David Keil Computer Science I with Java 9/09

Study Questions for CS I

Intro
T1 (Design)
T2
T3
T4 (standard types)
T6
T7
T7
T8
Multiple topics

Study questions on Introduction

1. The discipline of computer science

1. The study of computer science is most generally concerned with solving problems about (a) intelligence; (b) entertainment; (c) business; (d) information; (e) none of these
2. Information processing can occur in _____ systems (a) natural; (b) human-made; (c) natural or human-made; (d) only computer; (e) none of these
3. We most generally use _____ to help solve information-processing problems (a) trial and error; (b) force; (c) abstraction; (d) genius; (e) none of these

2. Computation and symbol manipulation

1. An abstraction by which we may represent parts of the real world is most generally a(n) (a) symbol; (b) problem; (c) solution; (d) bit; (e) none of these
2. The brain has ___ components than a computer but these components have the advantage of ___ (a) fewer, sequentiality; (b) slower, parallelism; (c) slower, objectivity; (d) fewer, precision; (e) none of these
3. A model is an abstract description that lets us focus on (a) maximum detail; (b) details of our choice; (c) essentials; (d) particular cases; (e) none of these
4. With _____, input may depend partly on previous output (a) interaction; (b) any computation; (c) algorithms; (d) all programs; (e) none of these

3. Some mathematics of computing

1. In the unary system, 12 is represented (a) 12; (b) 0b; (c) 1100; (d) 111111111111; (e) none of these
2. The natural numbers are a(n) (a) problem; (b) solution; (c) set; (d) illusion; (e) none of these

4. Computing today

1. An operating system runs (a) at brief intervals; (b) at all times when the computer is running; (c) at the speed of light; (d) when invoked by an application; (e) none of these
2. In client-server computing, the ____ responds to requests from the _____ (a) client, user; (b) client, server; (c) server, client; (d) server, user; (e) none of these
4. A server is (a) a computer; (b) a peripheral; (c) an application program; (d) a program that runs on all network workstations; (e) an agent
5. Email and Web data always travel in the form of (a) waves; (b) machine words; (c) packets; (d) HTML files; (e) none of these
6. In a user PC on the FSC network, the *Y:* drive is (a) a processor; (b) a physical device; (c) a folder on the user computer's hard drive; (d) a folder on the server; (e) none of these
7. An operating system provides services for (a) applications; (b) remote sites; (c) hardware; (d) Microsoft Corp.; (e) surgeons
8. Right-clicking on an object (a) opens or executes it; (b) deletes it; (c) selects it; (d) displays its operations and attributes; (e) none of these
9. Which is *not* hardware? (a) general-purpose computer; (b) operating system; (c) video game console; (d) printer; (e) all are hardware.

Study questions on event-driven and GUI apps

1. HTML and its extensions

1. HTML is mainly a(n) _____ notation (a) procedural; (b) formatting; (c) algorithmic; (d) interactive; (e) none of these
2. HTML supports formatting features using (a) tags; (b) operators; (c) methods; (d) user *commands*; (e) none of these
3. When a hyperlink is clicked (a) the user obtains exclusive use of the web server linked to; (b) the browser searches the Internet for the web page referenced; (c) the browser downloads the Internet file referenced by the hyperlink; (d) a search engine goes into action; (e) none of these
4. HTML supports interaction via (a) tags; (b) extensions such as JavaScript; (c) methods; (d) user commands; (e) none of these

2. Programming in JavaScript

1. JavaScript encodes (a) web-page formatting; (b) responses to input events; (c) database design; (d) statistical analysis; (e) none of these
2. JavaScript code is likely to appear in (a) spreadsheet formulas; (b) processor registers; (c) database queries; (d) HTML files; (e) none of these
3. Of the following, which is a high-level programming language? (a) assembler; (b) HTML; (c) machine; (d) JavaScript; (e) none of these
4. An event handler is (a) a program; (b) program code; (c) a browser; (d) a job category; (e) none of these
5. JavaScript is a _____ language (a) markup; (b) procedural; (c) functional; (d) machine; (e) none of these
6. *alert* is a(n) (a) JavaScript method; (b) HTML event; (c) HTML tag; (d) user warning; (e) none of these

3. Elements of JavaScript

1. In JavaScript, "+" signifies (a) addition; (b) concatenation; (c) addition and concatenation; (d) any operation; (e) none of these
2. Variables may be assigned values in (a) JavaScript statements; (b) HTML assignment statements; (c) algebra; (d) packet transmission; (e) none of these
3. *ParseInt* (a) is a variable; (b) converts strings to integers; (c) converts numbers to strings; (d) is a language; (e) none of these
4. A JavaScript assignment can change (a) the current URL; (b) a variable's value; (c) a variable's name; (d) user ID; (e) none of these
5. To use a JavaScript variable, it must be (a) requested; (b) initialized; (c) promulgated; (d) declared; (e) none of these

4. Some programming concepts

1. Interface refers most generally to how an application (a) sends data; (b) responds to the user; (c) retrieves data from the server; (d) sends data to server; (e) none of these
2. Implementation is (a) interface; (b) design; (c) coding; (d) analysis; (e) none of these
3. A window is a(n) (a) method; (b) class; (c) object; (d) operation; (e) none of these

Short answer

1. What does the following JavaScript code do?
   (a) *var a, b, total*
   (b) *x = ParseInt(prompt("Enter your age", ""))*
   (c) *alert("Your age is " + x)*
2. Describe two meanings of "+" in JavaScript.
3. Contrast JavaScript to HTML.
4. Describe how JavaScript and HTML may be used in the same web-resident file.
5. Circle the JavaScript declaration statement below. [show code]
6. Write a JavaScript statement that displays a rectangle with the word "Goodbye" in it and an "OK" button.
7. What is an event handler?

Study questions on Topic 1: Design

1. Specification and UML

1. (T-F) The problem-solving process presented in class places design before coding.
2. UML is (a) a language used for design of interactive systems HTML; (b) machine language; (c) database query language; (d) UML; (e) none of these

2. Object-oriented design

1. Object-oriented design focuses problem solving on (a) categories of things; (b) processes; (c) methods; (d) integers
2. A data item that is defined in terms of properties and operations is (a) simple; (b) a bit; (c) input; (d) an object; (e) binary
3. In event-driven programming, an event is usually (a) input; (b) output; (c) a sequence structure; (d) a program decision; (e) something that happens during web-site development

3. Algorithm-design tools

1. Multiple alternatives (a) are not supported by standard programming; (b) call for use of modules; (c) require multiple diamonds in a flowchart; (d) require repetition; (e) none of these
2. When a problem is complex, the complexity can often be conquered in the design stage by (a) brute force; (b) documentation; (c) modular decomposition; (d) input/output; (e) logic gates
3. (T-F) A branch may be nested inside a loop.
4. (T-F) A component of a structured flowchart has one entrance and one exit.
5. (T-F) The loop is a data structure.
6. (T-F) A flowchart using only three different control structures can diagram a solution to any solvable problem.
7. Pseudocode (a) has a precise syntax; (b) is a false solution; (c) is a low-level language; (d) is an informal notation; (e) none of these
8. The branch is a (a) hardware item; (b) control structure; (c) data structure; (d) flowchart rectangle; (e) module
9. (T-F) The body of a top-tested loop will always execute at least once.
10. Which is *not* a recommended tool for program design? (a) flowcharts; (b) pseudocode; (c) object-oriented analysis; (d) hierarchy charts; (e) use of keywords
11. Which is *not* a control structure? (a) sequence; (b) branch; (c) loop; (d) a file; (e) all are control structures
12. (T-F) The loop is a control structure.
13. Modular decomposition of processes is most closely associated with which kind of design? (a) web-site formatting; (b) spreadsheet; (c) database; (d) algorithm; (e) none of these
14. Which is *not* a feature of algorithms? (a) precision; (b) finiteness of time; (c) step-by-step sequencing; (d) limited set of possible inputs; (e) definiteness of result
15. Which of these is a control structure? (a) hyperlink; (b) Excel worksheet; (c) database table; (d) loop; (e) register
16. Algorithms (a) are by definition efficient; (b) take finite time; (c) are languages; (d) are a kind of program; (e) none of these
17. Design tools include (a) output; (b) flowcharts; (c) registers; (d) queries; (e) none of these
18. The loop is a (a) language; (b) control structure; (c) data structure; (d) program; (e) none of these
19. Control structures are used in (a) design; (b) output; (c) input; (d) formatting; (e) none of these
20. A trace of an algorithm provides (a) input; (b) a list of errors; (c) snapshots of the values of variables over time; (d) a specification of output; (e) none of these

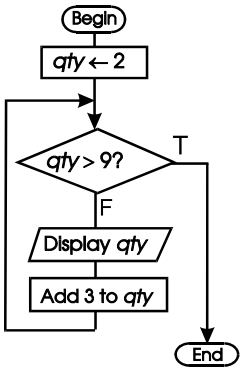
4. Functions and computation

1. A computer program or subprogram may compute a mathematical (a) expression; (b) function; (c) proof; (d) theorem; (e) none of these
2. An operator often corresponds to a(n) (a) interface; (b) function; (c) user; (d) program; (e) none of these
3. A mathematical function is a(n) (a) subprogram; (b) algorithm; (c) mapping between sets; (d) two-way relationship between ideas; (e) none of these

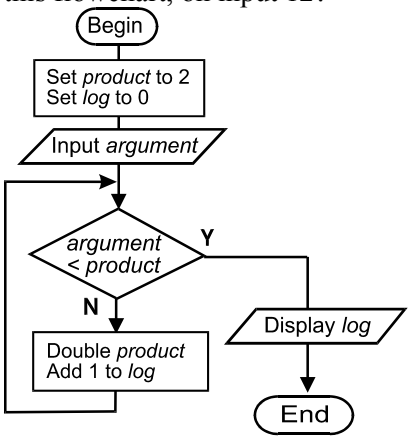
Short answer questions on topic 1

Objective 1: Trace a flowchart

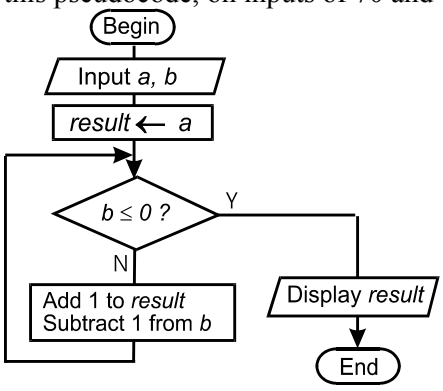
1. What is the output of the program diagrammed by the flowchart below?



2. What output would be produced by a program based on this flowchart, on input 12?

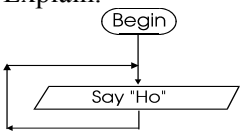


3. What output would be produced by a program based on this pseudocode, on inputs of 70 and 80?



Other

4. What is an algorithm?
5. Distinguish problem specification, system design, and program coding.
6. Put each of the following into one of the shapes (rectangle, parallelogram, or diamond):
   (a) count < 5;
   (b) display total;
   (c) total ← total + *x*.
7. What are two ways to express an algorithm?
8. What kind of structures are the sequence, branch, and loop, and what algorithms can they be used to specify?
9. What are the three standard control structures, sufficient to specify any algorithm?
10. A precise plan to solve a problem in finite time is ____
11. The word *otherwise* might be appropriate in pseudocode for which control structure?
12. In a module hierarchy, what is the location of a module used directly by module *x*?
13. Name the three fundamental control structures recommended in structured programming.
14. In software development, what steps are recommended before coding?
15. Name four chief tools used for writing down a program design before coding.
16. What kind of programming stresses use of three control structures and modular decomposition?
17. What kind of programming stresses defining categories of things and their behaviors?
18. The word *repeat* might be appropriate in pseudocode for which control structure?
19. Counting from 5 to 15 would require which control structure?
20. What is the computer-science term for a precise plan to solve a problem in a finite amount of time?
21. What is pseudocode?
22. Does the flowchart below diagram an algorithm? Explain.



23. In the software development process, what steps are recommended before coding a program? After coding?
24. Give an example of an object that is found on the screen in the Windows or OS/2 user environment. What are some of its data attributes? Its behaviors?
25. How many times will a counter-controlled loop iterate? A sentinel-controlled loop?
26. Name two variants of the loop control structure.

27. Put these phases or sub-phases of the problem-solving process in chronological order, number the first "1", the second "2", etc.
    ____ code program
    ____ desk check
    ____ write a design
    ____ get problem specifications
    ____ test program
    ____ debug code
28. What is a way to describe the relationship between input and output?

29. Label each term below with the letter of its appropriate definition
    ____ pseudocode
    ____ object
    ____ algorithm
    ____ branch
    ____ loop
    ____ module
    ____ desk checking
    ____ stepwise refinement
    ____ top-down design
    a) A precise plan to solve a problem or complete a task in a finite number of steps.
    b) Informal natural-language way to express an algorithm.
    c) The decision control structure, in which one action is taken or else another.
    d) The iteration control structure, in which an action is repeated.
    e) A way to design and code software characterized by use of only three control structures: sequence, branch, and loop.
    f) A data item that is defined partly in terms of its behavior.
    g) Verification of program correctness without running it on a computer.
    h) A program component which may consist of one or more subprograms.
    i) A method of developing a plan for a program, beginning with an overview of the problem and breaking it down.
    j) A method that uses repeated improvements in a program design.

Answers to questions on T1

Multiple-choice or T/F

1. Specification and UML

1. t
2. d

2. Object-oriented design

1. a An object is defined by its attributes and operations
2. d
3. a

3. Algorithm design tools

1. c. A multiway branch corresponds to a series of cascading flowchart diamonds.
2. c. Modular decomposition is a divide-and-conquer stratetgy for problem solving.
3. t
4. t
5. f
6. t
7. d
8. b
9. f
10. e
11. d
12. t
13. d
14. d
15. d
16. b
17. b
18. b
19. a
20. c

4. Functions and computation

1. b
2. b
3. c

Short-answer questions

1. branch
2. A module used by module x will be just below x in a module hierarchy chart
3. A precise plan to solve a problem in a finite number of step
4. Sequence, branch , loop
5. 1. Problem analysis or specification; 2. design
6. Pseudocode; flowcharts; module hierarchy diagram
7. 2 5 8
8. 5432154321
9. Tall
10. structured programming
11. object-oriented programming
12. loop
13. loop
14. algorithm

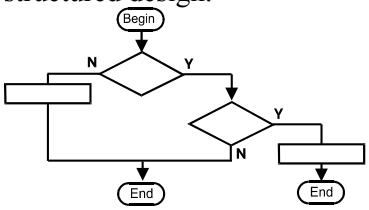
Longer-answer questions

Objective 2: Write a flowchart to solve a loop problem

1. Write a flowchart or pseudocode for algorithms to solve each of the following problems.
   (a) Input exactly 6 signed integers. Display only the largest of the input values, regardless of where it occurred in the input list. *Hint*: Let the data address for the first input also serve as the storage location for the largest integer found so far. Use a second data address for subsequent input values.
   Sample Input: –3,20,-4,5,7,0 *Output:* 20
   (b) Input signed integers until the current input is less than the previous input. Display the largest input value.
   Sample Input: 1,2,3,4,24,56,41 *Output:* 56
2. Design a program to accept keyboard input of three integers to represent the dollar amounts price, discount, and sales tax, in cents. It should display the sum of the price and the tax, minus the discount.
   Sample I/O:
   [Input:] 200
   [Input:] 20
   [Input:] 10
   [Output:] 190
3. Computers are often sold with service plans whose cost depends on the computer's value. Design a program to input the signed integers, *price* and *monitor*. If the sum of these is less than 1000, the program should display the message, "Plan costs $99.95"; otherwise it should display the message, "Plan costs $149.95".
   Sample I/O:
   Example 1: [Input:] 749 [Input:] 199 [Output:] Plan costs $99.95
   Example 2: [Input:] 1299 [Input:] 399 [Output:] Plan costs $149.95
4. Design a program to input three integers, A, B, and C. Make the necessary comparisons to display the greatest of the three.
   Sample I/O:
   [Input:] –38
   [Input:] 300
   [Input:] 77
   [Output:] 300
5. Design a program without input that uses a loop structure to display each of the integers from 1 through 10. The only data values that may be stored initially via data statements are 0, 1, and 10.
6. Design a program that will loop to accept input of exactly three pairs of integers (A,B) and compute and display the value of A – B for each input pair.
7. Design a program to compute and display the product of two input non-negative integers. Display nothing if input includes a negative number. (Hint: Perform the multiplication as repeated addition, using one of the integers as an addend and the other integer as a counter to determine how many times to add the addend to a sum representing the product.)
8. Design a program to input two non-negative integers (A,B) and a positive integer (C). Compute A * B / display the quotient and remainder. For each of the inputs A, B, and C, loop for new input if negative values are entered. The input value of C must also be tested to be sure it is not 0. Why?
9. A *geometric progression* is a sequence of terms in which each term after the first term is obtained by multiplying the previous term by a constant multiplier. For example, if the first term is 7 and the constant multiplier is 3, then the resulting geometric progression is:
   7 21 63 189 567 1701 etc.
10. We can compute the sum of the first *n* terms of a geometric progression. In the preceding example, the sum of the first 5 terms is
    7 + 21 + 63 + 189 + 567 = 847.
11. Design a program that will accept positive integers *n*, *first*, and *k*, input by the user, and display the first *n* terms and then the sum of those terms where the first term is *first*, the constant multiplier is *k*, and the desired number of terms is *n*.
12. Design a program to divide any signed integer by any other non-zero integer, using only addition, subtraction, and the three control structures. First the dividend and then the divisor are to be input from the keyboard. The divisor must be tested to avoid attempted division by 0, since division by 0 is not defined. The result is to be output as an integer quotient followed by an integer remainder. Remember to test your program with all possible sign combinations of the dividend and divisor.
13. Write an algorithm to find the tallest person in a room by comparing two persons at a time. *Hint:* some persons will only have to be compared only once to any other person. Once you compare the shortest person with even one other person, for example, you will know enough never to compare that short person again with anyone.
14. Write pseudocode or a flowchart for a program that displays each number from 1 to 100, putting an asterisk before each number that ends in '7'.
15. Write pseudocode or a flowchart for a program that repeatedly prompts for two quantities, displaying the larger one each time, and terminating when both quantities have the same value.

Other

1. Modify the flowchart below so that it will diagram a structured design.



2. Use pencil and paper to test a few argument values and determine what familiar mathematical functions are computed by the following recurrences:

   (a) $f(a,b) =$

   $$\begin{cases} 0 & \text{if } a = 0 \\ b + f(a-1,b) & \text{otherwise} \end{cases}$$

   (b) (Challenge:) $g(a,b) =$

   $$\begin{cases} 0 & \text{if } a = 0 \\ b + g(\lfloor a \div 2 \rfloor, 2b) & \text{if } a \text{ is odd} \\ g(\lfloor a \div 2 \rfloor, 2b) & \text{if } a \text{ is even} \end{cases}$$

3. Is the pseudocode below an example of a structured design? Why or why not?
   1. If input file exists, open input file; else exit program
   2. Read input file, summing up contents
   3. Display sum.
4. Design a program to input integers, A and B. Compute and display |A – B|. *Note:* Read "|A – B|" as "the absolute value of the difference (A – B)".

