Computer Science Structured Programming Approach in C 4th edition Forouzan Solutions Manual

SKU: 9780357506134-SOLUTIONS

Solution and Answer Guide

AFYOUNI/FOROUZAN, COMPUTER SCIENCE: A STRUCTURED PROGRAMMING APPROACH IN C, 4e,
©2023, 9780357506134, Chapter 1: INTRODUCTION TO COMPUTERS

TABLE OF CONTENTS

Review Questions 1

Exercises 6

Problems 9

Note to instructor: It is recommended that students be required to respond, where appropriate, with answers that are complete sentences.

REVIEW QUESTIONS

1. Computer software is divided into two broad categories: system software and operational software.

   a. True

   b. False

   Answer: b. False

   Feedback: Computer software is divided into two categories: system software and application software.

2. The operating system provides services such as a user interface, file and database access, and interfaces to communications systems.

   a. True

   b. False

   Answer: a. True

   Feedback: Operating system is the interface between the user and the computer hardware.

3. The first step in system development is to create a source program.

   a. True

   b. False

   Answer: b. False

   Feedback: The first phase in system development is system requirements phase.

© 2023 Cengage. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Computer Science Structured Programming Approach in C 4th edition Forouzan Solutions Manual

4. The programmer design tool used to design the whole program is the flowchart.

   a. True

   b. False

   Answer: b. False

   Feedback: Flowchart is a tool to represent the flow of data through a program and how it is processed.

5. Blackbox testing gets its name from the concept that the program is being tested without knowing how it works.

   a. True

   b. False

   Answer: a. True

   Feedback: Blackbox testing is a concept of testing where the test engineer and the user perform the testing without knowing how the program is actually built.

6. Which of the following is a component(s) of a computer system?

   a. Hardware

   b. Software

   c. Both hardware and software

   d. Pseudocode

   e. System test

   Answer: c. Both hardware and software

   Feedback: Pseudocode is a tool that describes in part English the algorithm of the program.

7. Which of the following is not an example of application software?

   a. Database management system

   b. Language translator

   c. Operating system

   d. Accounting system

   e. Virus detection

   Answer: c. Operating System

   Feedback: Operating system is not an application; it is a required software to interface with the computer hardware.

© 2023 Cengage. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Computer Science Structured Programming Approach in C 4th edition Forouzan Solutions Manual

8. Which of the following is not a computer language?

   a. Assembly/symbolic language

   b. Binary language

   c. High-level languages

   d. Machine language

   e. Natural language

   Answer: b. Binary language

   Feedback: a, c, d, and e. Assembly/symbolic, high-level, machine, and natural are categories of computer languages.

9. The computer language that most closely resembles machine language is _____.

   a. assembly/symbolic

   b. COBOL

   c. FORTRAN

   d. high level

   Answer: a. assembly/symbolic

   Feedback:

   b and c. COBOL and FORTRAN are high-level computer languages.

   d. High level is not a computer language.

10. The tool used by a programmer to convert a source program to a machine language object module is a _____.

    a. compiler

    b. language translator

    c. linker

    d. preprocessor

    e. text editor

    Answer: a. compiler

    Feedback:

    b. Language translator is a part of the compiler.

    c. The linker assembles all input/output processes and mathematical library functions.

    d. Preprocessor is part of the compiler.

    e. Text editor is used to create the program.

© 2023 Cengage. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Computer Science Structured Programming Approach in C 4th edition Forouzan Solutions Manual

11. The _____ contains the programmer's original program code.

    a. application file

    b. executable file

    c. object file

    d. source file

    e. text file

    Answer: d. source file

    Feedback:

    a. Application file could be a configuration file that contains settings for the application.

    b. Executable file contains compiled and linked code that the user can run.

    c. Object file is the file that contains the converted source code in machine language.

    e. Text file contains any text and not necessarily code.

12. The series of interrelated phases that is used to develop computer software is known as _____.

    a. program development

    b. software engineering

    c. system development life cycle

    d. system analysis

    e. system design

    Answer: c. system development life cycle

    Feedback:

    a. Program development is not a framework or methodology.

    b. Software engineering is an area of study in computer science.

    d and e. System analysis and design are phases of software development.

