covered in the chapter. But unlike other exercises you've seen, these are designed to give your students the most practice in the least time. That's why your students will start most of the exercises from program files that contain some of the routine code that the exercise requires. That way, your students can focus on the new skills that they're learning. These exercise starts are stored starting in a folder named *ex\_starts* within the *vs* and *xcode* folders.

And just so you know, the chapter exercises also differ from the norm in that they don't focus on trivial busywork. Instead, they guide students through the process of building and enhancing a variety of programs that show how new skills are used together. Yes, the exercises go step-by-step, but your students will be practicing with the kind of programs they'll encounter in the real world, instead of dealing with single, isolated skills that provide no perspective. In fact, if your students can successfully do all of the exercises, they will be well on their way to a professional level of competence.

## Exercise solutions

---

To help students get over any learning obstacles when they're working on their own, the download also provides the solutions to the book exercises in a folder named *ex\_solutions* within the *vs* and *xcode* folders. That way, students can check the solutions to see how something is done whenever they're wasting time on what is likely to be a trivial coding mistake. We think that providing the solutions is the right approach didactically because it helps students learn faster and better.

We realize, however, that this means you can't use the book exercises to test your students. That's why the instructor's materials include a set of chapter-by-chapter projects as well as two extensive case studies that can be used for testing.

## What's included in the instructor's materials

---

The instructor's materials are designed to make it easier for you to teach a course based on the text, to ensure that your students gain the C++ skills they'll need on the job, and to evaluate their progress. So besides the materials in the student download, we provide instructional objectives, test banks, projects, case studies, and PowerPoint slides. A summary of these materials follows.

## Book applications, exercise starts, and solutions

---

These are the same materials that your students can download from our retail website. We've included them in the instructor's files so you can demonstrate and review the book applications and exercise solutions in class, without having to download them yourself.

## Objectives

---

We believe that instructional objectives should be the start of any educational methodology, so we provide a set of objectives for each chapter in the book. We created these objectives based on the principles presented by Robert F. Mager in his classic book, *Preparing Instructional Objectives*. As a result, our objectives describe the skills that your students should have when they complete a chapter, and you should be able to test whether they have those skills.

If you review the objectives, you'll see that the first objectives for each chapter are what we refer to as *applied objectives*. These ask the students to apply what they've learned as they develop C++ programs. These, of course, are the critical objectives of a programming course, and they are best tested by having the students develop the projects or case studies that we provide.

After the applied objectives for each chapter, you'll find what we refer to as *knowledge objectives*. These objectives define skills like identifying, describing, and explaining the required concepts, terms, and procedures. These objectives determine whether your students are able to talk intelligently about the topics that are presented. And these objectives can be tested by the test banks that we provide.

To help you get the most from the instructional objectives, we include them at the start of the PowerPoint slides for each chapter. As we see it, if you can convince your students that they only need to have the skills that are described by the objectives, their study becomes more focused and efficient.

## Test banks

---

To test comprehension, you can use the test banks that we've created; there's one for each chapter in the book. We developed these test banks in ExamView, and we provide them in multiple formats, including those that can be used in various LMSs like Blackboard, D2L Brightspace, and Canvas, as well as Rich Text Format for Microsoft Word. (If you need a format that we haven't provided, please let us know, and we'll do our best to deliver what you need.)

Each test bank provides multiple-choice questions that are designed to test the skills described by the objectives for that chapter, and each test question is designed to test the skill described by one objective. This keeps the promise to the students that they will only be expected to have the skills that are described by the objectives.

We use only multiple-choice test questions because not only are they the easiest to score but they also have the highest validity when it comes to assessing a student's knowledge and skills. In contrast, matching and true/false questions have low validity, so we don't use them.

## Projects and solutions

---

To give your students practice and to test whether they can develop their own C++ programs, the instructor's materials include over 50 projects, broken down by chapter. For each chapter, the projects range from simple to complex so you can assign the ones that are appropriate for your students. If your students can do the more difficult projects for each chapter, that's proof that they're developing the skills that are needed on the job.

The instructor's materials also include any starting files that are needed for the projects, as well as the project solutions. That way, you can demonstrate the projects in class. You can also show the code for the solutions, and you can compare it to the solutions that the students develop.

