OBJECT-ORIENTED PROGRAMMING IN C++ 4TH EDITION Read Free

- Author: Robert Lafore
- ISBN: 9780768658842

Thanks to Serge Palladino, Quality Assurance Tester, whose attention to detail has guaranteed that this is a quality textbook. Thank you to my husband Geoff for whom I am more grateful every day. Finally, this book is dedicated to Rich and Sue in honor of their 30th wedding anniversary. Your instructor will provide the data files to you. You also can obtain the files electronically from the Course Technology Web site by connecting to www. Each chapter in this book has its own set of data files, stored in a separate folder. For example, the files for Chapter 3 are stored in the Chapter03 folder.

You can use a computer in your school lab or your own computer to complete the labs and exercises in this book. Appendix A contains instructions on getting started with Microsoft Visual Studio Appendix B contains instructions on getting started with some other compilers and suggests minor modifications you might have to make to your programs to get them to work correctly using different compilers. If your book came with a copy of Microsoft Visual Studio Express or Professional , then you may install that on your computer and use it to complete the material.

Data files. You will not be able to complete the steps and exercises in this book using your own computer unless you have the data files. You can get the data files from your instructor, or you can obtain the data files electronically from the Course Technology Web site by connecting to www. They may also be obtained electronically through the Course Technology Web site at www.

You can view the Help file using a text editor such as Notepad. Once the files are copied, you should instruct your users how to copy the files to their own computers or workstations. After learning or reviewing what it means to program, you will examine the characteristics of procedural programs and consider a few examples. Then you will compare procedural and object-oriented programs and learn the additional features object-orientation provides. In the rest of the chapter, you will consider the basic principles behind object-oriented programming techniques, including objects, classes, inheritance, and polymorphism. For example, you will learn how to create a main function, work with variables and constants, and create comments. Writing these sets of instructions, which are known as programs or software, requires using a computer programming language and resolving any errors in the instructions so that the programs work correctly.

Programs are also frequently called application programs, or simply applications, because you apply them to a task such as preparing payroll checks, creating inventory reports, or—as in the case of game programs—even entertaining someone. As with any language, learning a computer programming language requires learning both vocabulary and syntax. The rules of any language make up its syntax. In English, using incorrect syntax—that is, committing a syntax error—might make communication more difficult but usually does not prevent it altogether.

If you are vague or spell a word wrong when writing, most people will nevertheless understand your message. Computers are not nearly as flexible as most people. When you write programs, you write program statements that are instructions that are similar to English-language sentences. The statements you write in a programming language must be subsequently translated into machine language. Machine language is the language that computers can understand; it consists of 1s and 0s. A translator program called either a compiler or an interpreter checks your program for syntax errors.

If there are no errors, the translator changes your written program statements into machine language. Therefore, syntax errors are not a big problem; you always have an opportunity to fix them before you actually attempt to run the program. Usually, you do not choose whether to use a compiler or an interpreter; instead, the software you use to develop programs in a particular language contains one or the other.

Finding logical errors is much more time consuming for a programmer than finding syntax errors. For a program that is supposed to add two numbers and show the sum, logical errors arise when multiplication is used instead of addition, or when the sum is given before the arithmetic occurs. The language compiler will not tell you when you have committed a logical error; only running and testing your program will enable you to find inappropriate statements. You run a program by issuing a command to execute—that is, to carry out—the program statements. You test a program by using sample data to determine whether the program results are correct. For example, imagine that you write a program to add two numbers, and test the program with the values 2 and 2. You cannot be sure that the program is free of logical errors just because the answer is 4. Perhaps you used the multiplication symbol rather than the addition symbol.

For example, if you misspell a programming language word, you commit a syntax error, but if you use a correct word in the wrong context, you commit a semantic error. Identify the false statement and explain why it is false. The grammatical rules of any language make up its logic; violating these rules is a logical error. A translator program called either a compiler or an interpreter checks your program for syntax errors and coverts code to machine language. The false statement is 1. Using statements in the correct order at the correct time are its logic. For example, all programming languages provide methods for directing output—the information produced by a program—to a desired object, such as a monitor screen, printer, or file. Similarly, all programming languages provide methods for sending input—the data provided by an outside source, such as a keyboard, scanner, or file—into computer memory so that a program can manipulate it.

In addition, all programming languages provide a way to name locations in computer memory. These locations are commonly called variables. When writing a computer program, yourAge becomes the name of a position or location in computer memory; the value at that location or the state of that location might be 18 or 80, or it might be unknown. All modern programming languages require that variable names be one word; that is, although they do not have to match any word found in a conventional dictionary, they cannot include any embedded spaces. Each programming language has other specific rules as to which characters are not allowed, how many characters may be used in the variable name, and whether capitalization makes a difference. A payroll program, for example, is easier to read if a variable that is meant to hold your salary is called yourSalary, but it is legal—that is, acceptable to the language-translating software—to call the variable ImHungry or jqxBr.

A variable may have only one value at a time, but it is the ability of memory variables to change in value that makes computers and programming worthwhile. Because one memory location, or variable, can be used repeatedly with different values, program instructions can be written once and then used for thousands of problems. The data type of a variable defines what kind of values may be stored in a variable and what kind of operations can be performed on it. Most computer languages allow at least two types: one for numbers and one for characters. Numeric variables hold values like 13 or Many languages include even more specialized types, such as integer for storing whole numbers or floating point for storing numbers with decimal places.