13. The _____ is a program design tool that is a visual representation of the logic in a function within a program.

    a. flowchart

    b. program map

    c. pseudocode

    d. structure chart

    e. waterfall model

© 2023 Cengage. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Computer Science Structured Programming Approach in C 4th edition Forouzan Solutions Manual

Answer: a. flowchart

Feedback:

b. Program map is not a valid tool.

c. Pseudocode is in part English and not visual.

d. Structure chart does not depict the logic of the program.

e. Waterfall model is a software development methodology.

14. The test that validates a program by ensuring that all of its statements have been executed—that is, by knowing exactly how the program is written—is _____.

a. blackbox testing

b. destructive testing

c. nondestructive testing

d. system testing

e. whitebox testing

Answer: e. whitebox testing

Feedback:

a. Blackbox testing is a testing method where the tester does not know how the program is built.

b, c, and d. Destructive, nondestructive, and system testing are not valid testing methods in software engineering.

15. Which of the following is not an advantage of an Agile software development model?

a. Rapid development

b. Customer involvement

c. Very structured

d. Adaptive

e. Team collaboration

Answer: c. Very structured

Feedback:

a, b, d, and e. Rapid development, customer involvement, being adaptive, and team collaboration are major advantages of an Agile software development model.

© 2023 Cengage. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Computer Science Structured Programming Approach in C 4th edition Forouzan Solutions Manual

EXERCISES

16. Describe the two major components of a computer system.

Answer:

The two major components of a computer system are hardware and software. The hardware component of the computer system is made of five parts: the input devices, central processing unit (CPU), primary storage or main memory, output devices, and auxiliary storage devices. The software consists of system software, which includes the operating system, and application software used to solve the user's business requirements.

17. Computer hardware is made up of five parts. List and describe them.

Answer:

a. Central processing unit (CPU): It is responsible for the operations in the computer, such as arithmetic calculations, comparisons among data, and movement of data inside the computer.

b. Primary memory: It is a place where the programs and data are stored temporarily during processing. It will be erased when we turn off a personal computer or we log off from a time-sharing computer.

c. Input device: It is usually a keyboard where programs and data are entered into the computer. It could also be other devices such as a mouse, a pen or stylus, a microphone, or a touch screen device.

d. Output device: It is usually a monitor (screen or video) or a printer where the output will be shown. If the result is shown on the monitor, we say we have a soft copy. If it is printed on the printer, we say we have a hard copy.

e. Auxiliary storage device (secondary storage): It is a place where the programs and data are stored permanently. When we turn off the computer, our programs and our data remain in the secondary storage ready for the next time we need them. This includes devices such as disks (hard disks or floppy disks), tapes, or CDs.

18. Describe the major differences between a time-sharing and a client/server environment.

Answer:

In a time-sharing environment, each user has a terminal that does not have any processing capability of its own; all processing is done by a central computer. In a client/server environment, users have terminals that have some processing capabilities; a portion of the processing is done by the terminal workstation, and a portion is done by a central computer.

© 2023 Cengage. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Computer Science Structured Programming Approach in C 4th edition Forouzan Solutions Manual

19. Describe the two major categories of software.

Answer:

There are two categories of computer software: system software and application software. System software keeps the hardware running and it provides an interface between the hardware and the user, but does nothing to directly solve the user's needs. Application software, on the other hand, is directly responsible for helping users solve their business problems.

20. What is the purpose of an operating system?

Answer:

The operating system provides system services such as a user interface, file and database access, and communication services. Its primary purpose is system efficiency while providing user access to the hardware and applications.

21. Identify at least two types of system software that you will use when you write programs.

Answer:

System software used to develop our programs includes the text editor to create the program, a compiler to convert it to machine language, and debugging tools.

22. Give at least one example of general-purpose and one example of application-specific software.

Answer:

General-purpose software can be used for more than one purpose. Examples include word processors, spreadsheets, and database management systems. Application-specific software solves a specific business problem and cannot be used for other purposes. Examples include personal finance systems and general ledger accounting systems.