## Case studies and solutions

---

To provide a more extensive way to test the programming skills of your students, the instructor's materials also include two case studies. One case study has the students build a Shopping Cart program that lets the user add or remove items from a virtual shopping cart. The other has the students build a Blackjack game that lets the user play blackjack against a computer dealer. The instructor's materials also include the solutions for both case studies.

These case studies are made to be assigned on a section-by-section basis. That means that the students build one version of the case-study program for section 1 of the book, an enhanced version for section 2, an object-oriented version for section 3, and a version that uses a custom container and algorithm for section 4. To facilitate that, the instructor's materials provide section-by-section write-ups for the case studies, as well as the solutions for each section.

## PowerPoint slides

---

In our book, the figures (or illustrations) on the righthand pages present all of the critical information, including screenshots, diagrams, syntax, and code. Then, our PowerPoint slides are based on these figures. As a result, you don't have to worry about the slides introducing material that isn't explained fully in the book. Instead, they make it easy for you to review any skills that your students are having trouble with or to answer any

questions in class. And beyond the book information, the slides for each chapter start with the instructional objectives so you can go over them in class.

## How to get started with our materials

---

Once you have an instructor account at our instructor website ([www.murachforinstructors.com](http://www.murachforinstructors.com)), you can request the instructor's materials for our book and download them from your account page in a zip file. Then, you can install the materials on your computer as described below.

Once the installation is done, you can do a thorough review of the materials. In particular, you'll want to run some of the book applications, as well as some of the solutions for the book exercises, projects, and case studies, to see the level of competence that our book develops. You'll also want to click through some of the PowerPoint slides to see how they can help you review and reinforce the information that's presented in the book. To help you find what you're looking for, the entire file structure for the instructor's materials is shown on the next page.

### How to use the zip file

---

1. Download the .zip file for the instructor's materials from your Murach account page.
2. Create a folder named *murach* directly on your hard drive.
3. Unzip the zip file into the *murach* folder. This will create a file structure that starts with:

**murach/cpp**

## The student download files that get installed

murach\cpp\student_download\	Contents
dist\	The executable files for a console application that you can run as described in chapter 8.
vs\ book_apps ex_starts ex_solutions	The Visual Studio projects for the book applications, exercise starts, and exercise solutions.
xcode\ book_apps ex_starts ex_solutions	The Xcode projects for the book applications, exercise starts, and exercise solutions.

## The instructor materials that get installed

murach\cpp\instructors\	Contents
Objectives.docx Objectives.pdf	The instructional objectives for all chapters in both Word and PDF (the individual chapter objectives are repeated in the chapter slides).
student_projects\ Projects.docx Projects.pdf  vs\ project_starts product_solutions  xcode\ project_starts project_solutions	The specifications for the chapter-by-chapter student projects in both Word and PDF.  The project starts for Visual Studio. The project solutions for Visual Studio.  The project starts for Xcode. The project solutions for Xcode.
case_studies\ Case study – Shopping cart.docx Case study – Shopping cart.pdf Case study – Blackjack.docx Case study – Blackjack.pdf  vs\case_study_solutions xcode\case_study_solutions	The section-by-section specifications for the Shopping Cart case study in both Word and PDF.  The section-by-section specifications for the Blackjack case study in both Word and PDF.  The case study solutions for Visual Studio.  The case study solutions for Xcode.
slides\	One PowerPoint file for each chapter.
test_banks\	One test bank per chapter, organized by format: ExamView, RTF (Word), and Blackboard (which can be imported into Canvas and D2L Brightspace).

## Any comments?

---

If you have any comments about our book or its instructor's materials, we would be delighted to hear from you. If you discover any errors in our applications or solutions, we would appreciate hearing about them. And if you want to let us know that you're going to adopt our book for your course, that would make our day.

Just email us at the addresses below. But whether or not we hear from you, we want to thank you for your interest in our C++ book.

Joel Murach, Author  
[joel@murach.com](mailto:joel@murach.com)

Shawn Allen, Academic Representative  
[shawn@murach.com](mailto:shawn@murach.com)