Instructor's Summary for *Murach's Python Programming (2nd Edition)*

Welcome to the instructor's materials for *Murach's Python 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 you to these materials 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, the section entitled *How to get started with our materials* guides you in getting the materials on your system and gives you charts that summarize the components at a glance.

But first, some thoughts about the modular structure of this book that you should be aware of. This structure is important because it gives you instructional options that you just don't have with other books.

Mike Murach & Associates, Inc.
1-800-221-5528 • (559) 440-9071 • Fax: (559) 440-0963
murachbooks@murach.com • www.MurachForInstructors.com

About the structure of the book

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

Section 1: Essential concepts and skills

Section 1 presents an 8-chapter course in Python programming that gets your students off to a great start. By the time they finish chapter 4, they'll be able to develop, test, and debug Python programs that are built entirely of modules and functions. They'll also be able to use pseudocode to plan their control structures and programs, as well as hierarchy charts to plan the functions of their programs. Then, in chapter 5, they'll learn improved techniques for testing and debugging their programs.

In chapters 6, 7, and 8, your students will learn how to use lists and tuples, how to use files for persistent data storage, and how to handle exceptions. This completes a section that by itself is an excellent first course in programming, and this will put your students far ahead of what they can accomplish with competing books.

Section 2: Other concepts and skills

The first four chapters in this section present other skills that every Python programmer should have. That includes how to work with numbers, strings, dates and times, and dictionaries. Then, the last chapter in this section presents algorithms and recursion.

Since all of the chapters in this section are written as independent modules, you can assign them in whatever sequence you prefer, and you don't have to assign all of them. This makes it easy for you to adapt this book to the time constraints and requirements of your course.

Section 3: Object-oriented programming

The three chapters in this section present object-oriented programming: the OOP essentials, when and how to use inheritance, and how to design an OO program. These concepts are the same in all modern programming languages, so once your students master them…which is easier to do in Python using our approach…they'll be able to apply them in any other language they need to learn.

Depending on the emphasis of your course, you can skip to this section right after you complete section 1. Then, you can return later to the chapters in section 2.

Section 4: GUI and database programming

The two chapters in this section are solid introductions to database programming and GUI programming. This brings all of the skills of sections 1 and 3 into the context of real-world programming. For instance, the last database program in chapter 17 is a 3-tier, shopping cart program, complete with modules, classes, and functions.

About the modularity of the book

For most courses, *Murach's Python Programming* will present more concepts and skills than there is time for. So please keep in mind that our book has a modular design, which means you don't have to teach the chapters in sequence, and you can choose which ones to cover in your course.

With that in mind, we offer these thoughts on how you can use our book:

- When your students complete section 1, you've already taught an excellent first course in programming.
- You can teach the chapters in section 2 in whatever sequence you prefer.
- You can skip to section 3 after you finish section 1.
- You can teach the chapters in section 4 after your students complete sections 1 and 3.

So if your students are completely new to programming, you can focus on section 1, teaching it at a pace that won't leave anyone behind. But if they're able to move more quickly through that section, you have a number of options as to what to cover next, depending on your own interests and those of your students.

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 lets them download (1) the book applications, (2) the SQLite database that's used in chapter 17, (3) the starting code for the exercises in the book, and (4) *the solutions to the exercises*. Appendixes A and B in the book show your students how to download and set up these materials on their own Windows or macOS systems.

Book applications

All of the programs in this book are included in what we refer to as the *book applications,* and they're stored in a folder named *book_apps*. The chapter 17 applications require a SQLite database, and the files for that are stored in a folder named *_db*.

Once your students have set up the 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 Python 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 top-level folder named *exercises.*

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 tie all their newly learned skills together. In other words, instead of dealing with one single, isolated skill at a time, your students will be practicing with the kind of programs they'll encounter in the real world. In fact, if your students can successfully do all of the book 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 exercises in a top-level folder named *solutions*. 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 makes it difficult for an instructor to use the book exercises to test their students. That's why the instructor's materials include over 60 projects and 3 case studies that can be used for testing purposes, along with their solutions. The next section describes these problem sets in more detail.

What's included in the instructor's materials

The instructor's materials for this book are designed to save you time in preparing and running an effective course based on the text so that your students gain the Python skills they'll need on the job. 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 materials 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 Python 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) and QTI (the standard format for assessment content).

Each test bank provides 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.

In our test banks, we use only multiple-choice test questions because they not only are the easiest to score but also have the highest validity. As a result, we find that the students who get the best scores are those with the best 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 programs, the instructor's materials include over 60 projects. 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, which is likely to be written in a way that is more professional than 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 3 case studies. To make these case studies as useful as possible, we've constructed them so they can be assigned in two different ways.

One way they can be assigned is on a section-by-section basis. That means that the students build one version of the program for the case study for section 1 of the book, an enhanced version for section 2, an object-oriented version for section 3, and a database and GUI version 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.

The other way to assign the case studies is on a chapter-by-chapter basis. In other words, each case study can be started once the students complete chapter 2 and then enhanced after each chapter. To facilitate that, the instructor's materials provide chapter-by-chapter write-ups as well as the solutions for each chapter.

PowerPoint slides

The PowerPoint slides present all of the critical information from the figures of the book. That includes all of the screenshots, diagrams, tables, and code that you may want to review in class. As a result, these slides make it easy for you to review any of the skills that your students have difficulty with. Beyond the book information, the slides for each chapter start with the instructional objectives so you can review them in class.

How to get started with our materials

You can request the instructor's materials for our book at our instructor website (www.murachforinstructors.com) and download them from your account page there. The download is available as a zip file. Then, you can install the materials on your computer as detailed below.

Once the installation is done, you can thoroughly review of all of the materials. In particular, you'll want to run some of the book applications, exercise solutions, project solutions, and case study solutions 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 of 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/python`

The student folders and files that get installed

python/Student download	Contents
book_apps	The applications that each chapter presents.
_db	The files needed to install or restore the SQLite database that's used in chapter 17.
exercises	The starting points for the exercises at the end of each chapter.
solutions	The solutions for the exercises at the end of each chapter.

The instructor folders and files that get installed

python/Instructors	Contents
Objectives.pdf	The instructional objectives for all chapters.
Projects/Projects.pdf	The project specifications for all chapters.
Projects/project_starts	The starting files for the projects that require starting files.
Projects/project_solutions	The solutions to the projects.
Case studies/ Baseball case study by section.pdf Baseball case study by chapter.pdf Blackjack case study by section.pdf Blackjack case study by chapter.pdf Data importer case study by section.pdf Data importer case study by chapter.pdf	The specifications for 3 case studies in both section-by-section and chapter-by-chapter formats.
Case studies/case_study_solutions	The solutions to the case studies, both by section and by chapter.
Slides	One PowerPoint file for each chapter.
Test banks	One test bank per chapter, organized by format: ExamView, RTF (Word), Blackboard (which can be imported into Canvas and D2L Brightspace), Respondus, and IMS QTI (the standard test bank format).

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 e-mail us at the addresses below. But whether or not we hear from you, we want to thank you for your interest in our Python book.

Joel Murach, Author
joelmurach@gmail.com

Judy Taylor, Educational Liaison
judy@murach.com or murachbooks@murach.com