23. List the levels of computer languages discussed in the text.

Answer:

There are four levels of computer languages from the most primitive one to today's highly productive ones:

a. Machine languages

b. Symbolic languages

c. High-level languages

d. Natural languages

© 2023 Cengage. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Computer Science Structured Programming Approach in C 4th edition Forouzan Solutions Manual

24. What are the primary differences between symbolic and high-level languages?

Answer:

Symbolic languages, often called assembly languages, provide mnemonics for machine instructions, data identifiers, and other objects such as functions. They allow the programmer to write program instructions that basically mirror the machine instructions. High-level languages, on the other hand, are machine independent and allow the user to concentrate on the problem being solved rather than the hardware on which it is being solved. Generally, each high-level language statement generates many machine language statements.

25. What is the difference between a source program and an object module?

Answer:

A source program is written in a text editor so that it can be read or edited. An object module has been compiled into machine language from a source program, so we cannot read it.

26. Describe the basic steps in the system development life cycle.

Answer:

a. System requirements: Define the requirements for the system.

b. Analysis: Evaluate alternative solutions to requirements.

c. Design: Describe the specific implementation for the problem.

d. Code: Prepare and unit test programs based on the design.

e. System test: Verify that the programs integrate and work as a system to satisfy the user requirements.

f. Maintenance: Keep the system working in production.

27. What documentation should a programmer receive to be able to write a program?

Answer:

A programmer should receive:

a. The program requirements statement

b. The design of any program interfaces

c. An overview of the complete project

28. List and explain the steps that a programmer follows in writing a program.

Answer:

The four steps to develop a program are:

a. Understand the problem.

b. Develop a solution.

c. Write the program.

d. Test the program.

© 2023 Cengage. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Computer Science Structured Programming Approach in C 4th edition Forouzan Solutions Manual

29. Describe the three tools that a programmer may use to develop a program solution.

Answer:

There are three tools used to develop a program solution:

a. Structure chart: It shows the functional flow through our program. In other words, it shows how we are going to break our program into logical steps; each step will be a separate module and the whole structure chart shows the interaction between all of the modules.

b. Flowchart: It uses standard graphical symbols to represent the logical flow of data through a function.

c. Pseudocode: It is part English, part program logic. Its purpose is to describe, in a precise algorithmic detail, what the program being designed is supposed to do.

30. What is meant by the old programming proverb, “Resist the temptation to code“?

Answer:

“Resist the temptation to code” means that the programmer must fully understand the problem and design a solution before beginning the process of writing code. It is human nature to want to get to the coding step as soon as possible, but this often leads to poorly implemented and inefficient programs.

31. What is the difference between blackbox and whitebox testing?

Answer:

Blackbox testing consists primarily of testing based on user requirements and assumes no knowledge of the inner workings of the program while whitebox testing, executed by the programmer, tests the program with full knowledge of the program’s operation.

32. What is software engineering?

Answer:

Software engineering is the use of sound engineering methods and principles to develop software that works.

PROBLEMS

33. Write pseudocode for `calcLivingAreas`, based on the structure chart shown in Figure 1-14.

Answer:

Algorithm `calcLivingAreas`

1. Prompt user for famRoom width
2. Read famRoom width
3. Prompt user for famRoom length
4. Read famRoom length

© 2023 Cengage. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Computer Science Structured Programming Approach in C 4th edition Forouzan Solutions Manual

5. famRoom area = famRoom width * famRoom length
6. Prompt user for dineLive width
7. Read dineLive width
8. Prompt user for dineLive length
9. Read dineLive length
10. dineLive area = dineLive width * dineLive length
11. Living areas = famRoom area + dineLive area

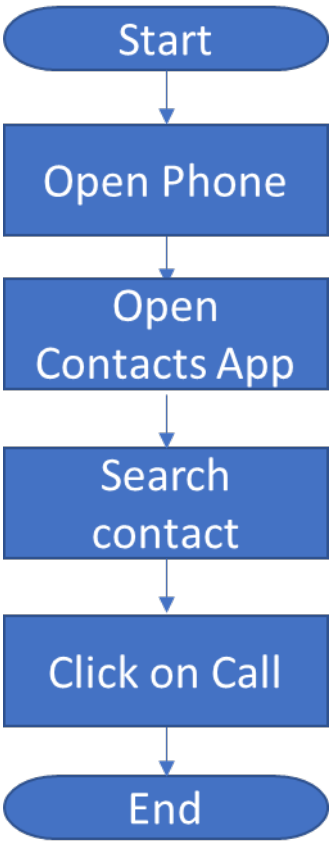
end Algorithm `calcLivingAreas`

34. Create a flowchart for a routine task, such as calling a friend, that you do on a regular basis.

Answer:

No standard answer.