Study questions on Topic 2: Hardware

1. The binary system of numerals

1. The binary system is appropriate for digital computers because (a) our hands have ten fingers; (b) binary arithmetic is simpler by nature; (c) binary may be easily translated to hexadecimal notation; (d) binary and decimal values are easily converted; (e) digital computers are based on two-state devices
2. $0011_2 + 0010_2$ = (a) 0101_2; (b) 0100_2; (c) 1000_2; (d) 0111_2; (e) 1011_2
3. (T-F) A yes-or-no answer to a question must be stored in at least three bytes of memory.
4. 3 = (a) 111_2; (b) 11_2; (c) 10_2; (d) 11000_2; (e) none of these
5. The hexadecimal system uses base (a) 2; (b) 4; (c) 8; (d) 16; (e) 60
6. $11_2 + 1011_2$ = (a) 0011_2; (b) 0100_2; (c) 1000_2; (d) 0111_2; (e) 1110_2
7. $1110_2 + 101_2$ = (a) 10011_2; (b) 11010_2; (c) 11001_2; (d) 10111_2; (e) 11011_2
8. Place values are used in (a) memory; (b) software design; (c) decimal numbers only; (d) binary, decimal, and hexadecimal numerals; (e) assembler programs
9. In the binary-to-decimal conversion presented, there is one step for each (a) value converted; (b) decimal digit; (c) binary digit; (d) remainder; (e) carry
10. In the decimal-to-binary conversion presented, there is one step for each (a) value converted; (b) decimal digit; (c) binary digit; (d) remainder; (e) carry
11. $1001_2 + 0011_2$ = (a) 1011_2; (b) 1100_2; (c) 0101_2; (d) 1000_2; (e) 0011_2
12. Negative binary values are stored using (a) overflow; (b) accumulator; (c) carry; (d) borrow; (e) a sign bit
13. When a computation produces a value greater than the hardware's capacity to store it, what is the result? (a) cacheing; (b) overflow and loss of data; (c) program termination; (d) complementing; (e) compression
14. Floating-point representation adds ___ to what is found in integer storage. (a) a sign bit; (b) an exponent; (c) a binary point; (d) flotation; (e) speed
15. An example of analog representation is (a) a file stored on a computer; (b) a message sent on the Internet; (c) the sound heard from an IPod; (d) a picture in RAM; (e) a register in a processor
16. A bit's value (a) is 0 to 255; (b) is 0 or 1; (c) fills a register; (d) fills a memory cell; (e) corresponds to a color pixel
17. Of the following, the smallest is: (a) bit; (b) kilobyte; (c) megabyte; (d) byte; (e) word
18. All data is communicated and stored by computers in what form? (a) analog; (b) digital; (c) megabyte; (d) packet; (e) other

2. Logic gates

1. A NOT gate is a(n) (a) software building block; (b) hardware building block; (c) design tool; (d) algorithm; (e) Java operator
2. The OR gate (a) is a peripheral; (b) contains a register; (c) yields a 0 if both its inputs are 1; (d) yields a 0 unless both its inputs are 1; (e) produces a 1 if either of its inputs is 1
3. The AND gate (a) is a peripheral; (b) contains a register; (c) yields a 0 if both its inputs are 1; (d) yields a 0 unless both its inputs are 1; (e) produces a 1 if either of its inputs is 1
4. A one-input circuit that outputs a 1 on input of 0 and a 0 on input of 1 is (a) NOT; (b) OR; (c) AND; (d) MAYBE; (e) XOR

3. A model computer

1. Which is *not* a peripheral? (a) the microprocessor; (b) the monitor; (c) the keyboard; (d) the printer; (e) a disk drive
2. Input/output is data that moves between RAM and (a) the program counter; (b) data registers; (c) the instruction register; (d) RAM; (e) peripherals
3. Electronic storage composed of silicon chips is (a) RAM; (b) CDROM; (c) hard disk; (d) keyboard; (e) monitor
4. Microprocessors have on them: (a) a disk; (b) a screen; (c) registers; (d) high-level code; (e) documentation
5. The characteristic feature of all general-purpose computers is that they (a) display colors; (b) have CDROMs; (c) can be upgraded; (d) can store programs; (e) run faster than 166 MHz.
6. (T-F) The fetch-execute cycle runs until the user inputs a Quit instruction.
7. A disk drive is a (a) software component; (b) register; (c) peripheral; (d) logic gate; (e) silicon-based storage device
8. (T-F) The operating system runs concurrently as applications execute.
9. Electronic storage composed of silicon chips is (a) RAM; (b) CDROM; (c) hard disk; (d) keyboard; (e) monitor
10. Microprocessors have on them: (a) a disk; (b) a screen; (c) registers; (d) high-level code; (e) documentation
11. (T-F) The language of a microprocessor is expressed in 0's and 1's.
12. Components of a CPU include (a) RAM; (b) control unit; (c) track; (d) packet; (e) software
13. The fastest-accessible of the following is: (a) RAM; (b) hard disk; (c) DVD; (d) register; (e) web site

14. What is fetched in the fetch-execute cycle?
    (a) instruction; (b) operand; (c) record; (d) byte; (e) file

4. Machine and assembler programs

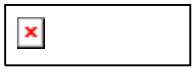
1. Which instruction below does *not* change the value in the accumulator? (a) *load*; (b) *store*; (c) *add*; (d) *sub*
2. A data statement in an assembler program introduces (a) a memory address by number; (b) a memory address using a name; (c) the accumulator; (d) the program counter; (e) a user-input value
3. (T-F) A microprocessor has its own special assembler language.
4. A language expressed in binary notation is (a) machine language; (b) assembler language; (c) Java; (d) all of the above; (e) none of the above
5. Assembler is (a) a language adapted both to the hardware and to human use; (b) an algorithm to convert numbers; (c) a high-level language; (d) a component of a microprocessor; (e) part of a network operating system
6. A mnemonic (a) specifies a data location; (b) specifies a data value; (c) specifies a constant; (d) specifies an action; (e) is a language
7. (T-F) The language of the microprocessor is assembler.
8. (T-F) At the processor level, a loop entails a backward jump.
9. Which type of language is closest to that used by a processor? (a) query; (b) formula; (c) markup; (d) assembler; (e) transfer-protocol
10. The code fragment,

```
          jump0  end
          print  sum
   end    stop
```

    implements a (a) sequence; (b) branch; (c) loop; (d) subroutine
11. Which is *not* a way to express an algorithm?
    (a) HTML; (b) assembler language; (c) flowchart; (d) pseudocode; (e) machine code

Short-answer

Objective 3: show knowledge of the binary system of representation

1. Convert to decimal: 1011_2
2. Convert to decimal: 0101_2
3. Convert to decimal: 0110_2
4. Convert to decimal: 1001_2
5. Convert to decimal: 1100_2
6. Convert to binary: 15_{10}
7. Convert to binary: 6_{10}
8. $1001_2 + 0011_2 =$
9. $0111_2 + 0100_2 =$
10. $1000_2 – 0110_2 =$
11. $0110_2 – 0010_2 =$
12. Convert 10 (decimal) to binary notation.
13. Convert the value 1001_2 to decimal notation.
14. Shifting left a binary representation of the number *x* by one bit produces what new value of *x*?
15. What is base-16 notation called?
16. Do each of the following number base conversions:
    (a) 23 = _____$_2$
    (b) 12 = _____ $_2$
    (c) 10110012 = _____ $_{10}$
    (d) 101_2 = _____ $_{10}$
    (e) 11011_2 = _____$_{10}$
    (f) 1010111_2 = _____$_{10}$
    (g) 35 = _____$_2$
    (h) 77 = _____$_2$
17. Using the binary storage format outlined in Section 1.5, show how each of the following decimal values would be stored in memory.
    (a) 375
    (b) –1270 (c) –1 (d) 32,127
18. What decimal values are represented by the following stored binary integer values?
    (a) 1000000000011101
    (b) 0100000011111000
    (c) 1111111111111111
    (d) 0111111100000000
19. What range of values can be stored in 16 bits? __
20. Do these conversions:
    (a) 6 = _____$_2$ (b) 100_2 = _____$_{10}$
    (c) 13 = _____$_2$ (d) 10101_2 = _____$_{10}$
21. Find the binary results:
    (a) $10_2 + 101_2$ = _____
    (b) $10111_2 + 1101_2$ = _____
    (c) $10100_2 – 1110_2$ = _____
22. Express each of these in exponential notation:
    (a) 0.47 _____
    (b) 392.1 _____
23. Write the truth table of the NAND gate:
    
24. Show the results of these bitwise operations:
    (a) $0101_2 \mid 1110_2$ = _____
    (b) $1010_2 >> 1$ = _____

Objective 4a: Recognize basic hardware components and their interactions

1. What is a low-level language, associated with a particular microprocessor, that uses binary code to represent instructions and data?
2. What information must a microprocessor have about a data item in RAM in order to assign it a value?
3. A circuit's output, given a certain combination of inputs, is diagrammed by a(n) _____
4. What is a data location that stores intermediate values while calculating?
5. What is a data location that stores the current instruction being executed?
6. What is a data location that stores the address of the next instruction that will execute?
7. What storage technique is used to store negative values?
8. What is the result when a computation produces a value that exceeds the hardware's storage capacity?
9. What is a low-level language that uses labels and mnemonics to express programs?
10. Name the loop that a microprocessor is running at all times.
11. Name the part of an assembler-language statement that specifies what action is to be taken.
12. Name the part of an assembler-language statement that specifies the location of the data to be acted upon.
13. Name the part of an assembler-language statement that specifies the name of the statement's address.
14. How many different values can be represented in 16 bits?
15. What information must a microprocessor have about a data item in RAM in order to access it?
16. Name four peripheral devices.
17. How much data is required to store the information as to whether or not a number is greater than zero?
18. What component of a microprocessor stores :
    (a) internal results of a computation?
    (b) the address of the next instruction to be executed?
    (c) the current instruction being executed?
19. What is the common name for primary storage?
20. If one of the sixteen bits of a memory cell were not used as a sign bit, what further restriction would be placed on the type of numbers a 16-bit computer could process?

21. In what language is the following written? Describe some of what it does as best you can.

```
          input a
          input b
          load  a
          sub   b
          jump- bgrtr
          print a
          jump  end
bgrtr     print b
end       stop
a         data  0
b         data  0
```

22. What component of a microprocessor stores
    (a) internal results of a computation?
    (b) address of the next instruction to be executed?
    (c) the current instruction being executed?

23.
```
          load      fee
          sub       discount
          store     bill
          stop
fee       data      100
discount  data      20
bill      data      0
```
    In the program above, what is the effect of the line
    a) `load fee`
    b) `sub discount`
    c) `store bill`
    d) `fee data 100`

24. What information must a machine-language program have about a data item in a computer in order to access it?
25. How much data is required to store the answer to a yes/no question? _____
26. Write statements in the assembler language of the model processor to compute $c \leftarrow a + b$, i.e., to assign the value of $(a + b)$ to c, if a, b, and c are labels for data statements.
27. Describe the fetch/execute cycle in one or two paragraphs.
28. In what ways do the human brain and a computer resemble each other? Differ?
29. Why is the binary number system appropriate for digital computers?
30. What is the advantage of using a special twos-complement format for negative numbers instead of normal binary format?
31. Label each term below with the letter of its appropriate definition.
    ____ accumulator
    ____ assembly language
    ____ bit
    ____ instruction register
    ____ label
    ____ machine language
    ____ microprocessor
    ____ operating system
    ____ mnemonic
    ____ operand
    ____ program counter
    ____ RAM
    ____ software
    a) Computer programs.
    b) Component of a software system
    c) Hardware that contains logic for controlling program execution and manipulating data.
    d) A binary digit, having a value of either 0 or 1.
    e) Electronic data storage that uses silicon chips.
    f) Register that holds the program instruction currently being executed.
    g) The register in the processor that at any instant specifies the location in memory of the next instruction to be executed.
    h) In the processor, the register that stores either data copied from a memory cell or the result of an operation on that data.
    i) Software that manages a computer's activities while other programs run, providing services to programs and to the user.
    j) Has one mnemonic for each machine instruction of a processor.
    k) Set of binary-coded instructions for a particular microprocessor
    l) A data item to be operated on by an instruction
    m) A machine instruction expressed in easily remembered word form.
    n) A name chosen by the programmer for an address in memory.

Longer answer

Objective 4 b. Write and trace an assembler program

1. Write a program for the model processor's assembly language that accepts two values and displays the smaller one.

Answers to hardware study questions

1. The binary system of numerals

2. e
3. a. 3 + 2 = 5
4. b
5. b. $3 = 1 \times 2 + 1 \times 1 = 11_2$
6. e
7. e
8. d
9. c
10. c
11. b
12. e
13. b
14. b
15. c
16. b
17. a
18. b

2. Logic gates

1. b. A NOT gate is a logic component built from transistors.
2. e. The OR gate yields a 0 only if both its inputs are 0.
3. d. The AND gate outputs 1 if both inputs are 1.
4. a

3. A model computer

1. a. The microprocessor is the central component of a computer system; the input/output devices are peripherals.
2. e
3. a. The language accepted by microprocessors, machine language, consists of 0's and 1's only.
4. c
5. d
6.
7. f. The fetch-execute cycle runs until a *stop* instruction is encountered.
8. c
9. t
10. a
11. c
12. t. A single bit may store the answer to a yes-or-no question, 0 or 1.
13. b
14. a

4. Machine and assembler programs

1. b. The *store* instruction copies the value in the accumulator to a memory location; the others change the value there.
2. b. A data statement begins with a label, which is a name for the memory location that stores a data value.
3. t. The assembler language of a microprocessor is a human-oriented way of expressing each machine-language instruction that the chip is built to execute.
4. a
5. a
6. d
7. f
8. t. To repeat an instruction the microprocessor must jump to an earlier program instruction.
9. d
10. b

Study questions on Topic 3: Java basics

1. Higher-level languages and software development

1. (T-F) The compiler reads machine code and outputs high-level code.
2. Java is (a) easier to understand than machine code; (b) the language of the executable program; (c) what the compiler generates; (d) a formatting language; (e) none of these
3. Machine language is (a) easier to understand than Java; (b) the language of the executable program; (c) the language of the source code; (d) what the compiler starts with; (e) none of these
4. (T-F) The compiler translates from byte code to Java.
5. A runtime package (a) is an add-on; (b) stays in the compiler; (c) is used mostly in debugging; (d) is part of every compiled program; (e) arrives in the mail of good programmers
6. (T-F) A machine-language instruction usually performs a more complex task than a Java statement.
7. A compiler produces (a) high-level code; (b) machine code or byte code; (c) documentation; (d) keyboard input
8. (T-F) Java is higher-level than assembler language.
9. (T-F) The Java Development Kit includes a compiler.
10. Byte code is like (a) Java; (b) ASCII code; (c) machine code; (d) HTML; (e) none of these

2. Java basics

1. Which is *not* a valid Java identifier: (a) *2days*; (b) *myvar*; (c) *Data12*; (d) *FooBar*; (e) *x*
2. (T-F) A data type is a memory location with space allocated for a value.
3. Which is the most important, for a program to be maintainable? (a) code has no syntax errors; (b) code generates no runtime errors; (c) code runs fast; (d) code is easy to understand; (e) none of these is important
4. (T-F) Variable identifiers stand for memory locations.
5. Variable declarations in a list following a type name are separated by (a) periods; (b) commas; (c) semicolons; (d) colons; (e) parentheses
6. The Java statement *x = 2;* corresponds to the processor-level operations (a) load 2 from memory, then store to location *x*; (b) store at *x*, then load from memory; (c) unconditional jump; (d) conditional jump backward; (e) conditional jump forward
7. (T-F) A variable is a memory location with a name and a data type.
8. (T-F) The compiler treats *int* the same as *INT*.
9. (T-F) An identifier may start with a digit.
10. Which is *not* a valid Java identifier: (a) *in1*; (b) *my_method*; (c) *3p2*; (d) *t4*
11. Which is *not* a valid Java identifier: (a) *xyz*; (b) *my variable*; (c) *x12345*; (d) *FooBar*
12. (T-F) *{}* is a valid statement in Java.
13. A variable does *not* have a (a) name; (b) address; (c) type; (d) precedence; (e) none of these
14. To document our source code we are encouraged to use (a) short variable names; (b) comments; (c) operators; (d) manuals
15. A global data item that might be of value in documenting code is a (a) named constant; (b) comment; (c) literal; (d) variable; (e) none of these
16. Java syntax requires that every program have: (a) a *main* method; (b) a variable; (c) input; (d) output; (e) comments
17. *Scanner* is (a) an input stream class; (b) an output stream class; (c) the assignment operator; (d) a relational operator
18. (T-F) Java programs use streams to handle input and output.
19. An instance of *Scanner* is (a) a class; (b) a keyword; (c) a data item; (d) an object; (e) none of these
20. (T-F) A package is a kind of program documentation.
21. (T-F) The compiler ignores white space
22. (T-F) Java has two kinds of statement: declarations and executable statements.

3. Syntax and semantics

1. (T-F) Semantics deals with structure, syntax deals with meaning.
2. Syntax is (a) documentation; (b) meaning; (c) grammar rules; (d) good-programming guidelines; (e) recursion
3. Which determines whether a program compiles: (a) code is easy to understand; (b) code has no syntax errors; (c) code runs fast; (d) code generates no runtime errors; (e) none of these

Answers to study questions on topic 3, Intro to Java

1.Higher-level languages and software development

1. F
2. A
3. B
4. F
5. D
6. F
7. B
8. T
9. T
10. C

2. Java basics

1. A
2. F
3. D
4. T
5. B
6. A
7. T
8. F
9. F
10. C
11. B
12. T
13. D
14. B
15. A
16. A
17. A
18. T
19. C
20. F
21. T
22. T

3. Syntax and semantics

1. F
2. C
3. B

Short answer

Objective 7: Recognize basic syntax rules of Java

1. Write a Java statement that declares an integer variable, *sum,* and initializes it with the sum of *x* and 2.
2. Write a Java statement that prompts for an integer variable, *quantity*, with appropriate labelling.
3. What is the last token in a compound statement?
4. What are
   (a) the set of grammar rules of a programming language? _____
   (b) the meanings embodied in a programming language? _____
5. Name a Java grammar rule that is recursive.
6. What is wrong with this definition of a named constant?
```
final int YEAR;
```
7. Write a Java statement that declares a variable that will store the number of people in a team.
8. Which of the items below are valid Java identifiers?
   (a) *A12*; (b) *cat*; (c) *39G*; (d) *out.put*;
   (e) *A3D*; (f) *employee-salary*;
   (g) *employee_salary*
9. Will these two lines of source code compile?
```
out.println( "This is a long sentence
continued on a second line";
```
   If not, what message is generated? Correct the error.
10. Declare a named constant to represent the number of days in a week.
11. How does a variable differ from a literal?