The distinction between variable types is important because computers handle the various types of data differently; each type of variable requires a different amount of storage and answers to different rules for manipulation. Later in the chapter, you will learn to create named constants. All programming languages provide methods for directing output to a monitor screen, printer, or file and for sending input into a computer program so that it can be manipulated. All modern programming languages allow you to declare variables which can have multiple values at the same time.

The false statement is 2. Procedural programs consist of a series of steps or procedures that take place one after the other. The programmer determines the exact conditions under which a procedure takes place, how often it takes place, and when the program stops. Although each language has a different syntax, they all share many elements. Over the years, as programmers have sought better ways to accommodate the way people work best on computers, procedural programming techniques have evolved into objectoriented techniques. Each step was then coded in an appropriate language. Consider a program that creates customer bills for a small business. If you could write the program in English rather than in a programming language, a simple version of the program might look something like Figure He or she also chooses unique, descriptive variable names, such as customerName and balanceDue. Most companies would store firstName and lastName in separate fields, but use both on a customer bill.

Similarly, streetAddress, city, state, and zipCode are likely to be separate variables. The example in Figure uses customerName and customerAddress to limit the number of statements. Three basic control structures are used in procedural programming. In the sequence structure, program steps execute one after another, without interruption. The order of some of the statements is important; for example, when producing a bill, you must determine the balanceDue before you can print it out. For some other statements, however, order is unimportant.

Procedural programs can also include a second control structure called a selection structure, which you use to perform different tasks based on a condition. Figure shows how this is accomplished. The selection occurs in the second statement from the bottom of the program. The actual program that produces bills for a company might have many more selection statements than this example and is usually far more detailed. What about taxes? What if the customer has a credit balance that should be applied to this order? What if one of the data items like quantityOrdered or customerAddress has been left blank? Some programs contain dozens or even hundreds of selection statements. The third control structure used in computer programs is the loop structure, which repeats actions while some condition remains unchanged. When companies bill customers, they usually bill many customers at one time.

The relevant program accesses a customer record from 6 C H A P T E R O N E an input file, produces a bill, and continues to repeat the same steps until no more customers remain in the file. The example in Figure shows a program that loops. The billing program shown in Figure does not contain nearly as many sequential steps, selections, or loops as a full-blown production program would, but even as it stands, the billing program contains quite a few statements. Pearson Education India publishes academic books and reference books in various fields like business and management, computer science and other engineering domains, competitive exam guides among other types of books. Certified Buyer , Gaya. Certified Buyer , Nagpur. Certified Buyer , Patna. Certified Buyer , New Delhi. Certified Buyer , Chennai. Certified Buyer , Jabalpur.

Certified Buyer , Tiruvallur. Certified Buyer , Hisar. Explore Plus. Higher Education and Professional Books. Computing and Information Technology Books. Special price ends in less than 17h: 23m: 00s. Enter pincode. Usually delivered in 3 days? Lafore Robert. TrueComRetail 4. About Pearson Pearson Education has been publishing books on all genres like science, technology, law, business, humanities and others, and has been educating more than a hundred million people across the world. Frequently Bought Together. Programming with Java. Computer System Architecture 3e Update by Pearson. Add 3 Items to Cart. Rate Product.

It's just too good book in contrast with E Balagurusamy or Yashwant Kanetkar book. We use cookies to provide our services , for example, to keep track of items stored in your shopping basket, prevent fraudulent activity, improve the security of our services, keep track of your specific preferences e. Performance and Analytics. ON OFF. We use cookies to serve you certain types of ads , including ads relevant to your interests on Book Depository and to work with approved third parties in the process of delivering ad content, including ads relevant to your interests, to measure the effectiveness of their ads, and to perform services on behalf of Book Depository.

Cancel Save settings. Home Contact us Help Free delivery worldwide. Free delivery worldwide. Bestselling Series. Harry Potter. Popular Features. Home Learning. Notify me. Educational Supplement Suggested solutions to the programming projects found at the end of each chapter are made available to instructors at recognized educational institutions. This educational supplement can be found at www. Other books in this series. Add to basket. Neon Colouring Kit with 6 highlighters: Butterflies -. Unix Shell Programming Stephen G. Kaleidoscope Coloring: Purrmaids, Llamacorns, and More! Educational SupplementSuggested solutions to the programming projects found at the end of each chapter are made available to instructors at recognized educational institutions. Table of contents Introduction. A Note to Teachers.

Jeruzalem download PDF
Sport Finance 2nd edition download pdf, epub
The Healing Power of the Christmas Rose : The Medicinal Value of Black Hellebore free book
Eat. Play. Love. : Life Lessons from My Dog download ebook
Your Attitude Defines Your Direction In Life - Composition Notebook : Motivation and Habit Quote (WI download pdf, epub
Professor Birdsongs 117 Dumbest Criminal Stories : The Southwest free book
American Pendulum Recurring Debates in U. S. Grand Strategy 1st edition free ebook