Possible steps are:



© 2023 Cengage. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Computer Science Structured Programming Approach in C 4th edition Forouzan Solutions Manual

35. Write pseudocode for the flowchart you created in Problem 34.

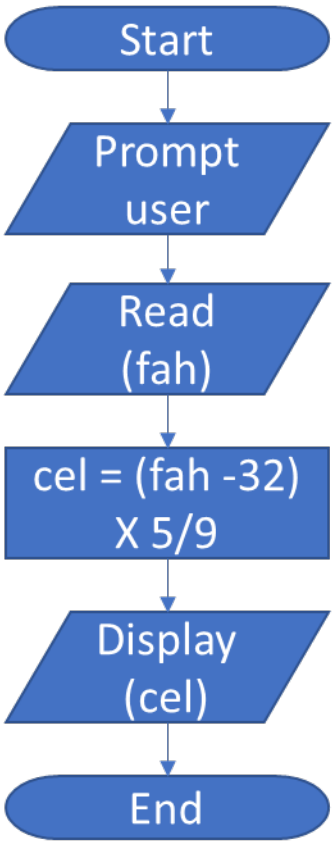
Answer:

No standard answer.

Possible steps are:

1. Open phone
2. Open contact app
3. Search for friend contact
4. Click on Call icon

36. Create a flowchart to convert Fahrenheit temperature to Celsius and then write pseudocode for the flowchart.




```
Prompt user to enter Fah
Read fah value
Set cel to (fah-32)X5/9
Display cel value
stop
```

© 2023 Cengage. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Computer Science Structured Programming Approach in C 4th edition Forouzan Solutions Manual

Instructor Manual

Afyouni, Computer Science: A Structured Programming Approach In C, ISBN 9780357506134, Chapter 1: Introduction to Computers

TABLE OF CONTENTS

Purpose and Perspective of the Chapter **2**
List of Student Downloads ..... **Error! Bookmark not defined.**
Chapter Objectives **2**
What's New in This Chapter **2**
Chapter Outline **3**
Additional Resources ..... Error! Bookmark not defined.
Cengage Video Resources ..... **Error! Bookmark not defined.**
External Videos or Playlist ..... **Error! Bookmark not defined.**
Internet Resources ..... **Error! Bookmark not defined.**
Primary Sources ..... **Error! Bookmark not defined.**
Cengage Audio Resources ..... **Error! Bookmark not defined.**
External Audio Resources ..... **Error! Bookmark not defined.**
List of Transcripts for Audio Assignments ..... **Error! Bookmark not defined.**

© 2023 Cengage. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Computer Science Structured Programming Approach in C 4th edition Forouzan Solutions Manual

PURPOSE AND PERSPECTIVE OF THE CHAPTER

The purpose of this chapter is to give students an overview of computer systems. First, basic computer system concepts are introduced, including a review of hardware and software. Next, we look at the different computing environments and their components. The different classifications of computer languages are also reviewed. Finally, we take a closer look at the steps in the development of a computer program, concluding with program testing.

CHAPTER OBJECTIVES

The following objectives are addressed in this chapter:

1.1 Describe basic computer system concepts.

1.2 Identify the different computing environments and their components.

1.3 List and describe the classifications of computer languages.

1.4 Identify the steps in the development of a computer program.

1.5 Describe the system development life cycle (SDLC).

WHAT'S NEW IN THIS CHAPTER

The following elements are improvements in this chapter from the previous edition:

- The evolution of computer languages
- Computing environments, specifically cloud computing
- Waterfall and Agile system development life cycles
- Updated end-of-chapter review questions, exercises, problems, and projects

[return to top]

© 2023 Cengage. All Rights Reserved. May not be scanned, copied or duplicated,
or posted to a publicly accessible website, in whole or in part.

Computer Science Structured Programming Approach in C 4th edition Forouzan Solutions Manual

CHAPTER OUTLINE

In the outline below, each element includes references (in parentheses) to related content. "CH.##" refers to the chapter objective; "PPT Slide #" refers to the slide number in the PowerPoint deck for this chapter (provided in the PowerPoints section of the Instructor Resource Center). Introduce the chapter and use the Ice Breaker in the PPT if desired, and if one is provided for this chapter. Review learning objectives for Chapter 1 (PPT Slide 3).