Other

1. What may the programmer do in a Java program to make sure his or her intention is clear?
2. Name three ways to give a value to a variable in Java.
3. Write a Java-language statement that displays the label "The quantity is" and the value of the integer variable, *quantity.*
4. Give two reasons for using named constants in a program.
5. How is Java like English? How is it different?
6. What should every program have that the compiler will ignore?
7. Write a Java statement to prompt the user for a quantity of items.
8. What is the output of this code?
```
int x1,x2 = 5;
out.println(3 * x1);
```
9. Write three Java statements to declare, prompt for, and input an integer variable that expresses the number of years in the term of a contract.
10. Name two different forms of documentation that you can write in your source code to make the program easier to understand.
11. Label each term below with the letter of its appropriate definition:

_____ assignment statement

_____ compiler

_____ compound statement

_____ documentation

_____ fourth-generation language

_____ method definition

_____ low-level language

_____ keyword

_____ identifier

_____ named constant

_____ stream

_____ third-generation language

_____ syntax

_____ statement

_____ source code

_____ variable declaration

_____ white space

a) a procedural vehicle for programming that expresses the programmer's intention at a higher level than the hardware
b) a way to write a program such that one statement appears for each machine-level operation
c) a high-level way of expressing programs, such that the programmer may concentrate on what is to be done, not how to do it
d) a program that translates from a higher-level language to a lower-level one
e) programs as written in a higher-level language
f) program code that spells out what *main*, for example, does, in full detail, including the name of the method
g) a series of characters going somewhere
h) a single line of a program
i) allocation of RAM to a data item that may be given values later
j) a vocabulary element reserved for special uses in a language
k) the name for a data item or other program element
l) program code that copies a data value to a variable location

m) a data item whose value may not be changed
n) grammar rules for a language
o) series of statements that appear in braces
p) program formatting that is ignored by the compiler
q) user prompts, comments, and appropriate naming of data items

Longer answer:

Objective 5: Use a programming environment to compile and test a Java program

Objective 6: Code screen output and keyboard input in Java

Objective 7: Recognize basic syntax rules of Java

Objective 8: Write appropriate comments to document code

Write a program, with documentation to:

1. Display the words "This is a program" on four lines
2. Initialize and display four integer variables, with values 1, 2, 3, 4
3. As #2, but label output with variable names
4. Input and display four integers
5. Display
   x
   xx
   xxx
   in *one* statement.
6. Declare named constants for tax as a percentage (5) and discount (33). Display amount due on *4* inputs, consisting of a price and quantity for each of two items. Use no fractions, use integer division if necessary.
7. Write a well-documented Java program that prompts for the unit price of an item, and the quantity of items purchased, and displays the total price to be charged.
8. Write a Java program that accepts three weights and displays their integer average.
9. Write a well-documented Java program that prompts for three quantities and displays their sum.
10. Write a Java program that declares a named constant to represent the number of weeks in a year, prompts for and inputs the number of weeks a person will go on vacation, and displays the number of work weeks in the year. (There are 52 weeks in a year.)
11. Consider the program:

```
// add.cpp
// Displays sum of 2 input
// inegers.
#include <iostream.h>

void main()
{
   out.println( "Enter 2 #s"
      + "to add: ");
   int input1,input2,sum;
   input1 = in.nextInt();
   input2 = in.nextInt();
   sum = input1 + input2;
   out.println(input1 + " + "
        + input2 + " = " + sum);
}
```

    If, in running this program, 20000 is input for A and 30000 is input for B, what output is generated by the last statement? Can you explain how this output value was generated? Does the discussion of integer storage and twos complements in Chapter 1 offer any hints?
12. Write, compile, and run a Java program that calculates and shows the integer *y* for any integer input value *x*, where $y = 3x^2 - 2x + 5$. The program should send output to a disk file.
13. Will the two-line statement:

```
out.println( "This continues
on a second line";
```

    compile? If not, what message is generated? Correct the error and explain the general rule that applies.
14. Debug this code:

```
int quantity,price;
out.println( "Enter quantity "
   + "and price: ");
quantity = in.nextInt();
price = in.nextInt();
out.println("Amount due: "
   + quantity * price);
```

15. Write a well-documented Java program that declares a named constant to represent the number of days in October, prompts for and inputs the number of weekend days that month, and the number of holidays, and displays the number of remaining, *work* days in the month.

Answers to short-answer questions on T3

1. int sum = x + 5;
2. out.println(“Enter quantity: “;
   Write comments or use meaningful identifiers
3. 1. Initialize in declaration
   2. Assign with assignment operator (=)
   3. Input with *cin* or *scanf*
4. ```
   out.println( “The quantity is “
      + quantity +  ;
   printf(“The quantity is %d\n”,
      quantity);
   ```
5. (1) To document meaning of a value used; (2) To permit updating in one step a constant used multiple times
6. A variable has an address and its value can change.
7. *Like:* Has grammar rules and vocabulary
   Different: English has more rules, is ambiguous
8. Comments, white space
9. ```
   Int quantity;
   out.println( “Quantity: “;
   cin >> quantity;
   ```
1.
2.
3.
4.
5.
6.
7. ```
   out.println( “Number of years: “;
   int num_years;
   cin >> num_years;
   ```
   ```

Study questions on Topic 4: standard types

1. Java classes, objects, and methods

1. A class is an abstract specification for (a) a program; (b) an algorithm; (c) arrays; (d) objects; (e) none of these
2. Members of classes include (a) comments and specifications; (b) algorithms and control structures; (c) operations and properties; (d) numbers and specifications; (e) none of these
3. I/O features are made available to Java programs through (a) the Internet; (b) cables; (c) packages; (d) packets; (e) none of these
4. *System* is a predefined Java (a) object; (b) class; (c) file; (d) package; (e) none of these
5. *Scanner* is a predefined Java (a) object; (b) class; (c) file; (d) package; (e) none of these

2. Numeric data types

1. (T-F) In the absence of parentheses, the compiler generates code that performs multiplication before addition.
2. * is (a) a sign; (b) a string; (c) a method; (d) an operator; (e) a call
3. In an assignment like *a = x*, data is (a) just copied; (b) copied and deleted; (c) discarded; (d) crunched; (e) displayed
4. Before outputting a variable, it should always be (a) given a value; (b) input; (c) processed; (d) used in an expression ‘
5. Before a variable is used in a statement, it must be (a) assigned a value; (b) declared; (c) output; (d) evaluated
6. 1+2*4 = (a) 0; (b) 1; (c) 8; (d) 9; (e) 12
7. In Java, 15/12 = (a) 0; (b) 1; (c) 1.25; (d) 12; (e) 15
8. 15 % 12 = (a) 0; (b) 1; (c) 2; (d) 3; (e) 12
9. 1998 % 100 = (a) 19; (b) 19.98; (c) 98; (d) 199800
10. 137 % 10 = (a) 13; (b) 13.7; (c) 7; (d) 1370
11. Convert to Java:

    b + c
    d + e

    (a) a; (b) abcde; (c) (b+c)/(d+e); (d) b+c/d+e; (e) b/d+c/e
12. A category that defines the meaning or intepretation of a pattern of bits is (a) an algorithm; (b) a register; (c) a method; (d) a data type; (e) a library
13. % is the _____ operator. (a) int; (b) initialization; (c) modulo; (d) division; (e) insertion
14. (T-F) Unary operators have two operands.
15. Which operator could help us shorten a statement like total = total + price;? (a) +; (b) =; (c) ++; (d) +=; (e) %
16. Which operator could help shorten a statement like total = total + 1;? (a) +; (b) =; (c) ++; (d) *=; (e) ~
17. If *n* equals 6, then (++*n*) equals (a) 5; (b) 6; (c) 7; (d) 12; (e) 1
18. If *n* equals 3, then (n++) equals (a) 1; (b) 2; (c) 3; (d) 4; (e) 6
19. What is the smallest capacity numeric data type, of the following? (a) *char*; (b) *int*; (c) *short*; (d) *long*; (e) *float*
20. What is the largest capacity data type, of the following? (a) short; (b) *int*; (c) *float*; (d) *double*; (e) *byte*
21. Which type stores its data in three components? (a) *int*; (b) *double*; (c) *short*; (d) *char*; (e) *long*
22. (T-F) A data item of type *int* can be positive or negative.
23. (T-F) A data item of type *unsigned* can be positive or negative.
24. (T-F) A value of type *float* can be negative.
25. (T-F) A value of type *double* can be between 0 and 1.
26. (T-F) A value of type *int* could be between 0.1 and 0.9.
27. (T-F) Floating-point storage may entail representational error.
28. A mantissa, or fraction, component appears in type (a) *double*; (b) *int*; (c) *char*; (d) *byte*; (e) *String*
29. (T-F) Floating-point storage uses an organization concept similar to scientific notation.
30. Type casting in Java (a) is automatic; (b) is a way to produce a fractional value when dividing two integers; (c) uses the keyword *throw*; (d) is discouraged; (e) is a syntax error
31. What is the data type of 3.14? (a) *char*; (b) *int*; (c) *double*; (d) *char*[]; (e) *void*
32. What is the data type of 2.0? (a) *char*; (b) *int*; (c) *double*; (d) char[]; (e) *void*
33. To convert an *int* value to a *float* in Java (a) is possible with just an assignment; (b) requires method calls; (c) requires writing an algorithm; (d) is not possible
34. If PI is 3.1416, and *n* is declared as an integer, then after *n* = 2 * PI; outputting *n* yields (a) 0; (b) 3; (c) 6; (d) 6.2832; (e) a compiler error
35. (T-F) If *x* is an integer variable and *y* is a *double*, then "*x = y*;" will compile without error.
36. A Java statement that begins with a type name may be a (a) branch; (b) loop; (c) declaration; (d) compound statement; (e) method call

3. Character, string, and stream data

1. (T-F) An item of type *int* occupies fewer bits than a *char* item.
2. An item of Java type *char* occupies how many bits? (a) 0; (b) 1; (c) 4; (d) 6; (e) 16
3. Which is a primitive data type? (a) *String*; (b) any class; (c) array of *int*; (d) *char*; (e) none of these
4. Which is a structured data type? (a) *String*; (b) *byte*; (c) *int*; (d) *char*; (e) none of these
5. Files are _____ objects (a) *String*; (b) stream; (c) *int*; (d) *char*; (e) none of these
6. Streams are of length (a) 256; (b) fixed by user; (c) indeterminate; (d) finite; (e) none of these
7. What is the data type of the expression '3' when it appears in a program? (a) *char*; (b) *int*; (c) *float*; (d) *char[];* (e) *void*
8. What is the data type of 'n'? (a) char; (b) int; (c) float; (d) char[]; (e) void
9. What is the data type of '\\'? (a) *char*; (b) *int*; (c) *float*; (d) *String*; (e) *void*
10. *char* is (a) a keyword; (b) a user-defined identifier; (c) an expression; (d) a method; (e) an operator
11. (T-F) Every printable character has a numeric encoding from 0 to 127.
12. The newline character is (a) ‘n’; (b) ‘\n’; (c) ‘\t’; (d) ‘\0’; (e) ‘\r’
13. Character literals are expressed in (a) capital letters; (b) italics; (c) single quotes; (d) double quotes; (e) parentheses
14. (T-F) To find the ASCII code of a character, typecast the character to *int* and display it.
15. (T-F) An instance of class *String* can be meaningfully compared to another string using the == operator.
16. What operator concatenates Java strings? (a) +; (b) ~; (c) =; (d) cat; (e) none
17. A stream is a(n) (a) algorithm; (b) device; (c) data item; (d) method; (e) simple data type
18. How is the newline character expressed in Java? (a) “0”; (b) ‘\n’; (c) ‘\0’; (d) ; (e) none of these.
19. (T-F) I/O takes the form of streams.
20. For input in Java, we use (a) the assignment operator; (b) data statements; (c) *read();* (d) *Scanner;* (e) *System.out*

4. Java applets

1. Code that runs on a browser’s Java virtual machine is in (a) web sites; (b) web services; (c) applets; (d) scripts; (e) none of these
2. Java byte code runs on (a) the command line; (b) the Java virtual machine; (c) the microprocessor; (d) a web site; (e) none of these

5. Java graphics

1. A Java program usually opens a window by (a) declaring a *JFrame* object; (b) calling the Windows API; (c) drawing a rectangle; (d) outputting a packet; (e) none of these
2. Java graphics uses (a) RGB color; (b) bitmap storage of lines; (c) inheritance; (d) exceptions; (e) none of these

Answers to multiple-choice and T/F questions on Topic 4

[*Correlation with questions to be verified*]

1. Java Classes

1. t1
2. d
3. c
4. c
5. b
6. b

2. Numeric data types

1. t
2. d
3. a
4. a
5. b
6. d
7. b
8. d
9. c
10. c
11. c
12. d
13. c
14. f
15. d
16. c
17. c
18. c
19. c
20. d
21. b
22. t
23. f
24. t
25. t
26. f
27. t
28. a
29. t
30. b
31. c
32. c
33. a
34. c
35. f
36. c

3. Character, string, and stream data

1. f
2. e
3. d
4. a
5. b
6. c
7. a
8. a
9. a
10. a
11. t
12. b
13. c
14. t
15. f
16. a
17. c
18. b
19. t
20. d

4. Java applets

1. c
2. b

5. Java graphics

1. a
2. c

Short-answer:

Objective 9: Declare numeric variables and assign them expressions as values

1. Write a statement in Java that will declare and assign the value 70.4 to a variable named, *height*.
2. Write an expression that converts the value 8.42 to *int* and stores the result in an *int* variable, *n*.
3. Write a declaration for a variable, *cost*, that can store a fractional value.
4. Write an expression that type casts the *double* variable *amt* to type *int*.
5. These declarations will be used to compute the area of a circle from an input radius value:
   float a,r;
   Rewrite the declarations, using meaningful identifierws
6. Write an appropriate constant declaration that would allow you to replace the numeral 0.05 (for sales tax) in the following Java statement with a meaningful identifier.
   tax_amt = 0.05 * sales_amt;

Objective 10: Manipulate characters and strings in Java

1. Write a declaration for a variable of the type most appropriate to store the value '9'.
2. Declare and initialize a variable, *letter*, to store the character value 'Q'.
3. Declare and initialize a variable, *letter*, to store the character value 'Q'.
4. Declare and initialize a variable, *team*, to store the string value "Rams".
5. Declare a character variable and initialize it with the null character.
6. Name a Java standard data type used to store sequences of characters.

Other

1. How are the newline, tab, and null character constants expressed in Java?
2. Name the data type of '4' when it appears in a program.
3. Name the data type of "4.5" when it appears in a program.
4. What is the character constant for the newline?
5. What is the name of the table that specifies the numeric encodings of characters?
6. What operation occurs when + or += are used with Java strings?
7. Name the standard header file that defines the Java class, *string*.
8. What is a stream?
9. What is a sequence of characters coming from or going to a device?
10. Name a data type for an object used to open and read a file.
11. Show the 8-bit binary value that is stored when a user types *R [Enter]* in response to the following Java statement, where *ch* is of type *char*: cin >> ch;
12. What is the data type of the parameters to *toupper* and *tolower*? What is the data type of the values these methods return?
13. How are the newline, tab, and null-character constants expressed in Java?
14. How can you convert an integer to a character value? A character to an integer?
15. What are an advantage and a disadvantage of Java 's automatic type conversion?
16. Write expressions that convert:
    a. "9" to *int*

    b. 30 to *float*

    c. "2.8" to *float*

    d. 0.9 to *String*

    e. 83 to *String*

7. What is the data type and memory allocation, in bits and bytes, of a single component of a string data item? How many bytes of memory are allocated by this declaration:
```
char prompt[] =
    "Enter name: ";
```
8. What is the data type and memory allocation, in bits and bytes, of a single component of a string data item?
9. Name the four standard identifiers associated with the Java console input and output statements. For each of the four identifiers, specify its library header file and whether it is a data item (D) or a method (F).
10. Label each term below with the letter of its appropriate definition:
    _____ *char*
    _____ ASCII
    _____ character string
    a) an 8-bit data type corresponding to the ASCII table.
    b) A standard table of characters and their decimal encodings.
    c) Storage of characters in consecutive RAM locations.
    d) Standard library file that supplies file input and output object types.=
11. Evaluate the Java expression, *1 + 7 * 4 + 15 / 3*.
12. Evaluate: 2 – 6 / 2 + 4
10. (a) What is the data type returned by the *next()* and *nextLine()* methods?
    (b) What differentiates these methods?

11. What is the output of this code?
```
int n = 25.9;
out.println( n + 1);
```
12. What is the relationship between a data type and a variable?
13. Name *two* data types whose instances can store fractional values.
14. Name *two* data types whose instances can store whole-number values.
15. List the the arithmetic operators in order of precedence.
16. Write an expression whose value is the remainder when 7 is divided by 2.
17. What Java package is used to access the methods *abs, sqrt* and *sin*?
18. Write a Java expression equivalent to $5x^3$
19. Write a shorter version of this statement: *quantity = quantity + input;*.
20. What operator assigns to its left operand the product of its left and right operands?
21. What operator assigns to its left operand the sum of its left and right operands?
22. What operator assigns to its left operand the difference between its left and right operands?
23. What operator assigns to its left operand the quotient of its left and right operands?
24. What operator assigns to its left operand the remainder when its left operand is divided by its right?
25. Write an expression that increments a variable *n* by 1 and whose value is the original value of the variable.
26. Write an expression that decrements a variable *n* by 1 and whose value is the original value of the variable.
27. Write an expression that increments a variable *n* by 1 and whose value is the incremented value.
28. Write an expression that decrements a variable *n* by 1 and whose value is the decremented value.
29. Name an integer data type that may store a larger range of values than an *int.*
30. Name an integer data type that may store a smaller ranger of values than an *int*.
31. Name a numeric data type that stores only values greater than or equal to 0.
32. What standard Java library defines the *sqrt* method?
33. What is the common term for coercion of data types?
34. When a value assigned to a variable is too large for the variable's storage capacity, it is a case of _____.
35. What is the output of:
```
int a = 12.8;
out.printl(a);
```
36. Wht is the output of:
```
int a = 9.6;
System.out.print(a);
```
37. List from smaller to larger (in storage space occupied) the following types: int; float; short; unsigned; double; char.
38. Write an appropriate constant declaration allowing you to replace the numeral 0.05 in the following Java statement with a meaningful identifier.
```
tax = 0.05 * sales_amt;
```
39. Will the following program compile? If not, what error message will you get?
```
void main()
{
    const int num_bldgs = 4;
    num_bldg = 5;
}
```
40. Evaluate:
sqrt (4)____
exp(0) ____
pow(3,2) ____
abs (–60)____
pow(2,3) ____
fabs(30.33) ____
abs (–20.33) ____
41. How many different floating-point numbers can be represented if 16 bits are allocated for an exponent and 31 bits are allocated for the normalized binary fraction? What range of values can be represented?
42. Is a real number a kind of rational number? Explain with reference to the Venn diagram in Section 3.2.
43. Why is it impossible to exactly represent the value (1/3) in binary floating-point notation? Why is it impossible to represent every square root exactly?
44. What are the data types of the Java constants '5', "5", 5, and 5.0, respectively?
45. What is wrong with these named-constant definitions?
(a) `final float LB_PER_OZ = 1/16;`
(b) `final TAX_RATE = 0.05;`
46. What is the value of *pounds* after these two lines execute? (Careful…)
```
int ounces = 8;
float pounds = ounces / 16;
```
47. Using the Java precedence rules, evaluate each of the following expressions:
(a) 12 / 4 – 2 _____
(b) 3 + 4 * 5 _____
(c) 7 % 3 + 1 _____
(d) 20 / (4 + 3 * –2) * 3 _____
48. Several people share the cost of a computer list-priced at $599.95; it is sold at a 10% discount and has a 5% sales tax.
(a) Write a Java expression that represents the price, in dollars and fractions of a dollar, of one person's share of a computer if it retails at $599.95, and is shared among the members of a club that has five members now and will have two more members when the computer is purchased.
(b) Write an expression using variables for price, discount, etc.

49. What are the outputs of each of these statements?
    (a) out.println(1 / 3; (b) out.println((8 / 2) / 3;
    (c) out.println((8 / 2.0) / 3; (d) out.println(26 % 4;
    (e) out.println(7.5 / 2.5; (f) out.println(2.5 * 4;
50. Write a Java statement that calculates and displays the corresponding integer *y*, for any integer input value *x*, applying the formula:
    $y = 3x^2 - 2x + 5$
51. Using type casting, write a statement that assigns the value 2/3 (the mathematical quotient of 2 and 3) to a *double* variable.
52. Label each term below with the letter of its appropriate definition:
    _____ *long*
    _____ floating-point storage
    _____ representational error
    _____ initialization
    _____ *float*
    _____ type conversion
    a) An integer data type
    b) A format for storing possibly fractional numbers
    c) Discrepancy between a floating-point numeral as stored, and the intended value.
    d) A way to give a value to a data item in the same statement that declares it.
    e) Data type for possibly fractional values.
    f) Automatic feature in Java that allows assigning value of one type to variable of another.
    g) Standard constant that generates new line of output.
    h) Standard library file that supplies output-formatting methods.
    i) Standard method that determines field width for output.
    j) Any of a set of standard methods that helps format floating-point output.