I. Computer Systems (01.01, PPT Slides 5-11)
   a. Computer systems are found everywhere. They are an essential part of daily life, facilitating interactions with family and friends, governments, and small and large businesses. A computer is a system made of two major components: hardware and software. The computer hardware is the physical equipment. The software is the collection of programs (instructions) that allow the hardware to do a specific job.
   b. The hardware component of a computer system consists of five parts: input devices, central processing unit (CPU), primary storage, output devices, and secondary storage devices such as internal and external hard drives, USB (Universal Serial Bus) devices, and external backup tapes.
   c. Computer software is divided into two broad categories: system software and application software. System software manages the computer resources. Application software, on the other hand, is directly responsible for helping users solve their problems.
   d. System software consists of programs that manage a computer's hardware resources and perform information-processing tasks. These programs are divided into three classes: the operating system, system support, and system development.
   e. Application software is divided into two classes: general-purpose software and application-specific software. The first, general-purpose software, is purchased from a software developer and can be used for more than one application. The second type, application-specific software, can be used only for its intended purpose.

II. Computing Environments (01.02, PPT Slides 13-22)

© 2023 Cengage. All Rights Reserved. May not be scanned, copied or duplicated,
or posted to a publicly accessible website, in whole or in part.

Computer Science Structured Programming Approach in C 4th edition Forouzan Solutions Manual

a. In the early days of computers, there was only one environment, the mainframe computer hidden in a central computing department. With the advent of minicomputers and personal computers, the environment changed, with computers on virtually every desktop.
b. In 1971, Marcian E. Hoff, working for Intel, combined the basic elements of the central processing unit into the microprocessor. This first computer on a chip was the Intel 4004 and was the grandparent many times removed of the chips now used in computers around the world. The rapid development of computer chips ultimately led to the transition from large mainframe computers to a smaller, self-contained device known as a personal computer in the 1970s.
c. The term PC (short for “personal computer”) is now used to refer primarily to computers that run Microsoft operating systems. The second type of personal computer is those manufactured exclusively by Apple. The early versions of these computers were designed to sit on a desktop. Now, personal computing devices take several forms, including smart phones, tablets, laptops, desktop setups with monitors and towers, and all-in-one desktop computers, which combine the CPU and the monitor into one device.
d. In the time-sharing environment, many users are connected to one or more computers. These computers may be minicomputers (nowadays known as servers) or central mainframes. All computing must be done by the central computer.
e. A client/server computing environment splits the computing function between a central computer and users’ computers. In the client/server environment, the users’ workstations are called the clients. The central computer, which may be a powerful computer, minicomputer, or central mainframe system, is known as the server.
f. Distributed computing is a large network of servers and clients scattered geographically to provide a seamless integration of computing functions. All resources are pooled together to provide high processing power for applications. This environment provides a reliable, scalable, and highly available network.
g. In a cloud computing environment, servers and storage devices are spread out across multiple geographic areas and connected via the Internet. Cloud computing environments provide services such as Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS).

III. Computer Languages (01.03, PPT Slides 24-28)
a. To write a program for a computer, we must use a computer language. Over the years computer languages have evolved from machine languages to natural languages. In the earliest days of computers, the only programming

© 2023 Cengage. All Rights Reserved. May not be scanned, copied or duplicated,
or posted to a publicly accessible website, in whole or in part.

Computer Science Structured Programming Approach in C 4th edition Forouzan Solutions Manual

languages available were machine languages. Each computer has its own machine language, which is made of streams of 0s and 1s.

b. Symbolic languages mirrored machine languages using symbols, or mnemonics, to represent the various machine language instructions. Because symbolic languages had to be assembled into machine language, these languages became known as assembly languages.

c. The desire to improve programmer efficiency and to change the focus from the computer to the problem being solved led to the development of high-level languages. High-level languages are portable to many different computers, allowing the programmer to concentrate on the application problem at hand rather than the intricacies of the computer. High-level languages are designed to relieve the programmer from the details of the assembly language. High-level languages share one thing with symbolic languages, however: they must be converted to machine language. The process of converting them is known as compilation.

d. **Activity 1.1: Knowledge Check (01.01, 01.02, PPT Slides 29-31)**

   i. What are the five parts of a computer system's hardware component?
   Answer: The five parts of a computer system's hardware component are input devices, a central processing unit (CPU), primary storage, output devices, and secondary storage devices.

   ii. Explain the difference between machine language and symbolic language.
   Answer: Machine language is made of streams of 0s and 1s. Symbolic languages use symbols, or mnemonics, to represent the various machine language instructions.