Answers (Short-answer):

1. 1 + (7 * 4) + (15 / 3) = 1 + 28 + 5 = 34
2. 25
3. *scanf("%f",&height);*
cin >> height;
4. *A variable is an instance of a data type.*
5. *int n = 8.42;*
float cost
float; double
int; char; long int; short int
/, %, *; +, –
7 % 2
math.h
6. 5 * *x* * *x* * *x*
7. *quantity += input;*.
8. *= assigns to its left operand the product of its left and right operands?
9. += assigns to its left operand the sum of its left and right operands?
10. –= assigns to its left operand the difference between its left and right operands?
11. /= assigns to its left operand the quotient of its left and right operands?
12. %= assigns to its left operand the remainder when its left operand is divided by its right?
13. *n*++ increments *n* by 1 and its value is the original value of *n*.
14. *n*– – decrements *n* by 1 and its value is the original value of *n*.
15. ++*n* increments *n* by 1 and its value is the incremented value.
16. – –*n* decrements *n* by 1 and its value is the decremented value.
17. Type *long* may store a larger range of values than an *int.*
18. Type *short* may store a smaller ranger of values than an *int*.
19. *unsigned int* stores only values greater than or equal to 0.
20. 2 – 6 / 2 + 4 = 3
21. The standard library *math.h* defines the *sqrt* method.
22. Type casting is the common term for coercion of data types?
23. *(int)amt* type casts the *double* variable *amt* to type *int*.
24. When a value assigned to a variable is too large for the variable's storage capacity, it is a case of overflow.
25. *12*
26. *9.6*
27. The standard library used in formatting numeric output is iomanip.h

Longer answer problems for Topic 4 (Standard types)

Objective 11: Read and write a sequential file

1. Write a program that displays the *second* line of a data file, named by the user in response to a prompt. Test it using the name of some program on your work disk; the result should be lines that beginning "//" and describe program examples.

Other

1. Write a program that prompts for a unit price, discount rate, and quantity and displays a subtotal without tax; a tax amount, given a constant tax rate of 5%; and a total amount due. Format to two places.
2. Write a program that prompts for the height and diameter of an oil drum and displays the area of its surface, given that the area of a circle is ($p \times radius^2$) and p = 3.14159. Format to two places.
3. Write a Java program that prompts for the length of piece of lumber, in feet, and displays that length in meters. A meter is 39.37 inches and a foot is 12 inches.
4. Write a program to input a salesperson's monthly sales total and base monthly salary, and compute her or his gross pay according to the formula that gross pay is the base salary plus five percent of monthly sales.
   Compute net pay according to the formula that net pay is eighty percent of gross. Print a monthly sales report for the salesperson as a table, using the following input/output guidelines, rounding off to the nearest cent.
   Sample Input/Output:
   ```
   Enter monthly sales total: 10000
   Enter base monthly salary: 2000.00
     Sales Base salary Gross pay  Net pay
   10000.00   2000.00   2500.00  2000.00
   ```
3. Write a program to compute the length of the diagonal of a square, prompting for the length of the edge of the square as input. The diagonal is the square root of the sum of the squares of two edges. Output a table, formatting numeric values to three decimal places.
   Sample Input/Output
   ```
   Enter edge: 1.0
      Edge   Diagonal
     1.000      1.414
   ```
4. Write a program to input a salesperson's employee ID, his or her monthly sales total (dollars and cents), and his or her base monthly salary (dollars and cents). Compute gross pay = base monthly salary + 5% of monthly sales total
   Compute net pay = 80% of gross pay. Display a monthly salary report for the salesperson, using the following input/output format:
   ```
   Enter salesperson's ID: 8192
   Enter monthly sales total: 5560.75
   Enter base monthly salary: 1500
   sales    sales                base              gross                net
   person
   81925560.75 1500.00     1778.04           1422.43
   ```
5. Write a program to compute and print the *y*-coordinate (*y*) of a point on the straight line defined by the formula $y = mx + b$, where *m* (slope), *b* (vertical offset), and *x* (the *x*-coordinate of the same point) are *float* values input from the keyboard. Format the computed value of *y* correct to two decimal places.
   Sample I/O:
   ```
   Enter M: 3
   Enter: -4
   Enter X: 2.75
   ```
6. Write a program that computes and prints the average *speed* (miles per hour) and the gas *mileage* (miles per gallon) for an automobile trip. The *distance* (miles), *time* (of trip in hours), and *gallons* (of gas used) are floating-point numbers input from the keyboard. Display computed output correct to 1 decimal place in a display field of 10 columns.
   Sample I/O:
   ```
   Distance: 200
   Gallons: 10.5
   Time: 4
   Average speed: 50.0 MPH
   Gas mileage: 19.0 MPG
   ```
7. Write a program to compute the maximum trip distance (in miles) possible in a car that has a gas tank capacity (in gallons) and which averages *MPG* miles per gallon of gas on trips. Capacity and MPG are *float* data input from the keyboard. Display the computed distance truncated to an integer number of miles.
   Sample I/O:
   ```
   Average miles per gallon: 23.7
   Tank size (gallons): 15.0
   Maximum trip distance without refueling: 355 miles
   ```

53. Write a program that accepts input of an integer invoice number, an integer quantity, and a real number unit price; computes a total price (quantity multiplied by unit price); and displays a simple invoice as shown below.
    Sample Screen I/O:
    ```
    Enter invoice number: 23001
    Enter quantity ordered: 53
    Enter unit price: 27.95
    Invoice # 23001
    ```

Quantity Unit price Total price
53 27.95 1481.35

8. Write a program to compute the area of a circle if the radius is input from the keyboard. Use the *M_PI* constant. The relationship between the radius and area of a circle is area = Pi * radius2. Display the area correct to 1 decimal place.
   Sample I/O:
   Enter radius: 10.2
   Area: 326.9
54. Write a program that computes the radius of the circle whose area is input from the keyboard. Use the *M_PI* constant and *sqrt* method. Display the radius correct to 2 decimal places.
55. Write a program that converts an input integer number of pounds plus an input integer number of ounces into a *float* number of kilograms. There are 16 ounces in a pound. One pound = 0.453592 kilogram. Display kilograms correct to 2 decimal places.
56. Write a program to compute the length of the hypotenuse of a right triangle with two 45-degree angles, if the length of one of the equal legs is input. Use the *sin* method for one of the equal angles; the *sin* method returns the ratio of the length of the leg opposite the angle to the length of the hypotenuse. The (angle) argument for the *sin* method must be expressed in radians. One degree equals (Pi / 180) radians.
57. Find the base-2 logarithm of any input value, using the *log* method and the fact that for any value *A*, $\log_B(n) = \log_B(A) * \log_A(n)$. It may help you to know that the Java *log* method returns $\log_e$ of its argument, where *e* is approximately 2.71828.
58. Write a program that tells how many different passwords can be formed using eight uppercase letters of the alphabet. Assume that letters may be repeated, so that, for example, "reindeer" is a valid password. Do you trust the computed result? Do the assignments
    P = 26 * 26 * 26 * 26 * 26 * 26 * 26 * 26
    and
    P = 26.0 * 26.0 * 26.0 * 26 .0 * 26.0
    * 26.0 * 26.0 * 26.0
    where P is a declared *float* variable, give the same result? If not, which is correct? Can you explain?
59. Write a program that prompts for three locations on a set of coordinate axes. Each point should have an *x* and a *y* value. Assume that the three points form a rectangle and display the lengths of its three sides. Use a comment or screen message to document the constraints on input data needed to avoid misleading output, such as a side of length 0.
60. Display these values, accurate to three decimal places: (a) the square root of two; (b) Pi; (c) the sine of a 45 degree angle; (d) the cosine of a 60 degree angle; (e) *e*, the base of the natural logarithm
9. Debug this code, whose output is 0 on input of 2 lb., 3 oz.:

```
const KILOS_PER_OUNCE =
 0.0283495;
out.println( "Enter pounds"
"and ounces: ";
float pounds, ounces;
cin >> pounds >> ounces;
float total_ounces =
16 * pounds + ounces;
float kilos =
 total_ounces *
KILOS_PER_OUNCE;
out.println( "Kilograms: "
+ kilos +  ;
```

10. Show, through compiler error messages or successful compilations, what the syntax rules of Java are regarding the order in which the type qualifiers *const*, *unsigned, short, long,* and *double* may appear with the types *int* and *float*.
11. Write a Java program that finds the area of a square whose side is twice the height of the user, in inches. Prompt the user for that height.
12. Write a program that displays the hour it will be, *duration* hours after *start* o'clock, given input of both integers. The program can be written using only three statements and only those Java elements presented in this chapter. On input of 11 and 1, output should be 12; on input of 6 and 7, output should be 1; on input of 2 and 24, output should be 2; on input of 12 and 120, output should be 12.
13. What is the last (rightmost) digit in the binary numeral expressing 2^{10001}? (This may be done without a program.) Write a program to output the rightmost *decimal* digit of the number 2^{10001}. The following table may help you:

n	1	2	3	4	5	6	7	8
2^n	2	4	8	16	32	64	128	256

14. Write a Java program that prompts for the costs of two items for sale, and displays the difference, accurate to two decimal digits.
    Sample Input/Output:

```
Enter two costs: 399.95 240
399.95 – 240.00 = 159.95
```

15. Prompt for integers *height* and *width* and draw a rectangle of X's in those dimensions.
61. Repeatedly prompt for integers until the user enters 0; display the largest.
2. Input a series of numbers, exiting on 0. Display length of the longest nondecreasing sequence in the series
3. *Challenge:* Using one integer variable, display dates in a calendar for a January that starts on a Monday, formatted as a grid.
4. Write a program that accepts two floating-point numbers and displays the larger one.

62. One ASCII character produces nonvisual output. Write a program that outputs *char (7)* and tell what its result is.
63. Write a Java program that uses the *char* data type name in a type cast to produce the following screen output. Do not use any character or string literals.
    ```
    This is
    a piece
    of cake
    ```
64. Write a program that will print out a five-character ASCII string generated by five integer input values.
65. Write a program that uses a type cast to convert integers to characters and that directly displays an input *int* variable as an ASCII code.
66. Find out experimentally what the return values of the methods *isalpha* and *isdigit* are for character parameters ‘e’, ‘Q’, ‘&’, and ‘4’.
67. Debug this code:
    ```
    ofstream outfile(“x.out”)
    int x,y;
    out.println( “x=” + x + “ y= “ + y;
    ```
68. Show your initials by assigning ASCII values to character elements of a string variable one at a time and displaying the string.

Study questions on Topic 5: Branches and loops

1. The if branch statement

1. What Java token always follows the *if* keyword? (a) statement; (b) expression; (c) left parenthesis; (d) identifier; (e) then
2. (T-F) The keyword *else* may begin a statement in Java
3. The *if* statement implements which control structure? (a) sequence; (b) branch; (c) loop; (d) no control structure; (e) method call
4. (T-F) A variable is accessible from any statement that appears after the variable's declaration in the same block.
5. (T-F) A block is enclosed by braces.
6. (T-F) A block defines a scope of access
7. (T-F) A block is enclosed in parentheses.
8. A typical Java branch statement contains (a) *if*; (b) *while*; (c) *do*; (d) a type name; (e) an assignment

2. Relational and Boolean expressions

1. > is (a) an assignment operator; (b) an insertion operator; (c) an arithmetic operator; (d) a relational operator; (e) none of these
2. To say "or" in Java, we write; (a) !; (b) &&; (c) ||; (d) ==; (e) "oar"
3. ! represents (a) disjunction; (b) conjunction; (c) logical negation; (d) abstraction; (e) encapsulation
4. Evaluate *a* > *b* && *b* == *c*, where *a* = 2, *b* = 1, and *c* = 3? (a) *a*; (b) *false*; (c) *true*; (d) 2; (e) none of these
5. Evaluate *a* != *b* && *a* < *c*, where *a* = 2, *b* = 1, and *c* = 3? (a) *a*; (b) *false*; (c) *true*; (d) 2; (e) none of these
6. && is (a) a logical operator; (b) a relational operator; (c) an arithmetic operator; (d) a Boolean expression; (e) none of these
7. (T-F) (3 > 2 || 1 == 0)
8. (T-F) 4 > 1 || 2 != 3
9. The relational operator that tests for equality is (a) =; (b) >; (c) ==; (d) !=; (e) <>
10. The relational operator that tests for inequality is (a) =; (b) >; (c) ==; (d) !=; (e) <>
11. What is the value of *a* == *b* || *b* > *c*, where *a* = 1, *b* = 2, and *c* = 3? (a) *a*; (b) *false*; (c) true; (d) 2; (e) none of these
12. || is (a) a logical operator; (b) a relational operator; (c) an arithmetic operator; (d) a Boolean expression; (e) none of these
13. != is (a) a logical operator; (b) a relational operator; (c) an arithmetic operator; (d) a Boolean expression; (e) none of these
14. (T-F) = is a relational operator.
15. (T-F) > is a logical operator
16. It is normally reasonable to compare items of type _____ for equality usijng ==. (a) *String*; (b) *int*; (c) *float*; (d) *void*; (e) none of these
17. Which is *not* a relational operator? (a) >; (b) <; (c) >=; (d) ==; (e) =
18. Which is *not* a logical operator? (a) !; (b) ||; (c) <; (d) &&; (e) none is a logical operator
19. A Java type with just two possible values is (a) *int*; (b) *double*; (c) *boolean*; (d) *char*; (e) *String*

3. Multiway branches: switch

1. (T-F) The *switch* statement is a way to avoid multiple nested *if*s.
2. Which is *not* a kind of statement? (a) assignment; (b) declaration; (c) input; (d) branch; (e) case label
3. The *break* statement is used with almost all (a) *if* statements; (b) loops; (c) *switch* statements; (d) methods; (e) none of these
4. The *case* keyword is used in which statement? (a) *while*; (b) method call; (c) assignment; (d) *if*; (e) *switch*
5. (T-F) The *switch* keyword implements the loop control structure.
6. (T-F) The *switch* statement is used for multiway branches.
7. (T-F) An alternative within a *switch* statement may have multiple case labels.

4. The loop control structure

1. Counters are used (a) in all loop statements; (b) in all well-written loop statements; (c) normally in *for* statements; (d) only in *for* statements
2. ++ is (a) a binary operator; (b) the increment operator; (c) the decrement operator; (d) an operator that has no effect on its operand; (e) a syntax error
3. An infinite loop (a) is the goal of every programmer; (b) is generally a logic error; (c) is very rare; (d) can be fixed by inserting a semicolon before the loop body; (e) none of these
4. (T-F) In a bottom-tested loop the body is always executed at least once.
5. Which is *not* favored in the world of structured programming? (a) the middle-tested loop; (b) the top-tested loop; (c) the bottom-tested loop; (d) simple branches; (e) nested branches
6. The *break* statement terminates (a) a program; (b) a method; (c) the current loop or branch; (d) all loops or branches; (e) none of these

5. Java loop statements

1. What appears after the *do* keyword is (a) a Boolean expression; (b) always a compound statement; (c) a statement; (d) a parenthesis
2. The *for* statement implements (a) a counted loop; (b) a top-tested loop; (c) a multi-way branch; (d) a recursive method
3. In a *for* statement the parentheses after *for* contain, in order, (a) an exit condition, then an update, then an initialization; (b) an update, an exit condition, and an initialization; (c) an initialization, an exit condition, and an update; (d) an initialization, an update, and an exit condition; (e) any of the above
4. (T-F) It is possible to use any of the three Java loop statements to solve a given problem that involves repetition.
5. The most appropriate statement for reading data from a file is (a) *if*; (b) *while*; (c) *do*; (d) *switch*; (e) assignment
6. The *while* statement is (a) counter driven; (b) bottom tested; (c) recursive; (d) top tested; (e) to be avoided
7. A Java statement that starts with *while* is (a) an assignment; (b) a compound statement; (c) a method call; (d) a loop; (e) a branch

6. Writing correct loops

1. Termination of a loop can be assured by ___ of an exit-test value (a) zeroing; (b) increase; (c) convergence; (d) decrease; (e) none of these
2. For a loop to exit, the body must _____ a value that is tested by the exit condition (a) initialize; (b) declare; (c) make invariant; (d) change; (e) none of these
3. Potentially infinite loops are consistent with good coding of (a) algorithms; (b) graphics; (c) interaction; (d) solutions to hard problems; (e) none of these
4. ___ -tested loops are to be avoided according to software-engineering practice (a) top; (b) middle; (c) bottom; (d) user; (e) none of these
5. Loop invariants can help verify (a) correctness; (b) performance; (c) specification; (d) design methodology; (e) none of these
6. Debugging may be aided by (a) syntax checking; (b) specification review; (c) tracing loops; (d) user training; (e) none of these

Short-answer questions

Objective 12: Identify declarations, loops, branches, and method calls and definitions in a Java program

1. The if branch statement

1. Name three Java statements that implement the decision control structure.
2. What statement in Java takes one of exactly two different courses of action depending on the result of a test of a value?
3. Write an expression in Java that tests whether the following is true: either *height* is greater than 72 or *age* is not less than 30
4. Write a statement that shows an error message if the value of numeric variable *quantity* is negative.
5. Write an expression in Java whose value is *true* if *height* is in the range of 60 to 72, otherwise *false*.
6. What is the output of the poorly indented code below?

```
int x = 1, y = 3;
if (x < 2)
if (y > 4)
out.println( “A”;
else
out.println( “B”;
out.println( “C”;
```

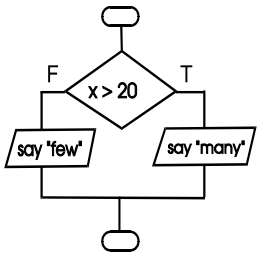
7. Name the syntax error in this statement:

```
if (X > Y)
    out.println(X)
else
    out.println(Y);
```

8. What does the following program output? Indent the nested *if* statements properly.

```
void main()
{
    int p = 1,q = 0;
    if (p) if (!q)
        out.println( "Y";
    else
        out.println( "N";
    else
    if (q)
        out.println( "Y";
    else
        out.println( "N";
}
```

9. A token is a program element, such as a variable, a keyword, an operator, or a punctuator, that cannot be broken down further into meaningful elements. What Java token always immediately follows the *if* keyword?
10. Why would a single set of test data be insufficient to thoroughly test a program containing an *if* statement?
11. What do indents mean in Java syntax?
12. What category of Java statements would correspond to the flowchart below if it were filled in?



13. What is the output of this code? Debug it if you find a logic error.

```
int n = 3;
if (n = 2)
    out.println( “2”);
else
    out.println( “not 2”);
```

14. What is the output of this code? Debug it if you find a logic error.

```
String name = new String();
out.println( “Name? “);
name = in.next();
if (name == “”)
    out.println( “Is blank”;
```

2. Relational and Boolean expressions

15. What does | | mean in Java?
16. What are the Java logical operators discussed in this topic?
17. List the Java relational operators.
18. Write a binary addition table and a binary multiplication table. If *true* = 1 and *false* = 0, how do the binary addition and multiplication tables compare with the truth tables for the logical operators || and *&&*?
19. Prepare a truth table for a logical binary operator XOR (eXclusive OR) which returns true if exactly one of the Boolean operands A and B is true, and returns false otherwise. Do the same for a NAND (Not AND) operator that returns true if at least one operand is false.
20. What is the value of the Boolean expression

```
A || B || (!A && !B)
```

where *A* and *B* are Boolean variables? (Try all combinations of values for *A* and *B*.)

21. Is “a > b > c” a syntax error? What rule does it follow or violate?
22. Distinguish “=” from “==”.
23. What is wrong with this statement?

```
if (ltr == ‘a’ || ‘b’ || ‘c’)
    out.println( “Letter is early “
            + in the alphabet\n”;
```

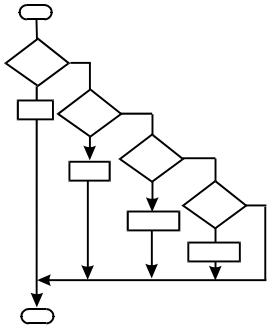
24. What is the output of this code? Debug it if you find a logic error.

```
if (x == 2)
```

```
        y = 1;
        out.println( “same”;
    else
        out.println( “different”;
```

3. Multiway branches: switch

1. What keyword is used to prevent falling through from one case label to the next in a *switch* statement?
2. If you were prompting the user for a number in the range of 1 to 10, and displaying a different message in response to each different number, which Java statement would be most recommended?
3. What keyword is a case label for the condition where no other case label is matched?
4. What keyword is used once for almost every case label in a *switch* statement?
5. Is there a set of braces in every *if* statement? Every *switch*?
6. Write one statement that displays different messages, depending on the value of a character variable named *input:* “Hello” if *input* has the value ‘a’, “One moment” if ‘b’, and “Bye” for ‘c’.
7. What Java statement corresponds to the flowchart below?



1. When would you not use *break* in a *switch* statement?
2. Does *2 < 3 && 4 > 1* mean the same as *!(!(2 < 3) || !(4 >1))*? Explain.
3. Assume that *x, y*, and *z* are *float* variables, *is_valid* is a Boolean variable, *x* = 3.0, *y* = 4.0, *z* = 2.0, and *is_valid* = *false.* Assign a value of *true* or *false* to each of the following Boolean expressions.
   a. (x > z && y > z)
   b. (x + y / z) <= 3.5
   c. (z > x) || (z > y)
   d. ! is_valid
   e. (x < 1.0) || (x >= 3.0)
   f. (0.0 < x && x < 3.5)
   g. (x <= y && y <= Z)
   h. ! (is_valid ||
      ((y + z) >= (x – z)))
4. Debug this error-ridden code:

```
out.println( "Do you drink soda pop?";
char input;
input = in.nextChar();
switch (input)
{
   'y': out.println("Recycle bottle\n";
   'n': out.println( "Enjoy juice\n";
   default: out.println("'y' or 'n'
please\n";
}
```

5. Label each term below with the letter of its appropriate definition:

   _____ Boolean expression
   _____ branch
   _____ *break*
   _____ case label
   _____ conjunction
   _____ conditional statement
   _____ *default*
   _____ disjunction
   _____ enumerated type
   _____ logical operator
   _____ logical error
   _____ *typedef*
   _____ logical negation
   _____ nested *if*
   _____ relational operator
   _____ relational expression
   _____ *switch*

   a. The decision control structure
   b. Program element in which a comparison is made and a value of *true* or *false* is returned
   c. A branch statement inside another branch statement
   d. The code that is executed when the expression in parentheses after *if* is true.
   e. Program element used in any comparison
   f. Program element used for negation, conjunction, disjunction
   g. Any program element with a Boolean value
   h. Expression that is true if, and only if, both its sub-expressions are true
   i. Expression that is true if, and only if, at least one of its sub-expressions is true
   j. Expression whose value is true if its sub-expression is false and false if its sub-expression is true
   k. Declared set of values associated with a set of identifiers listed in the declaration
   l. Keyword used to declare one data type as equivalent to another
   m. Keyword that must be used in the Java multi-branch statement

n. Program element tested for a match with a selector value and associated with a series of statements that will execute when there is a match

o. Keyword used as a case label to cover the possibility that no other case label matches the selector

p. Statement that follows a case label and prevents falling through to the statements following the next case label

8. List and identify (a) the Java relational operators; (b) the logical operators.
9. Write *one* Java *expression* that is true in cases where both of two conditions hold: (i) *a* is at least 3; (ii) the value of *answer* is 'Y'. Assume that *a* has been declared as an *int* and *answer* as a *char*.
10. Write a statement that tests whether a variable, *quantity*, is over 10, and if so reduces the value of another variable, *price*, by 5% and displays the message "discounted"
11. Evaluate these Java expressions:
    ____ ! (2 > 1)
    ____ 3 < 2 || 9 > 5
    ____ 1 + 1 == 2 && 4 != 5

4. The loop control structure

1. Write a Java statement that displays the numbers from 100 to 200.
2. Name three Java loop statements.
3. How many asterisks would the code below display?

```
for (int i=0; i < 40; ++i)
    for (int j=0; j < 10; ++j)
            out.println( "*";
```

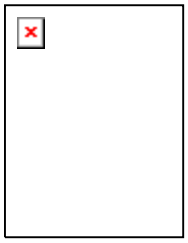
4. What is the output of this code?

```
for(int i=10; i < 16; ++i)
    if (i % 5 != 0)
            printf("*");
```

5. How many asterisks are output by this code?

```
for(int i=0; i < 40; ++i)
    out.print( "*");
for(int j=0; j < 10; ++j)
    out.print( "*");
```

6. Which Java keyword or keywords would be most convenient to implement the design pictured in the flowchart below?



7. What are the three items in parentheses just after the keyword *for* in a Java program?
8. What must a *while* loop contain if its body has more than one statement?
9. What is the result if a loop exit condition cannot be fulfilled?
10. Name the three Java loop statements, using keywords and brief (two-word) descriptions.
11. How many statements comprise the loop body in each Java loop statement?
12. What is the Java code in parentheses after *while* for?
13. What does the Java code after the *second* semicolon in parentheses in the *for* statement do?
14. Which Java statement would be most appropriate for getting user input until the user enters a blank string?
15. Rewrite the following statement as a *while* loop; as a *do...while* loop:

```
16.    for (int i=1; i < n; ++i)
       out.println( setw (i * i) + '*';
```

17. What would you do to ensure that a program that uses a loop to read file data will not fail due to the inability to open the file?
18. A simple *for* loop has a nesting level of 1; a *for* loop within a *for* loop has a nesting level of 2; a *for* loop within a *for* loop within a *for* loop has a nesting level of 3; and so on.
19. How many iterations do nested *for* loops execute if the nesting level is 3 and each nested loop's control variable has an initial value of 0 and a final value of 9?
20. What if the nesting level is *m* and each nested loop's control variable has an initial value of 0 and a final value of *n*–1?
21. What predefined Java method would you use to simulate a process that produces a sequence of unpredictable numbers, such as a list of winning lottery numbers?
22. What is the output of the following program code?

```
int x = 10;
while (x > 0)
    x = x – 3;
out.println( x );
```

23. What is the risk in using a *do...while* loop to read data items from a text file?
24. Construct a flowchart of the logic of the *for* loop.
25. The code below is intended to display the numbers from 1 to 10. (a) Will it compile? (b) Does it accomplish its objectives? If not, correct it.

```
int a = 1;
while (a < 10);
    a = a + 1;
out.println( a );
```

26. Debug this code, which compiles to an error message, "Multiple declarations of *m*":

```
for (int m = 0, m < 10, ++m)
    out.println(m);
```

27. Debug this code:

```
int num = 5,i;
for (i=0, i < num, ++num)
    out.println( i);
```

28. Here is a program fragment that was written to find the sum of all positive integers less than *n*, where *n* is input from the keyboard. Does it work? Explain your answer.

```
int n;
out.println( "Number? ";
n = in.nextInt();
if (n > 0)
{
    int sum = 0;
    int i = n – 1;
    while (i > 0)
    {
           sum = sum + i;
           out.println( "The sum up to “
            + n + " is " + s);
    }
}
```

29. What is wrong with the following *main* method, intended to compute 2^{10}?

```
void main()
{
    int power;
    while (power < 1000)
    {
        out.println( power +  ;
        power = power * 2;
    }
}
```

30. How many stars does the code below output? How many if the braces are removed?

```
for (int i=0, i < 5, ++i)
{
     out.println( ‘*’;
     for (int j=0; j < 10; ++j)
           out.println( ‘*’;
}
```

31. Name the three Java loop statements, using keywords and brief (two-word) descriptions.
32. What does the Java code after the second semicolon in parentheses in the *for* statement usually do?
33. Label each term below with the letter of its appropriate definition:

    ____ logic error
    ____ runtime error
    ____ specification error

    (a) Mistaken instructions to programmer concerning what a program should do
    (b) Bug
    (c) System message reporting division by zero, invalid operand to a method, etc.

34. Does the code below, when included in a program, generate a compiler error? If not, what does happen? Can you explain the observed results?

```
int a = 1;
while (a < 10)
   a = a + 1;
out.println( a );
```

35. Label each term below with the letter of its appropriate definition:

    _____ bottom tested
    _____ *break*
    _____ control variable
    _____ counted loop
    _____ *do...while*
    _____ end-of-file
    _____ file pointer
    _____ *for*
    _____ infinite loop
    _____ iteration
    _____ nested loop
    _____ *return*
    _____ top tested
    _____ *while*

    a) repetition
    b) loop that iterates a predetermined number of times
    c) top-tested Java loop statement
    d) loop without working exit condition
    e) state of affairs when all available characters have been read
    f) method used to tell when to exit file-reading loop
    g) system value used to keep track of where next access should occur
    h) loop whose exit condition is before body
    i) loop whose body is before exit condition
    j) bottom-tested C++ loop statement
    k) data item used to count a loop’s iterations or otherwise set exit condition
    l) unstructured statement enabling exit from loop but not method
    m) unstructured statement enabling exit from method
    n) loop within a loop
    o) Java keyword for counted loop

Longer answer problems on T5

Objective 13: Translate a flowchart into Java code

Objective 15: Debug a defective program

Branches

1. Write a program that accepts a distance in miles and displays "OK" if it is not higher than 100, otherwise "Too far."
2. Write a program that accepts three floating-point numbers and displays the smallest and largest. For full credit, show output verifying that your logic is correct for all cases.
3. Write a program to input two real numbers and to compute and display the absolute value of the difference between them. Use an *if* statement rather than a method to do this.
4. If a program performs a division operation with zero as the divisor, it will terminate with an error message. Write a program that prompts for two integers and displays their quotient, showing its own error message if the divisor is 0.
5. Write a program that sets a Boolean variable *is_factor* to true if the second of two input integers is a factor of the first. Otherwise *is_factor* is set to false.
6. Write a program that allows the user to input any two real numbers and then choose, using a *switch* statement, one of the four operations +, –, *, or /. The computed result, correct to a number of decimal places specified by the user, will be displayed. If the second input number is 0.0 and the operation "/" is chosen, the message "Can't divide by zero" should be displayed instead of a computed result.
7. Write a program to convert a letter of the alphabet to the corresponding telephone dial digit, based on the following conversion table:

Letter	*Dial digit*	
'A'..'C'	2	
'D'..'F'	3	
'G'..'I'	4	
'J'..'L'	5	
'M'..'O'	6	
'P','R','S'	7	{Q is not included on telephone dial}
'T'..'V'	8	
'W'..'Y'	9	{Z is not included on telephone dial}

8. If a non-convertible ASCII character is input, print an appropriate error message.
9. Write a program to classify a person on the basis of input height and weight. Use the following classification scheme:

Height	*Weight*	*Classification*
> 72 in.	> 190 lb.	Tall and heavy
> 72 in.	≤ 190 lb.	Tall and light
≤ 72 in.	> 170 lb.	Short, heavy
≤ 72 in.	≤ 170 lb.	Short, light

10. Write a program that prompts for an integer from 0 to 99 and accepts it into a string variable. Convert it to an integer variable without using *atoi* and display the integer variable's value. Display an error message if the input string is longer than two characters or if any character input is not a digit.
11. Debug the following program.

```
// weighbug.cpp
// Asks weight, categorizes
// user according to weight.
// HAS BUGS.
#include <iostream.h>

void main()
{
    out.println( "Your weight? ";
    int weight;
    cin >> weight;
    if (weight == 0)
      out.println( "Invalid ";
    else
      out.println( "You are ";
      switch(weight / 100)
        case 0:
             out.println( "light\n";
        case 1:
            out.println( "normal\n";
        case 2,3,4:
            out.println( "heavy\n";
}
```

12. Write a program to prompt for a letter of the alphabet and use a *switch* statement to convert it to the corresponding telephone dial digit, based on the following conversion table:

```
A B C  2
D E F  3
G H I  4
J K L  5
M N O  6
P R S  7
T U V  8
W X Y  9
```

    Be sure to validate input.
13. Write and test a program that prompts for six floating-point numbers and displays the largest. You will need to use only one relational operator, five times.
14. Write a program that accepts
    (a) three numbers and displays the largest;
    (b) four numbers; (c) five numbers

15. Write a program that prompts for two integers and displays their quotient; show an error message if the divisor is 0.
16. Write a program that allows the user to input any two real numbers and then choose one of the four operations +, –, *, or /. Display the appropriate computed result. Use a *switch* statement to select an operation.
17. Simplify this code:

```
char input;
input = in.nextChar();
switch(input)
{
   case '0':
      out.println( "digit"); break;
   case '1':
      out.println( "digit"); break;
   case '2':
      out.println( "digit"); break;
   case '3':
      out.println( "digit"); break;
   case '4':
      out.println( "digit"); break;
   case '5':
      out.println( "digit"); break;
   case '6':
      out.println( "digit"); break;
   case '7':
      out.println( "digit"); break;
   case '8':
      out.println( "digit"); break;
   case '9':
      out.println( "digit"); break;
```

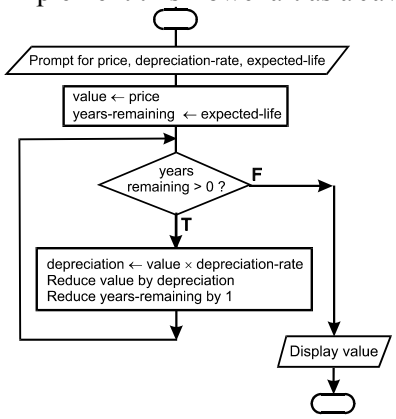
Loops

Objective 14: Solve a loop problem in Java

1. Write a program that repeatedly prompts for integer values and displays the largest one. Terminate the input on entry of a zero value.
2. Write a program that prompts for a loan amount for a prime-rate loan at 8.5%, and displays the amount to be repaid after 30 years, including principal and interest. Show results accurate to the cent. For numeric values use named constants. On input of a zero or negative amount, display an error message.
3. Write a program that repeatedly accepts a numerical score from 0 to 100 and displays a letter grade, according to the rule: 0-59 is 'F', 60-69 is 'D', 70-79 is 'C', 80-89 is 'B', and 90-100 is 'A'. Terminate the loop on input of a negative score and display an error message on invalid input.
4. Write a program that prompts for 100 numbers and displays the largest one.
5. Write a program that prompts for integers, until the user enters 0, and displays each one, except the zero.
6. Write a program that repeatedly prompts for integer values and displays the largest one. Terminate the input on entry of a zero value.
7. Write a program that displays all the numbers from 1 to 1000 that are divisible by both 5 and 3.
8. Write a program that repeatedly prompts for three ages and displays their average. Display an error message if invalid input is entered. Repeat the prompt-input-display process until the first input value is zero.
9. Write a program that prompts for two characters and displays all characters in the range from the first to the second. For example, input of 'A' and 'E' should yield output of "ABCDE".
10. Write a program that displays this figure once, *using a loop:*

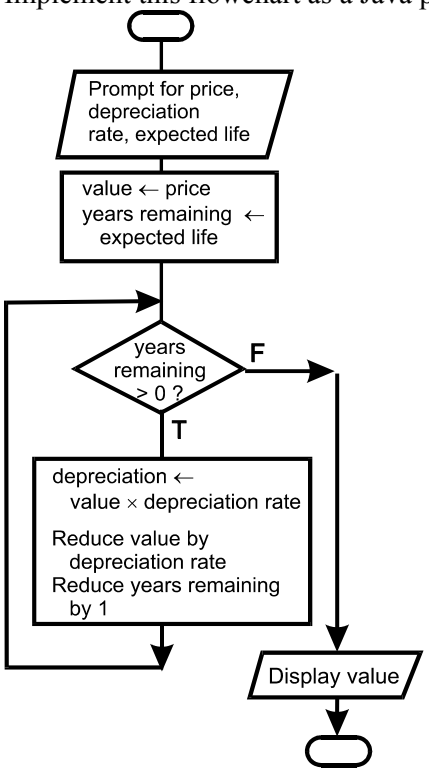
```
X
XX
XXX
XXXX
XXXXX
```

11. Write a program that repeatedly prompts for a character and displays the string "yes" if the input is the letter *Y*, "no" if *N*, and "maybe" if *M*, and an error message otherwise.
12. Write a program that repeatedly prompts for integer values and displays the largest one. Terminate the input on entry of a zero value.
13. Implement this flowchart as a Java program:



14. Write a program that displays a logarithm table, showing the natural logarithms of integer values from 1 to 100.
15. Write a program that displays the screen graphics characters and their ASCII codes. The graphics characters are those with values 128 to 255 in the ASCII table.
16. Input pairs of real numbers (e.g., *r1* and *r2*). Display the absolute value of the difference, *r1 – r2*, for each input pair. Exit from the loop if *r1* < 0.

17. Implement this flowchart as a Java program:



Sample output:

```
r1      r2          |r1 – r2|
5.00    16.90       11.90
21.50   13.25       8.25
```

18. Input real numbers *a* and *b*. Compute and display (*a* +*b*) / (*a* – *b*). Use a *while* loop to test input values to be sure (*a* – *b*) ≠ 0 before attempting computation. If (*a* – *b*) is 0, input new values for *a* and *b*.
Sample displayed input:

```
Enter two real values whose sum is not 0:
23.5 -23.5
Enter two real values whose sum is not 0:
23.5 10.6543
```

Corresponding output:

```
23.500
10.654
(a-b)/(a+b) = 0.376
```

19. Loop to input Celsius (Centigrade) temperatures and print corresponding Fahrenheit temperatures until 999.0 is input. The formula for converting Celsius values to Fahrenheit values is:
Fahrenheit = 1.8 * Celsius + 32
20. Write a program that has no input and outputs the following:

```
abcde
bcdef
cdefg
defgh
efghi
```

using nested loops.
21. Write a program that inputs values until the user enters 0 and outputs “Ascending” if each value before the last is greater than or equal to its predecessor; otherwise it outputs, "Not ascending."
22. If you invest $1.00 today and the investment accumulates 5% interest each year for 100 years, what will it yield to your great-great-grandchildren then? Write a program to compute the result.

Answers to study questions on topic 5

[*Correlation with questions to be verified*]

1. The if branch statement

1. c. Statements and expressions are not tokens; the left parenthesis is mandatory.
2. f. The keyword *else* is part of the *if* statement and should be preceded by a semicolon.
3. b. The *if* statement is a selection statement.
4. t.
5. t.
6. t.
7. f.

2. Relational and Boolean expressions

8. d. The other relational operators are <, ==, <=, >=, !=.
9. c. The OR logical operator takes two logical operands.
10. c. The Java *not* operator is "!", which turns all zero values to 1 and all nonzero values to zero.
11. b. The expression is false because *b* (1) is not equal to *c* (3). False evaluates to 0.
12. c.
13. a. The && operator is the logical operator of conjunction, AND.
14. t. 3 is greater than 2, and the OR operator returns true if either operand is true.
15. t.
16. c. The two equal signs are a single relational operator. The answer is not (a) because the = operator is the assignment operator.
1. d.
2. b. The test fails on both terms.
3. a. || is the logical operator of disjunction; OR
4. b. != is the relational operator for inequality.
5. f.
6. f.
7. b.
8. e.
9. c.
10. c.

3. Multiway branches: switch

1. t. A cascade of *if*s is replaced in *switch* by a list of case labels and associated statements.
2. e. A case label is part of a *switch* (branch) statement.
3. c. In a *switch* statement, a *break* is needed after the statements under a case label to prevent the statements following later case labels from executing.
4. e. A *switch* statement should contain case labels, followed by statements that will execute when a match to the switch selector is found.
5. f.
6. t.
7. t.

Answers to study questions on loops

[To be allocated to mult-choice questions under subtopics 4-5]

1. b. The *while* loop can test for end-of-file at the start, as is necessary.
2. d. The test occurs before the body of the *while* loop.
3. c. Any loop statement may have a counter; the *for* statement is designed especially to support counted loops.
4. b. The ++ operator adds 1 to the value of its variable operand.
5. b. Any loop should have an exit condition that is able to become *true*.
6. t. The exit condition is tested after the body, so the body will execute at least once.
7. c. The body of a *do...while* loop may be a simple or compound statement.
8. a. The counter variable is manipulated by the expression-statements in parentheses after *for*.
9. c. The three expression-statements are separated by semicolon and their order is significant.
10. t. Any of the three loops may be used to solve a given problem; which one is best is a matter of convenience.
11. a. A loop should test and exit either before or after the execution of the body.
12. c. To exit a loop or a *switch* statement, we use *break*.