IV. Creating and Running Programs (01.04, PPT Slides 33-38)

a. It is the job of the programmer to write and test programs. There are four steps in this process: (1) writing and editing the program, (2) compiling the program, (3) linking the program with the required library modules, and (4) executing the program.

b. The software used to write programs is known as a text editor. In a text editor, you can enter, change, and store character data. Text editors include features designed to make writing code easier, including applying colors to specific parts of a program. After completing a program, you need to save the file so it can be input to the compiler. This saved file is known as a source file.

c. A compiler translates the code in a source file into machine language. The compiler consists of two separate programs: the preprocessor and the translator. The preprocessor reads the source code and prepares it for the translator, producing a translation unit. The translator reads the translation

© 2023 Cengage. All Rights Reserved. May not be scanned, copied or duplicated,
or posted to a publicly accessible website, in whole or in part.

Computer Science Structured Programming Approach in C 4th edition Forouzan Solutions Manual

unit and writes the resulting object module to a file that can then be combined with other precompiled units to form the final program.

d. When you write a C program, you might actually write some of the functions yourself. However, some functions, such as input/output processes and mathematical library functions, exist elsewhere and must be attached to the program. The linker assembles all of these functions, as well as the system's, into a final executable program.
e. After your program has been linked, it is ready for execution. To execute a program, you use an operating system command, such as `run`, to load the program into primary memory and execute it. Getting the program into memory is the function of an operating system program known as the loader. It locates the executable program and reads it into memory.
f. **Activity 1.2: Think, Pair, and Share (01.04, PPT Slide 39)**
    i. Form pairs/groups of two to four class members.
    ii. Your group will practice working with a text editor.
    iii. Take turns using a variety of text editor features, such as using search and formatting commands.
    iv. Practice using copy and paste commands to move statements from one part of the program to another.

V. System Development (01.05, PPT Slides 41-45)

a. Developing a program is a critical process that determines the overall program quality and success. If we carefully design programs using good structured development techniques, they will be efficient, error-free, and easy to maintain.
b. Today's large-scale, modern programming projects are built using a series of interrelated phases commonly referred to as the system development life cycle. One very popular development life cycle is the waterfall model. Agile, a widely used software development methodology, takes an adaptive, iterative approach to software development, with the goal of expediting the development life cycle and involving the user early in the development process.
c. Program development involves these major steps: understanding the problem, developing a solution, writing the program, and then testing it. Understanding the problem involves reading the requirements statement carefully, reviewing your understanding, and asking questions. After you fully understand the problem and have clarified any questions you may have, the next step is to develop the solution. Three tools will help in this task: (1) structure charts, (2) pseudocode, and (3) flowcharts.
d. After you write a program, you must test it. Program testing can be a very tedious and time-consuming part of program development. Blackbox testing gets its name from the concept of testing the program without

© 2023 Cengage. All Rights Reserved. May not be scanned, copied or duplicated,
or posted to a publicly accessible website, in whole or in part.

Computer Science Structured Programming Approach in C 4th edition Forouzan Solutions Manual

knowing what is inside it—that is, without knowing how it works. Whereas blackbox testing assumes that the tester knows nothing about the program, whitebox testing assumes that the tester knows everything about the program. Blackbox testing is done by the system test engineer and the user. Whitebox testing is the responsibility of the programmer.

e. **Activity 1.3: Discussion (01.05, PPT Slide 46)**
   i. What is the difference between blackbox and whitebox testing? Who is responsible for each?
   ii. Why do you think testing important?

f. **Self-Assessment (01.05, PPT Slide 47)**
   i. What basic computer system concepts did you find difficult and thus need to review?
   ii. What are your initial thoughts on developing computer programs?

[return to top]

© 2023 Cengage. All Rights Reserved. May not be scanned, copied or duplicated,
or posted to a publicly accessible website, in whole or in part.

Computer Science Structured Programming Approach in C 4th edition Forouzan Solutions Manual