Answers to short-answer questions

1.
```
for (int i = 100; i<= 200; ++i)
   out.println( i+ " ";
```

13. while, do…while, for
14. 400 asterisks
15. ****
16. 50 asterisks
17. do...while
18. A for loop has in parentheses an initialization, an exit/continuation test, and an update.
19. The loop must have a compound statement as its body.
20. The loop will be infinite.

Short-answer:

1. if, if…else, switch
2. or
3. if … else
4. break
5. height > 72 || age >= 30
6. switch
7. if (quantity < 0)
   out.println(“Invalid value\n”;
8. height >= 60 && height <= 72
9. BC
10. default
11. !, ||, &&
12. <, >, <=, >=, !=, ==
13. break
14.
15.
16.
17.
18.
```
switch (input)
{
   case ‘a’:
         out.println( “Hello”;
         break;
   case ‘b’:
         out.println( “One moment”;
         break;
   case ‘c’:
         out.println( “Bye”;
}
```
19.
20.
21.
22.
23.
24.
25.
26.
27.
28.
29.
30.
31.
32.
33.
34.
35.
36.
37. **(a) ==, !=, <, >, <=, >= (b) !, ||, &&**
38. **`a >= 3 && answer == ‘Y’`**
39.
```
if (quantity > 10)
{
   price *= 0.95;
   out.println( “discounted”;
}
```
40. 0 (false); 1 (true); 0 (false)
41.

Home

Study questions on Topic 6: Methods and classes

1. Modular decomposition

1. (T-F) Procedural abstraction consists of breaking down a problem or program into simpler parts.
2. A way to make a program more modular is to (a) document variables with comments; (b) print clear output; (c) use file input; (d) write method definitions
3. The top-down approach (a) breaks down a problem and solves it step by step; (b) begins with ready-made components and puts them together; (c) focuses on minimizing the number of methods; (d) originated with object-oriented programming
4. (T-F) According to standard practice, a single method should perform a wide variety of tasks.

2. Java methods (subprograms)

1. (T-F) A method body must contain the name of the method.
2. (T-F) The *void* data type is used when it is not known what the user will input.
3. How many definitions must a method have in a program? (a) 0; (b) exactly 1; (c) 2; (d) at least one
4. Reasons to define methods include (a) maximizing the number of identifiers and reducing code repetition; (b) modularity and maximizing the number of identifiers; (c) modularity and reducing code repetition; (d) reducing code repetition and increasing the size of programs; (e) modularity and increasing program size
5. A method definition (a) is a method call; (b) has a header and a body; (c) terminates with a semicolon; (d) may be used as a program statement; (e) none of these
6. In a method call we might find (a) a method definition; (b) a method header; (c) a formal parameter; (d) an actual parameter; (e) none of these
7. (T-F) A method call should contain a data type name.
8. (T-F) A method definition should contain a data type name.
9. A Java expression that ends with () is always (a) a loop (b) an arithmetic expression; (c) a method call; (d) a declaration; (e) an assignment

3. Variables, parameters, and return values

1. Variables are located (a) within the method's machine code; (b) in secondary storage; (c) on the stack; (d) in the microprocessor; (e) all of these
2. A value may be passed out of a method to the calling statement with a (a) value parameter; (b) reference parameter; (c) variable parameter; (d) *goto* statement
3. (T-F) All parameters and variables declared by name are passed using the stack.
4. (T-F) A formal parameter appears in parentheses in a method call.
5. With an *int* parameter, what is passed to the method? (a) the address; (b) the value of an expression; (c) the full text of the expression; (d) nothing; (e) a request for information
6. A method whose name is used as an expression in a program should have (a) a value parameter; (b) a reference parameter; (c) a return value; (d) none of these; (e) all of these
7. Parameter values are stored (a) with the calling method's machine code; (b) with the called method's machine code; (c) in the calling method's activation record on the stack; (d) in the called method's activation record on the stack; (e) in a program's variable memory
8. The stack is like a (a) bulletin board; (b) rope; (c) bookshelf; (d) chain; (e) none of these
9. (T-F) The values of all parameter expressions are copied from the calling method to the called method.
10. A method that returns a value (a) must do so with a parameter; (b) should specify the data type of that value in the method header. (c) does so with an assignment statement; (d) does so automatically; (e) none of these
11. (T-F) A recursive method should have an *if* or *switch* statement in it.
12. (T-F) A recursive method is one that calls itself.
13. Which is *not* a kind of loop statement in Java? (a) *repeat*; (b) counter-driven; (c) top-tested; (d) bottom-tested; (e) recursive
14. Recursion is (a) a way to write a loop; (b) a kind of file input; (c) a nested loop statement; (d) used in all loops; (e) none of these

4. Data abstraction and classes

1. (T-F) A class is a data type as opposed to a data item.
2. In a Java program, the identifier *part.price* could represent (a) a class; (b) an object type; (c) a method header; (d) a member of an object; (e) none of these
3. (T-F) A method may be a member of a class.
4. (T-F) An object may be passed as a parameter.
5. Up to how many objects may be instances of the same class? (a) 0; (b) 1; (c) 2; (d) several; (e) there is no particular limit
6. (T-F) An identifier declared in braces after *enum* is a sub-component inside an enumerated-type data item
7. (T-F) With *enum* we supply a full list of possible values of a data type
8. An enumerated-type value is (a) a compound item; (b) true or false; (c) an *int*; (d) a *float*; (e) a *char*
9. A compound data type may be created with (a) *class*; (b) *enum*; (c) *int*; (d) *for*
10. (T-F) An object value may be passed as a parameter or returned by a method
11. (T-F) A object may be assigned a value only by assigning values to its members one at a time.
12. It is often useful to (a) create a class especially for a certain method; (b) create a method especially for a certain class; (c) declare an object without a type name; (d) declare global methods to manipulate global object variables
13. A data variable may have several data attributes and a set of characteristic behaviors is (a) a type; (b) an integer; (c) an object; (d) a class; (e) a control structure
14. (T-F) A member item in a class is accessible to only one method.
15. Creating new data types is (a) impossible; (b) data abstraction; (c) procedural abstraction; (d) to be discouraged
16. (T-F) A class is an instance of an object.
17. (T-F) Objects are widely considered a factor in making it harder to write longer programs.
18. (T-F) Methods that are members of a class share access to all data members of an object of that class.

5. Interface and implementation

1. What keyword is used to prevent a member item from being accessed from outside a class's methods? (a) *restricted*; (b) *local*; (c) *private*; (d) *public*; (e) *protected*
2. (T-F) A constructor should be called only when the programmer is ready to initialize members of an object with new values.
3. An object has its members initialized in the statement that declares it, by (a) the initialization operator once for each member; (b) a constructor; (c) an access method; (d) *Scanner*; (e) *System.out*
4. (T-F) A program calls a constructor only if the programmer uses its name in a statement.
5. (T-F) An instance of a class may have its members automatically initialized at the time it is declared.

Short-answer questions on T6 (Methods and classes)

Methods

1. If *items* is a object type with a *float* member named *price*, then write a statement to assign the value 4.99 to an instance of *items* named *notebook*.
2. Write an enumerated type declaration denoting the four seasons and a declaration for a variable, *season,* of that type.
3. What Java techniques presented thus far may be used to share data among different methods? Which do the textbook authors prefer?
4. Is there a syntax error in this program? If so, find it and say how to fix it.

```
/* persnbug.cpp
Prompts for and displays person name and age.
*/
#include <iostream.h>
struct persons
{
  char name[80];
  int age;
}
void main()
{
  out.println( "\nName: ";
  persons person;
  cin >> person.name;
  out.println( "Age: ";
  cin >> person.age;
  out.println( person.name
      + " is "
      + person.age
      + " years old.\n";
}
```

5. How are objects and character strings alike, as opposed to the data types *int, char,* and *float*?
6. Is the following declaration legal? What if anything is wrong with it?

```
struct employees
{
  char name[40];
  long salary;
  employees supervisor;
}
```

7. What is the difference between (a) an array and an object; (b) a member method and a member variable; (c) a class and an object?
8. How is a time capsule similar to an object that encapsulates data and methods?
9. Write a declaration of a *car* class, naming some of its features and allowing values to be specified for them.
10. Why is the following code invalid?

```
struct bits
{
    char name[80];
    bits bit;
};
```

11. Why is the following code invalid?

```
struct ID
{
  char name[80];
  int size = 10;
};
```

12. What is the number of bytes occupied by an instance of the following object type?

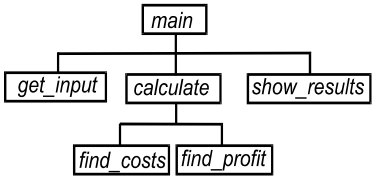
```
struct employees
{
  char name[20];
  long salary;
  int age;
};
```

13. In a program that declares a class *children* and reads the names and ages of several children from a data file in order to find the average age, will the C++ code that finds the average appear in a class's member method or in a free method?
14. Why would a programmer wish to declare a new data type?
15. In a program that declared a class *employees* and an instance *emp* of that class, would it be appropriate for method *employees::display_info* to refer to *emp*? Why or why not?
16. Define a class *Circle* with members *x, y* and *radius,*including a constructors that initializes these all to 0.
17. What tools does an application programmer have in addition to the keywords and operators of the Java language?
18. Make up names for three methods that might be used in a program that consists of these three modules:
    (a) Input an item purchase price from the keyboard;
    (b) Compute a sales tax and a total sale amount for the purchased item;
    (c) Display purchase price, tax amount, and total sale amount.
19. What is missing from this method definition?

```
void fetch_price()
out.println( "Enter price: "
cin >> price;
```

20. Give two reasons for using a modular approach to program development.
21. What is the difference between a method declaration and a method definition?
22. What must be true of calls to methods with the *void* type designation?
24. What are *two* ways that a method can pass a data value back to the statement that called it?
25. Write the *header* for a method that accepts two *float* parameters and returns their *integer* sum.

26. Write an appropriate *declaration* for a method that accepts two integers, returns no value, and draws a figure. You need not spell out everything that the method does.
27. In the module hierarchy diagram below, (a) name methods that call *show_results;* (b) name methods that are called by *calculate*; (c) does *get_input* call *show_results*? (d) does *find_costs call calculate*? (e) does *main* call *find_profit*? (f) does *calculate* call *find_profit*?



28. Name a technique used in debugging to pinpoint the source of an incorrect output value.
29. What is one way to share data among methods without using parameters or return values? What are its disadvantages?
30. Put these phrases or subphrases of the problem-solving process in chronological order, numbering the first "1", the second "2", etc.
    _____ code program
    _____ desk check
    _____ write a design
    _____ get problem specifications
    _____ test program
    _____ debug code
31. Draw a module-hierarchy chart for the following skeleton program.

```
// prog.cpp: Does nothing
void fa();
void fb();
void fc();
void fd();
void main()
{ fa(); fd(); }
void fa() { fb(); fc(); }
void fb() { }
void fc() { }
void fd() { }
```

32. Do the statements in a method always execute at least once when the program containing the method executes? Justify your answer.
33. Find the syntax or other errors in the programs below

(a)
```
// hello2.cpp
#include <iostream.h>
void say_hello();
void main() { say_hello(); }
void say_helo() {}
```

(b)
```
// hello3.cpp
#include <iostream.h>
void say_hello();
void main() { say_helo(); }
void say_hello() { }
```

(c)
```
// hello4.cpp
#include <iostream.h>
void say_hello()
void main() { say_hello(); }
void say_hello() { }
```

(d)
```
// hello5.cpp
#include <iostream.h>
void say_hello();
void main() { say_hello(); }
void say_hello() { };
```

(e)
```
// hello5.cpp
#include <iostream.h>
void say_hello();
void main()
{
   // says hello
   say_hello();
}
void say_hello()
{
   // This method is empty
}
```

22. Could a *void* method contain a *return* statement, and if so, what would it look like?
23. How many different integer memory locations called *x* does this program contain? Where are they declared?

```
// x.cpp
#include <iostream.h>

void getx(int& x);
void putx(int x);

void main()
{
   int x = 8;
   getx(x);
   putx(x);
}

void getx(int& x)
{
   cin >> x;
}

void putx(int x)
{
   out.println( x +  ;
}
```

24. In program example *x.cpp* (Problem 18 above), name:
    (a) An actual value parameter;
    (b) An actual variable parameter;
    (c) A formal value parameter;
    (d) A formal variable parameter.
25. What is the difference between an actual parameter and a formal parameter?
26. What is the difference between a reference parameter and a value parameter?

27. What information other than a parameter name is required in a parameter declaration?
28. Could the expression (*age – 2*) be a reference parameter? Explain.
29. What can you conclude about two variables or parameters, used in two different methods, if the data items have the same identifier?
30. When a reference parameter of type *char* is passed, how many bytes of data are copied to the stack and communicated to the called method?
31. Name one positive aspect of recursion as opposed to iteration, from a programming point of view, and one negative aspect from a machine-operation point of view.
32. What operator, if any, is used to declare (a) a reference parameter; (b) a value parameter.
33. What will the compiler produce from the following code?

```
// thunk.cpp
#include <iostream.h>
const int ZIP = 01776;
void blip(int& jabr);

void main()
{
   blip(ZIP);
}

void blip(int& jabr)
{
   JABR = -JABR;
}
```

34. What is wrong here?

```
// sub.cpp
#include <iostream.h>
void subtract(int a,int b,
              int& diff);
void main()
{
   subtract(2,3,-1);
}

void subtract(int a,
      int b,int& diff)
{
   diff = a - b;
}
```

35. What would be a good declaration for a method that takes parameters and that displays the interest amount on a loan, given principal, interest, and term of loan?
36. What is wrong with this method declaration:
    `void draw_rectangle(int width, ht);`
37. What is wrong with this recursive method?

```
void get_amt()
{
   out.println( "Sales amt: ";
   float amt;
   cin >> amt;
   get_amt();
}
```

38. What is a difference between the body of any *void* method and the body of a non-*void* method?
39. Label each term below with the letter of its appropriate definition:

    _____ debugging
    _____ trace statement
    _____ method
    _____ method declaration
    _____ method definition
    _____ global variable
    _____ intermediate variable
    _____ local variable
    _____ module
    _____ procedural abstraction
    _____ recursion
    _____ activation record
    _____ actual parameter
    _____ base case
    _____ formal parameter
    _____ method call
    _____ method declaration
    _____ parameter passing
    _____ pass-by-reference
    _____ pass-by-value
    _____ recursive case
    _____ return value
    _____ stack

    a) Code that spells out all steps taken by a C++ subprogram.
    b) Giving a name to a series of program statements, for use in a program.
    c) Finding and fixing logic errors.
    d) A data item not input or output.
    e) Code that allows a programmer to see values of variables while debugging.
    f) A data item accessible to all methods in a program.
    g) A data item accessible only to method that declares it.
    h) A C++ subprogram.
    i) use of a parameter that may communicate a new value back to the calling statement
    j) mechanism for passing a value to calling statement using the method call as an expression
    k) introduction of subprogram name to a program
    l) argument as found in method header

m) data item that stores a method invocation in memory
n) nonrecursive branch in recursive method
o) language mechanism for communicating data between methods
p) use of a parameter that communicates data into called method but not out
q) invocation of subprogram
r) argument found in method call
s) attribute of a method that calls itself
t) memory structure for sharing data between methods
u) mechanism for automatically initializing member data items
v) statement in recursive method in which method calls itself

Classes

1. Name a keyword that may be used to declare a data type with which to declare objects.
2. Write a statement that assigns the value 82 to the *quantity* member of an object, *part*, which is an instance of class, *parts.*
3. Distinguish between a object variable and a object type.
4. Where is a member identifier declared—in a object variable declaration, in a object type declaration, or in its own variable declaration?
5. If *employee.id_num* is an integer, what kind of data item is *employee*?
6. Consider the following class declaration:

```
class employees
{
public:
  employees();
  employees(char nm[],int hrs);
  char* get_name();
  int get_hours();
  void set_name(char nm[]);
  void set_hours(int hrs);
  void input();
  void display();
private:
  String name;
  int hours;
};
```

It has:
How many members? _____
How many member data items? _____
How many constructors?

7. Write a module hierarchy chart for a program that inputs two numbers from the user, *a* and *b*, calculates the value a^b (*a* raised to the *b* power), and displays the results of the calculation. You do not need to write a program, but your answer should include meaningful module names and show how the modules are related to each other. Write method headers, including parameter declarations.
8. Label each term below with the letter of the appropriate definition:

_____ class
_____ data abstraction
_____ encapsulation
_____ member method
_____ member variable
_____ object
_____ object-based design
_____ object
_____ object type

(a) A compound data item composed of member items whose types are chosen by the programmer.
(b) A data item that is a component of a object or object.
(c) A named category of data items that may be used to declare a object.
(d) The practice of defining new data types.
(e) An object data type, whose instances are defined by their data attributes and their behaviors.
(f) An instance of a class.
(g) The practice of writing plans for programs with an eye to the concepts being modeled, as defined by their data components and behaviors.
(h) The practice of aggregating several data items and separating them from the rest of a program.
(i) A subprogram associated with a class .

9.

Longer answer problems for topic 6 (Methods and classes)

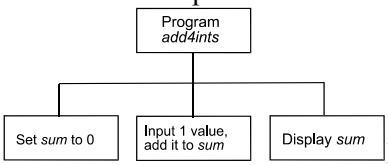
Objective 16: Define a Java method with parameters and a return value

1. Write a program that defines and calls *two* methods: one to prompt for, input, and return the size of a square, and the other to draw a *rectangle,* composed of X's like that below. Your drawing method should accept parameters for both the height and the width of the figure. For a square, the width and height are the same.
   Sample I/O:
   Size of square: 3
   XXX
   XXX
   XXX
2. Write the definition of a method that accepts the positions of two opposite corners of a rectangle as parameters and returns the area of the rectangle.
3. Write a program that reads a text file composed of lines and displays it on the screen.
4. Write a method that accepts an integer parameter, *length*, and displays a line of that many hyphens.
5. Write the *definition* of a method that accepts two integer values as arguments and that returns the smaller one
6. Break down the following program into three methods, *input, calculate,* and *display,* each called from *main*. Use parameters rather than global variables to share data among methods.

```
// final.cpp
#include <iostream.h>
void main()
{
int in1,in2;
do {
out.println( "Enter 2 non-negative integers:";
in >> in1 >> in2;
} while(in1 < 0 || in2 < 0);
int product = 1;
for (int i=0; i < in1; ++i)
product *= in2;
out.println( input1 + " to the "
    + input2 + " power is " + product +
}
```

Objective 17: Define a Java class

7. Write a program that declares, calls, and defines a method that accepts three integer values as parameters and returns the smallest one.
8. Use two methods, one to draw a horizontal line of eight asterisks and the other to draw two asterisks separated by six spaces on the same line.
10. Modify your solution to Exercise 1 to include a new method that calls the other two. Each time the new method is called, a box is drawn.
11. Write a method that repeatedly prompts for and inputs integer values, until the user enters a zero value, and displays the smallest one and the average.
12. Write a program containing a method, *get_food*, that asks the user for a name of a food item to be input from the keyboard, and writes the name to a disk file. Call the method four times to put a meal together. Print the meal file.
13. Here is a structure diagram of a modularized program that adds four input values and shows the sum:



Implement this design, using a method for each of the three indicated modules. Note that the method for inputting a single value and adding it to the *sum* will need to be called more than once. Document your program with comments, user prompts, and clearly identified output.

14. State income tax for the State of Panic is computed according to the following formula:
    3% on net annual income up through $8000.00, plus
    5% on net income in the range $8000.01 – 15000.00, plus
    8% on net income in excess of $15000.00.
    Net income is *gross* income minus deductions.
15. Use a *while* loop to permit input of multiple sets of data (name, gross sales, and deductions). Use separate methods for (1) data input, (2) tax computation, and (3) a tax report written to the screen.

16. *Sample Tax Report:*

```
17. Name       Gross  Deductions     Net    Tax
18. Joe Smith  23000.00      7000.0016000.00
    670.00
19. Linda Tucker      25000.00      15000.00
    10000.00      340.00
```

20. **Harry Carey** **17000.00** 9000.008000.00 240.00
21. Use separate methods to compute the area of a square and the area of a circle using the same input real value as side and diameter, respectively. Use the *M_PI* constant (See Chapter 3) in your calculation of the area of the circle. Use a *do...while* loop to permit multiple entries. The area of a circle of radius r is equal to $\pi r2$.
22. The faculty of Commuter Community College have been voted an across-the-board 5.5% pay increase. Input an ID number and old salary for each professor. Use a single method to : (a) compute a new salary; (b) keep a running total of the old salaries; and (c) keep a running total of the new salaries. Use no global variables. Display a salary report similar to the

following sample:

ID	OLD SALARY	PAY RAISE	NEW SALARY
1	25000.00	1375.00	26375.00
2	28058.00	1543.19	29601.19
3	32000.00	1760.00	33760.00
4	26500.00	145.50	2795.50
5	31000.00	1705.00	32705.00
TOTALS	364558.00	20050.69	384608.69

23. Write a program to input a positive real number *a* and an integer *b*. Use a method to compute *ab*. Include a loop in the method *main* to allow for multiple sets of input data. Use an input of $a = 0.0$ as a signal value to exit the loop. Remember that $a0 = 1$ for $a \neq 0$.
24. Use a method to compute the sum of the first *n* terms of the geometric series:
    $a + ar + ar2 + ar3 + ... + ar(n–1)$
    where *a* (real), *r* (real), and *n* (integer) are input from the keyboard. The method should accept three parameters.
    For example, if $a = 2$, $r = 3$, and $n = 4$, then the method should return $(2 + 2*3 + 2*3*3 + 2*3*3*3) = 80.0$.
25. The number sequence (1, 1, 2, 3, 5, 8, 13, ...) is known as the Fibonacci sequence. The first and second terms are both 1; the third and all subsequent terms are the sum of the two immediately preceding terms. For example, the 7th term is the sum of the 5th and 6th terms.
26. Write a program to input an integer *n*, where $n \geq 1$; compute and display the first *n* terms of the Fibonacci sequence. Use a method to compute and display the third and all subsequent terms.
    Sample I/O:
    ```
    Enter number of Fibonacci terms desired (n > 0): 8
    1 1 2 3 5 8 13 21
    ```
27. Input an unsorted list of integers. Let the user enter a sentinel value of 999 to terminate input. Use a single method to pass back to *main* both the largest and smallest values in the list. Don't include the sentinel value (999) in the output, or for consideration as largest or smallest.
28. (Challenge) Input two positive integers. Use a method to compute the greatest common divisor (GCD). The greatest common divisor of two integers is the largest integer that is a factor of both integers. Include a loop for multiple sets of input. Use a sentinel value of 0 to exit the input loop.
    Sample I/O:
    ```
    Enter first integer (0 to  exit): 19626
    Enter second integer: 20184
    Greatest common divisor: 6
    Enter first integer (0 to  exit): 2001
    Enter second integer: 15
    Greatest common divisor: 3
    Enter first integer (0 to  exit): 0
    ```
29. Descriptive statistics are grouped into two categories, measures of central tendency and measures of dispersion. Measures of central tendency tend to locate the middle of a set of data. Common measures of central tendency are:
    • The mean: The sum of a set of *n* values, divided by *n*. (The mean is commonly referred to as the 'average'.)
    • The median : The middle value when the data is placed in order of size. The position of the median of *n* ordered values is $(n+1)/2$.
    • The mode: The most frequently occurring data value.
30. Measures of dispersion tend to measure the spread of a set of data. Common measures of dispersion are:
    • The range: The difference between the largest value and the smallest value.
    • The standard deviation: One general formula for the standard deviation of *n* values ($x_1, x_2, x_3, ..., x_n$) having a mean (a) is:
    $\sqrt{(x_1 - a)^2 + (x_2 - a)^2 + ..+ (x_n - a)^2}$
31. Write a program that will compute the mean and standard deviation of four input values. Use separate methods to compute the two statistics.
    Sample I/O:
    ```
    ENTER 4 VALUES (0 0 0 0 to  quit): 90 75 83 62
    AVERAGE: 7500
    STANDARD DEVIATION: 10.404
    ENTER 4 VALUES (0 0 0 0 to  quit): 58 98 72 72
    AVERAGE: 75.000
    STANDARD DEVIATION: 14.457
    ENTER 4 VALUES (0 0 0 0 to  quit): 0 0 0 0
    ```
32. Write a program that does Roman numeral arithmetic. Two Roman values and an operator (+, –, *, /) are input. After each Roman operand is input, it is converted to, and redisplayed as, a decimal value. After the operator is input, both the Roman and decimal results of the operation are displayed. Test for, and reject, invalid input data.
33. Write a program to list all the factors of an input integer. If the input value is a prime number, display a message to that effect. Loop for multiple inputs. Exit from the loop if the input value is less than or equal to 1.
34. Write a program that prompts the user for four specifications (column value of upper left corner, row value of upper left corner, width, and height) for a rectangular box, composed of asterisks, to be displayed on the screen. Use a method that accepts the four specifications as parameters.
35. Write a method that tells whether its integer parameter is odd.
36. (Challenge) What common method presented in calculus courses is computed by this C++ method?
    ```
    int f(float n)
    {
        if (n < 2)
                return 0;
        else
                if (n == 2)
                        return 1;
                else
    ```

```
    return f(n / 2) + 1;
}
```

37. Write a program containing a Boolean method to accept a string and to test whether or not an uppercase letter has been input. Include a loop to allow for multiple inputs. Exit the loop on any non-letter input.
38. Use a *float* method to return the greater of two input values. Include a loop to allow for multiple sets of input. Exit from the loop when the two input values are equal.
39. Write a program that accepts a person's full name and prints the initials. Include a method that returns the first letter after a space in a string, starting at a given character position.
40. Write a string editor—a program that allows the user to edit a source string of not more than 60 characters. The program is to allow at least the following choices:
    1. Enter a new string
    2. Insert a substring
    3. Replace one substring with another
    4. Delete a substring
    5. Quit

    Use a separate method for each of the first four menu options.
41. Write a method *pos* (*substring*,*string*) that returns the position of the first character of *substring* in *string*, or returns –1 if *string* does not contain *substring*.
42. Write a recursive method that returns the product of two positive-integer arguments. The method may not use the '*' operator or any loop statement.
43. Write a recursive method that returns the highest power of 2 that is less than or equal to an input number.

    Sample I/O:

```
Enter a number: 14
8
Enter a number: 233
128
Enter a number: 64
64
```

44. Write two Boolean methods that each take two Boolean arguments and simulate a logic gate, one for the XOR gate, one for the NAND gate, as these are defined in Chapter 1, Keil/Johnson.
45. Write a method that accepts two strings and an integer. The integer should represent a position in the first string. The method should insert the second string into the first at the position indicated. For example, if string variable *s1* has the value "resion", a method call *insert(s1,"cur",2)* should result in *s1* having a value "recursion".
46. Write a method that accepts a string and a character as parameters and returns the number of occurrences of the character in the string. For example, on parameters of "tree" and 'e', your method should return 2.
47. Write and demonstrate three methods all named *prompt_for* that accept a prompt message string and return an integer, a character, and a floating-point value, respectively. Test all three methods in one program. What is the C++ feature that lets you use *prompt_for* as the name of all three methods in the same program?
48. Write a method to return the integer value of a string parameter. For example, if the string parameter is "43", the method should return the integer 43.
49. Write a short Java program that displays a table of the first 20 powers of 2 (2^1, 2^2, …, 2^{20}), formatted below using *bitwise operators* (not the *pow* method) to calculate the entries in the right-hand column of the table. The first three rows will be:

```
          2 to the
n         nth power
1         2
2         4
3         8
```

50. Write a program that prompts for a person's first name, up to 80 characters, and display it backwards.
51. Write a method that takes a string parameter and swaps its first and second words, using loops to iterate. Define words as sequences of characters separated by a space.

    Text of main:

```
char s[] = "Jane Smith";
swap_strings(s);
out.println( s +  ;
```

    Output:

```
Smith Jane
```

52. Write a program with no input that for $n = 1$ to 100, displays $\frac{n^2 (n+1)^2}{4}$ and $\sum_{k=1}^{n} k^3$ and the difference between them.

    Output:

```
n  quotient    sum   diff
1  1     1     0
2  …     …     …
…
```

Methods

1. Declare a class to represent a machine part, with a name, an inventory quantity, and a price. Declare and define methods to input data and to display the member data values and the part's current inventory-on-hand value (the product of quantity and price).
2. Write a declaration for a class to represent models of refrigerator. Each model has a price, an identification number, and a name. Write input and display methods for the class, access methods, and a constructor.
3. Declare a object type or class for a machine part, with an identification number, an inventory quantity, and a price. Declare, define and call methods to input and display data about a part.

4. Write a program that prompts for prices and sales figures (quantity 0 to 40 thousand) for a product. Define a class of products and store a product as an instance of the class. Write member methods to input and display information about a product, including a bar graph that reflects the quantity sold, in thousands.
5. The data file *children.dat* (below) records names and ages of pupils. Using what you have learned of objects and objects, write a program that reads these four records, displays the names, and displays the *average* age.

   Contents of *children.dat*:

   ```
   Wendy 13
   Jessica 15
   Tom 10
   Bill 12
   ```
6. Debug the following program.

```
/* employee.cpp
  Prompts for, displays,
  employee data. */
#include <iostream.h>
struct employees
{
   char name[20];
   long salary;
   int age;
}
void main()
{
   employees emp;
   out.println( "Name,salary,"
         + “age: ";
   cin >> emp.name
      >> emp.salary
         >> emp.age;
   out.println( "Name: "
    + emp.name +
      + "Salary: "
      + emp.salary +
      + "Age: " + emp.age
      +  ;
}
```

7. Declare some C++ classes and define some of their member methods to represent a registra-tion for classes at a college. A registration has a student ID number, a course number, and a date. A date has a month, day, and year.
8. Name some classes and their members, or draw a class diagram, to correspond to the following description: “The Environmental Protection Agency reports to the administration and to Congress. It monitors the activities of industries that may affect the environment and measures levels of certain substances in the air and water.”
9. Write a object type or class for times of the day that could be expressed on a clock.
10. Write a object type or class for durations of time using the units of time found in a clock.
11. Write a object type or class for a date on a calendar.
12. Write a object type or class for a bank customer’s transaction such as the kind that an ATM could carry out.
13. Write a class, *rationals*, to encapsulate rational numbers. Each rational number has a numerator and a denominator, which are integers. Include member methods *plus*, *minus*, *times,* and *divided_by*. Each should accepts a rational-number parameter and return a rational number that expresses the result of the indicated arithmetic operation.
14. Write a set of methods that implement addition, subtraction, multiplication, and division for the object type *rationals* by accepting a parameter of type *rationals.*
15. Write a method that accepts two instances of type *rationals* and returns a value of type *rationals* that is the product of the two parameters.

Answers to study questions on Topic 6 (Methods and classes)

[*Correlation with questions to be verified*]

ST1

1. t. Defining methods in order to modularize a program is called procedural abstraction.
2. d. Writing method definitions is a way to break down a program or problem into smaller, easily understood, independent units.
3. a. The solution to each sub-problem becomes a module or method.
4. f. A goal of modularity is *cohesion* within each method: the purpose of the method should be singular and focused.

ST2

5. f. The method name is in the header.
6. f. The user's input is rarely known by the programmer; the *void* type is used for methods that return no value.
7. b. A method must have one definition and may have many declarations.
8. c. Breaking down a program into components and grouping related, repeated sequences of statements are the main goals of writing methods.
9. b. The method header has the same form as a method declaration, without the semicolon; the body is a compound statement.
10. d. Actual parameters occur in method calls, formal parameters in method definitions and declarations.
11. f. A method call may be an expression or statement and needs no type name.
12. t. A method definition's header must begin with the method's return type.

ST3

13. c. A local variable is allocated when a method begins to execute and disappears when the method terminates.
14. b. A reference parameter is specified with "&" after the type name.
15. t. The stack allows a method to share data and to allocate space that is freed up when the method terminates.
16. f. The formal parameter appears in the method definition; the method call may contain actual parameters.
17. b. A value parameter is an expression that is evaluated before the value is passed.
18. c. The return value follows the keyword *return* and its type should be named in the method header.
19. d. Parameters declared in a method's header disappear when that method call's activation record disappears from the stack.
20. a. The stack, like a bulletin board, allows methods to share information as coworkers do.
21. f. Reference and array parameters have only their addresses copied.
22. b. A data type is associated with every method.
23. t. The branch should allow the method to process the base case, allowing the method to terminate, or the recursive case, causing the method to call itself.
24. t. A recursive method calls itself and must also have a way to terminate without calling itself.
25. a. Java has no *repeat* statement.
26. a. A method that invokes itself is recursive and the result is a loop.

ST4

1. t. An object is a data item that is an instance of a class. A class is like a template or category.
2. d. The period separates the object name from the member item.
3. t. A method is declared in the class declaration.
4. a. The scope resolution operator specifies the class of which an identifier is a member, whether a constant, as in *ios::fixed*, or a method in the method's definition.
5. t. Use the object type name to declare the object parameter.
6. e. A data type may be instantiated any number of times consistent with memory capacity.
7. f. A constant named in an enumerated type declaration is a possible value of an item, not a component.
8. t. As in *enum answers {Yes, No, Maybe};*
9. c. The values of the constants named in an *enum* are 0, 1, 2, …
10. a. A object is a compound item, composed of its members.
11. t. A object is like a simple-type data item in parameter passing.
12. f. A object variable may be initialized when declared, with a list of values between braces, or assigned the value of another object with the assignment operator.

ST5

1. b. A method that accepts a object as a parameter in effect implements an operation associated with that object type.
2. f. Members are private, unless they follow the keyword *public*.
3. c. A private member is visible only to methods of its class.
4. f. A constructor is called automatically when an object instance is declared.
5. b. A parameter list may follow the object's name in an object instance declaration. The parameters are passed to the constructor of the object's class.

Short-answer (classes):

1. notebook.price = 4.99;
2. struct; class
3. part.quantity = 82;
4. enum seasons { Spring, Summer, Fall, Winter };
5. global variables, parameter passing, member data items
6. A object variable occupies memory, has a value, and is an instance of a object type, which does not have a value or occupy memory.
7. object type declaration
8. employee is a object or object.
9. error: need semicolon after object type declaration
10. Objects and character strings are compound, the other types named are simple.
11. Missing semicolon at end
12. (a) A object, unlike an object, may not call methods.

Short-answer (methods):

1. return value; reference parameter
2. *int sum(float a, float b);*
3. *void draw_fig(int a, int b);*
4. (a) *main;* (b) *find_costs, find_profit;* (c) no; (d) no; (e) no; (f) yes

Study questions on Topic 7: Arrays in Java

1. Arrays in Java

1. An array is a(n) (a) compound data item; (b) simple data type; (c) package; (d) dynamic data structure; (e) control structure
2. The components of an array are called (a) members; (b) items; (c) elements; (d) enumerations; (e) return values
3. An indexed sequence of storage locations for data items all of the same type is a(n) (a) object; (b) class; (c) array; (d) list; (e) none of these
4. A subscript is (a) an address; (b) an index to the location of an array element; (c) the size of an array; (d) a decrementing operation; (e) a special graphics character
5. If an array is declared *int x = new int[8]*, then *x*[8] is (a) the seventh element; (b) the eighth element; (c) out of bounds; (d) the address of the array itself; (e) zero
6. Array elements are accessed by (a) key; (b) subscript; (c) size; (d) value; (e) none of these
7. To declare an array in Java requires use of the operator (a) +; (b) =; (c) *malloc*; (d) *new*; (e) none of these

2. Populating arrays

1. Curly braces may be used (a) to declare an array; (b) to assign a value in an assignment statement; (c) to initialize an array; (d) to select an array element; (e) none of these
2. (T-F) An array may be initialized by listing elements in braces ("{...}").

3. Array operations and boundary errors

1. (T-F) A Java program performs an automatic check to determine whether an array subscript is out of bounds
2. When an array name is used as a parameter, the data copied is (a) all the values of its elements; (b) the memory address of the array; (c) the value of its first element; (d) all of the above; (e) none of the above
3. *x[6]* may be a method (a) header; (b) call; (c) definition; (d) parameter; (e) name

4. Two-dimensional arrays

1. *part[5][7]* is: (a) an element of a one-dimensional array; (b) an element of a two-dimensional array; (c) an array declaration; (d) a method call; (e) a violation of Java syntax
2. A two-dimensional array is a (a) vector; (b) class; (c) object; (d) matrix; (e) set
3. In a multi-dimensional array, the address of an element is calculated in part by (a) adding row and column number; (b) multiplying row number by the number of columns; (c) adding the size of the base type to the column number; (d) using the column number as an exponent; (e) none of these

5. Collections

1. A collection is (a) atomic; (b) a set of objects of the same class; (c) a compiler operation; (d) a predefined class; (e) none of these
2. A collection may be implemented with a single (a) integer; (b) string; (c) array; (d) object; (e) none of these
3. A collection is implemented as a(n) (a) integer; (b) string; (c) class; (d) object; (e) none of these
4. *ArrayList* is a(n) (a) predefined object; (b) specialized class; (c) generic class; (d) method; (e) none of these
5. One wrapper class is (a) *Integer*; (b) *int*; (c) *String*; (d) *char*; (e) none of these

Short-answer

1. Declare an array of temperature readings, where readings are taken on each day of the week in each of four cities. You need not name cities or initialize the array.
2. Declare an array to store the weights of a series of grocery items.
3. Declare an array that stores the number of computers sold for each 7-day week in a month of up to 5 weeks. You need not initialize the array.
4. Declare an array of sales figures in five corporate divisions, one value of an appropriate numeric type for each division in each.quarter of the year (Q1, Q2, Q3, Q4).
5. Write a declaration for a method, *calc_paycheck*, that could accept as a parameter an element of an array of *float*s.
6. Write a declaration of a method, *calc_average*, that could accept an array of *float*s as a parameter.
7. Roughly how many steps would it take to insert a new value at the *end* of an array of *n* elements?
8. Roughly how many steps would it take to insert a new value at the *beginning* of an array of *n* elements?
9. Identify these as expressions or declarations, and say of what kind they are:
   (a) *int x[5]*;
   (b) *x[2]*
   (c) *char *name;*
   (d) *char name[50];*
   (e) *char *name[100];*
   (f) *int A[5][10];*
   (g) *A[2][4]*
10. Declare an array of 20 data items that each could store a fractional numeric value, using a named constant as the dimension.
11. If *x* is an array of integers, write a statement that passes this array to the method *calc_average*.
12. If *x* is an array of integers, write a statement that passes the first element of this array to the method *calc_paycheck*.
13. If *A* is an array of integers, write a declaration of a method named *display_scores* that returns no value and can accept *A* as a parameter.
14. In one statement, declare and initialize an array of integers to the values 1, 2, and 3.
15. What error do you find in this code?
```
void display(int list[])
{
    out.println( list;
}
```
16. What error do you find in this code?
```
    int list[80];
    list = 5;
```
17. What *two* errors do you find in this code?
```
void disp_sum(int list[],int n)
// Displays sum of <list>
{
    int i,total;
    for(i=0; i < n; ++i)
        total += list[0];
    out.println( "Total = "
         + total +  ;
}
```
18. If the following code outputs garbage, what could be the reason?
```
void disp_sum(int list[],int n)
// Displays element <n> of <list>
{
    out.println( list[n];
}
```
19. Declare an array of 100 floating-point values to record a series of distances in miles. Use a named constant to dimension the array.

Longer-answer

Objective 18: Define and manipulate an array

1. Write a program that reads a series of twelve integers from a file, *budget.txt*, into an array and passes the array to a method that tells whether the integers are in ascending numeric order.
2. Define a data type and associated methods to implement a collection of integers. Write an initializer method, or constructor, and method to input and display the elements of the collection. Write a short *main* method that calls these.
3. Write a program that defines a method that accepts as parameters an array of integers, a single integer denoting the current size of the array, and the value of a new element. The method should append the third parameter, the value of a new element, to the array. Declare and initialize an array in *main* with values 2, 6, 3, 1, and 8, and call the method to insert the value 5 at the end of the array.
4. Write a program that declares and initializes an array of integers, leaving room at the end for uninitialized values. Write code to insert the value 5 at the *beginning* of the array.
5. Write a program that defines a method to accept an array of integers as a parameter. The method should delete the *first* element of the array. Declare and initialize an array in *main* and call the method to delete the first element.
6. Write a program that declares the appropriate object types or classes to implement a collection of inventory records, each with an ID number, a quantity, and a cost. Write associated methods to input and to output inventory records, and to display the whole collection.

7. Write a program that declares and initializes a two-dimensional array that has two rows of seven integers each. Define and use a method that accepts such an array as a parameter and displays all the elements of the array in two rows and seven columns.
8. Declare a class or object type and define associated methods to manage an array of amounts paid for grocery purchases, designed so that the user may add to the list one item at a time. Using Java code or plain English, explain what will be done when a new purchase is added to the list.
9. Write precise pseudocode or C or Java code for a method to do one of the following:
   (a) insert a new value at the beginning of an array of *float*;
   (b) insert a new value at the end of an array of *float.*
10. Write a short program to calculate 2^{100} exactly. (Hint: use a string to store your number as you build it from 1 to 2 to 4 to 2^{100}. To double a number, stored in a string, use the algorithm you learned in elemen-tary school for multiplication, going from the rightmost digit leftward and carrying a tens digit when necessary.)
11. Display the standard deviation of a series of floating-point input values. The standard deviation reflects how widely the members of a series of results tend to differ from the average. It is defined as the square root of the sum of the squares of the average amount by which values differ from the average. For example, the average of {1, 3, 6, 2} is (1 + 3 + 6 + 2) / 4, or 3.0, and the standard deviation is the square root of ($|3–1|^2$ + $|3–3|^2$ + $|3–6|^2$ + $|3–2|^2$), or $\sqrt{4}$, or 2.
12. Some people think that an improbable series of events, such as a run of several heads or tails in flipping coins, is likely to be followed by a series of events that will balance out the improbable sequence. Test this idea by seeing if run of heads generated by the random-number method *rand* is more likely to be followed by a head or a tail. Report how many heads and how many tails follow runs of different lengths. Generate a series of 1000 flips of the coin and keep track of how many runs of different lengths occur. Report results in a table.
13. Write a method that accepts two parameters, equal-sized arrays of *double*, and their size, and copies the second array to the first array. Assume the first array starts as empty.
14. Write a method that accepts an array of *int* as a parameter and finds the length of the longest ascending sequence, i.e., the longest sequence of elements in which each element after the first is at least as large as the previous one.
15. Write a program to input a string and display its vowels (‘a’, ‘e’, ‘i’, ‘o’, ‘u’) in the order in which they appear.
16. Do any of the above using vectors.

Home

Answers to study questions on topic 7

Multiple-choice and T/F

[*Correlation with questions to be verified*]

T1

1. a. The array is composed of its elements.
2. c. The number of elements in an array is specified in braces in the array's declaration.
3. b. The subscript appears in brackets. The first array element has the subscript 0.
4. b. An array identifier passed as a parameter to a method, for example, refers to the address of the array.

T2

1. c. A formal parameter with empty brackets after its name denotes the address of an array of items, as does a parameter with an asterisk preceding its name in the formal parameter declaration.
2. b. The formula is: element address = (array address) + size of base type * (row * num-cols + col #)

T3

3. a. The components of an array are its elements.
4. t. A linear list of values, separated by commas, will be assigned as the initial values of elements of either a linear or multi-dimensioned array.
5. c. The array element *x[6]* can only be a data item; a data value may be used as a parameter to a method.

T4

1. f. Permitting out-of-bounds errors is a common logic error.
2. b. An array element with two subscripts is of an array of arrays, a two-dimensional array.
3. b. A table is a set of records. Since all records are of the same format and type, an array is an appropriate way to store a table.
4. d. A matrix, or grid, has two dimensions.

T5

1. t. An array may be full to capacity or not full at any given moment.

Short-answer

1. `int reading[4][7];`
2. `int sold[5][7];`
3. `float sales[5][4];`
4. *float calc_paycheck(float amt);*
5. *float calc_average(float amt[]);*
6. One step: assign the value to the array element with subscript *n*.
7. *n*+1 steps: one step to move each element one to the right, and one to assign the new value at the beginning.
8. (a) *int x[5];* is a declaration of an array of 5 *int*s
   (b) *x[2]* is an array element expression
   (c) *char *name;* is a declaration of a pointer to character
   (d) *char name[50];* is a declaration of a 50-element string (*char* array)
   (e) *char *name[100];* is a declaration of an array of 100 pointers to *char*
   (f) *int A[5][10];* is a declaration of 2-dimensional array of *int*
   (g) *A[2][4]* is an element of 2-dimensional array
9. const int MAX = 20;
   float sales[MAX];
10. calc_average(x);
11. calc_paycheck(x[0]);
12. void display_scores
    (int ele[]);
13. `int A[] = { 1, 2, 3 };`
14. An array of integers cannot be displayed in one *cout* statement.
15. A value cannot be assigned to an array identifier.
16. (1) *total* not initialized to 0;
    (2) only list[0] contributes to total
17. Subscript *n* may be out of bounds.
18. `const int MAX = 100;`
    `float distances[MAX];`

Study questions on Topic 8: GUIs and file-maintenance applications

1. Class and application design

1. An array would be an appropriate way to store (a) a set of database tables; (b) one database table; (c) one database record; (d) one database field; (e) one database field schema
6. To assure that all records are different, database designers use (a) a sort field; (b) queries; (c) data analysis (d) searches; (e) a primary key
7. A database normally consists of (a) pixels; (b) tables; (c) keys; (d) protocols; (e) none of these
8. A selection query corresponds to (a) a table; (b) a view; (c) a logical assertion; (d) a set of records; (e) all of these
9. Which of the following is *not* associated with database management? (a) query design; (b) table design; (c) global control of formatting; (d) entities and relationhips; (e) all of these are associated
10. In databases, an object or instance corresponds to a (a) record; (b) table; (c) bit; (d) relation; (e) all of these
11. In databases, an entity or class of objects is implemented by a (a) record; (b) table; (c) bit; (d) relation; (e) all of these
12. To display information from a database, we use a (a) format command; (b) named style; (c) master page; (d) query; (e) all of these
13. Non-duplication of data in tables is enforced by use of (a) formulas; (b) primary keys; (c) formats; (d) protocols; (e) all of these
14. A database table's columns correspond to (a) records; (b) tables; (c) instances; (d) attributes; (e) all of these
15. A use case is a(n) (a) interaction; (b) algorithm; (c) data item; (d) control structure; (e) none of these
16. Use cases are initiated by (a) web sites; (b) applications; (c) users; (d) schedules; (e) none of these
17. Use cases are represented in which language? (a) Java; (b) machine language; (c) HTML; (d) UML; (e) none of these

2. Inheritance and polymorphism

1. The object-oriented feature that allows one class to acquire all the attributes and behaviors of another one is (a) inheritance; (b) containment; (c) composition; (d) data abstraction; (e) pass by reference
2. An ISA or kind-of relationship between two classes is a case of what is called (a) encapsulation; (b) inheritance; (c) message passing; (d) container classes; (e) none of the above
3. Inheritance is a relationship between (a) classes; (b) objects; (c) simple variables; (d) pointers; (e) methods
4. A subclass is (a) a class from which another class inherits; (b) a derived class; (c) a based class; (d) a class declared inside another class
5. (T-F) A derived class may in turn have classes that inherit from it.
6. (T-F) An object inherits from other objects
7. (T-F) A class may have only one descendant.
8. After the declaration *class supervisors: employees { }* (a) a variable named *supervisors* exists; (b) any *supervisors* object will have all the members that an employees object does; (c) any *supervisors* object has an *employees* member; (d) any *employees* object has a *supervisors* member; (e) none of these is true
9. What feature allows us to define a class that automatically has all the members of another class? (a) encapsulation; (b) top-down design; (c) constructors; (d) inheritance; (e) all of these
10. (T-F) Composition and inheritance are two words to describe the same relationship between two classes.
11. Which is *not* a relationship between classes? (a) containment; (b) inheritance; (c) friendship; (d) reference; (e) negation
12. (T-F) A taxonomy in biology is a kind of inheritance hierarchy.
13. (T-F) In a hierarchy of categories of numbers, the category of real numbers inherits from the category of integers.
14. A menus class could reasonably inherit from a _____ class. (a) screen locations; (b) colors; (c) windows; (d) text; (e) application
15. (T-F) A derived class's methods have direct access to the private members of its base class.
16. (T-F) An instance of a subclass occupies less memory than an instance of its superclass.
17. (T-F) An inheritance hierarchy of shapes could have more than two levels.
18. A generic array would be composed of (a) strings; (b) *void* pointers; (c) derived-class objects; (d) base-class objects; (e) *void* methods
19. In object-oriented design, the concept of subcategories could be implemented by (a) encapsulation; (b) reference; (c) analogy; (d) inheritance; (e) polymorphism
20. In declaring a subclass by inheritance in Java, we follow the subclass's name with a _____ and the name

of the base class. (a) period; (b) double colon; (c) semicolon; (d) colon; (e) slash
21. A generic array class implemented by inheritance (a) could store items of any data type; (b) is a descendant of an ordinary array class; (c) could store items allocated statically; (d) inherits the class of the contained objects; (e) none of these
22. The MFC is a(n) _____ hierarchy. (a) module; (b) containment; (c) organization; (d) file; (e) inheritance
23. Inheritance is said to replace certain uses of which keyword? (a) *switch*; (b) *if*; (c) *while*; (d) *void*; (e) *class*
24. (T-F) A base class may have no instances.
25. (T-F) The derived-class constructor runs before a base-class constructor.
26. A derived-class constructor may pass data to a base-class constructor with a (a) method parameter; (b) member initializer; (c) method call; (d) *if* statement; (e) declaration
27. Protected members are accessible to (a) a class's ancestors; (b) a class's descendants; (c) any method; (d) *main*; (e) library methods
28. (T-F) Protected members are accessible to any method.
29. (T-F) Private inheritance makes all base-class members private.
30. (T-F) A derived class may inherit from only one base class.
31. We implement pure polymorphism through (a) ancestor data members; (b) free methods; (c) methods in general; (d) virtual methods; (e) none of these
32. In polymorphism, the behavior of an object depends on whether (a) it is an instance of a base or derived class; (b) the object is statically or dynamically allocated; (c) it is dynamically allocated; (d) it has a pointer member; (e) it has a private method
33. (T-F) With polymorphism, a descendant class may change the behavior of its ancestor class.
34. Pure polymorphism is implemented through _____ methods. (a) global; (b) base-class; (c) virtual; (d) access; (e) *void*
35. (T-F) In a mixed collection of graphical objects of different types, the same *draw* method would be used to display each shape.
36. The address of a call to a virtual method is resolved at (a) compile time; (b) the time a program is loaded; (c) the time the method actually executes; (d) the time the method terminates; (e) the time the program terminates
37. A virtual method is a member of _____ classes. (a) simple and complex; (b) container and contained; (c) base and derived; (d) application and view; (e) model and view
38. A virtual event handler will be called by an object of a _____ application class. (a) base; (b) derived; (c) model; (d) container; (e) view
39. Determination of the call address of a method call at run time rather than compile time is (a) inheritance; (b) encapsulation; (c) early binding; (d) late binding; (e) forward reference
40. An array of _____ makes possible a collection of shapes of mixed classes with polymorphic behavior. (a) integers; (b) method pointers; (c) character pointers; (d) base-class references; (e) none of these
41. (T-F) A data item may be polymorphic.
42. (T-F) Polymorphism is used only in cases where a base class has a single derived class.
43. (T-F) An array of pointers may sensibly point to instances of a variety of data types.
44. Polymorphism is an alternative to the use of (a) *if…else*; (b) *switch*; (c) *while*; (d) *goto*; (e) method calls
45. Virtual methods are used most often in (a) encapsulation; (b) data hiding; (c) memory management; (d) inheritance; (e) polymorphism
46. An abstract base class (a) has no virtual methods; (b) cannot be instantiated; (c) has a definition that may not be overridden; (d) has a null constructor; (e) inherits from a concrete base class

3. GUI construction

1. A mouse click is (a) a data item; (b) an algorithm; (c) an event; (d) a prompt; (e) none of these
2. Events are represented by (a) integers; (b) objects; (c) strings; (d) arrays; (e) none of these
3. In a Java GUI, button presses are detected by a (a) virtual machine; (b) compiler; (c) HTML file; (d) listener object; (e) none of these

4. File-maintenance applications

1. Which is *not* supported by typical file-maintenance applications? (a) create new document; (b) upload web site; (c) enable document update; (d) export to new file format; (e) none of these
2. File I/O in Java requires (a) administrator permission; (b) Internet access; (c) creation of *FileReader* or *PrintWriter* objects; (d) a special kind of compiler; (e) none of these
3. If an attempt is made to open a file that does not exist, (a) the program crashes; (b) an alert dialog is displayed; (c) a file exception is thrown; (d) a file is opened at random; (e) none of these

5. Testing and correctness

1. Testing can (a) fix errors; (b) determine that an application is error-free; (c) prove the existence of

errors; (d) show that an application never hangs; (e) none of these
2. Regression testing finds errors introduced by the ____ process (a) specification; (b) design; (c) coding; (d) maintenance; (e) none of these
3. Integration testing determines whether (a) one subprogram works; (b) separately developed modules work together; (c) a distributed application works;w (d) a networked system works; (e) none of these
4. Postconditions are (a) specifications; (b) design; (c) comments that give the steps taken by a program; (d) follow-up steps in program development; (e) none of these

6. Social and professional issues

1. System integration (a) connects users for job networking; (b) involves data sharing in a large organization; (c) is not part of enterprise computing; (d) is the purpose of most database queries; (e) none of these
2. A major factor in raising ethical and legal issues is (a) malleability of information; (b) accuracy of processing; (c) hard-disk speed; (d) interoperability; (e) none of these
3. A major factor in raising ethical and legal issues is (a) unreliability of communication; (b) accuracy of processing; (c) opportunities for anonymity; (d) interoperability; (e) none of these
4. A major factor in raising ethical and legal issues is (a) unreliability of communication; (b) accuracy of processing; (c) hard-disk speed; (d) cheap copying; (e) none of these
5. A major factor in raising ethical and legal issues is (a) unreliability of communication; (b) ease of human communication; (c) hard-disk speed; (d) cheap copying; (e) none of these
6. The social purpose of awarding intellectual property rights has been (a) to enable maximum profit; (b) to encourage innovation; (c) to discourage sharing; (d) to discover geniuses; (e) none of these
7. Fair use is (a) justice; (b) copying for purposes of comment or research; (c) copying for resale; (d) use of copyrighted data at a fair price; (e) none of these
8. Privacy in the electronic era includes (a) anonymity at all times; (b) control of dissemination of personal information; (c) sufficient time alone; (d) freedom from exposure to undesirable ideas; (e) none of these
9. Privacy issues are raised directly by IT due to the (a) existence of data storage media; (b) existence of digital processing; (c) ease of copying and communication; (d) existence of curiosity; (e) none of these

Answers to study questions on topic 8

[*Correlation with questions to be verified*]

Inheritance

1. a. Inheritance permits reuse of classes.
2. b. The class that is a kind of another class inherits from that class.
3. a. All the members of a base class are automatically members of any derived class.
4. b. A subclass inherits from its superclass, or ancestor.
5. t. There is no limit to the depth of an inheritance hierarchy.
6. f. A class inherits from another class.
7. f. Two or more classes may inherit from the same ancestor.
8. b. In a class declaration, the colon followed by a class name allow the class being declared to inherit from the second class.
9. d. Inheritance lets one class inherit or acquire all the member items of its ancestor class.
10. f. Composition, or containment, is a *part-of* relationship (an instance of one class is part of an instance of another), while inheritance is a *kind-of* relationship (a derived class is a *subclass* of a base class).
11. e. Containment, inheritance, friendship, and reference correspond to *has*, *kind-of, has-access-to,* and *knows-about* relationships.
12. t. For example, mammals inherit the features of animals.
13. f. An integer is a kind of real, but a real number is not a kind of integer.
14. c. A menu can be considered a kind of rectangle drawn on the screen.
15. f. A derived class declared *public* has access to the protected and public members of its base classes, but no derived class has access to private members of its base classes.
16. f. A derived class has all its own members plus all the members of its base classes.
17. t. For example, rectangles are a kind of two-dimensional shape, which are a kind of shape.
18. b. An array of *void* pointers could be used to access dynamically allocated objects of any data type.
19. d. A subclass in an inheritance hierarchy can encapsulate a subcategory in the problem domain.
20. d. Example: *class menus : public windows...*
21. a. Each data item is allocated dynamically; the array is an array of pointers.
22. d. An application programmer inherits, for example, a class from MFC's *CFramWnd*, which in turn inherits from *CWnd*.
23. a. Whereas *switch* might detect the type of an object according to a tag, a derived-class object knows its own type.
24. t. Some classes are *abstract* and are used only for inheritance
25. f. The base-class constructor initializes base-class members first.
26. b. A derived-class constructor may set a base-class member data value by using the base-class name in a member initializer, as in ": employees(I)".
27. b. A base class's *protected* members may be seen from derived-class methods.
28. f. Protected members are accessible only to a class's methods and methods of classes derived from the class.
29. t. A base-class member is inaccessible in a derived class if the inheritance is not qualified as *public*.
30. f. Java supports multiple inheritance.

Polymorphism

1. d. An instance of a derived class may call a member of a base class, which will call a virtual method of the derived class.
2. a. Which virtual method of a certain name is called depends on the class of the calling object.
3. t. If an ancestor class method calls a virtual method that is redefined in a derived class, then the derived class is effectively redefining the ancestor's method.
4. c. A method may be declared *virtual* in the base-class declaration.
5. f. A mixed collection would be a good place to use polymorphism, having a different *draw* method for each class.
6. c. Which method is called depends on the class of the object that calls it.
7. c. The derived-class virtual method is the one that an object of its class will call, even through a call to a base-class method.
8. b. The application programmer defines the event-handling method, which is called by the *run* method of the base application class.
9. d. Virtual methods are subject to late binding, an important mechanism in polymorphism.
10. d. When the pointers to objects of different classes are dereferenced, then each object will call a virtual method that is a member of its own class.
11. f. Polymorphism is a feature of classes, method names, or operators.
12. f. Two derived classes may have different virtual methods called by methods of the same base class.
13. t. If the pointers are *void* and point to instances of classes derived from a single base class, the array may

be scanned and the same virtual method called on each one in turn.

14. b. An object knows its class and in polymorphism need not store a specific member that reports the class.
15. e. Pure polymorphism with late binding is possible due to virtual methods.
16. b. An abstract base class contains at least one virtual method with a null definition, a pure virtual method.

Longer-answer questions

Objective 19: Manage a collection

Study questions on Database

1. What is a database table?
2. What is required for an efficient search of a database?
3. What is database filtering?
4. What do columns in a database table represent?
5. What do rows represent in a database table?
6. What is a database query?
7. Contrast selection queries with projection queries.
8. Give a case where a database table may represent a relationship between two entities.
9. Referring to Entity-Relationship Design, design a database to represent three entities, each entity represented by one table with a primary key. The entities are: customers, products, transaction detail. Detail is one instance of a product purchased, possibly along with other products. Describe the relationships among these entities. Create a small Excel table for each entity.
10. Following the pattern in the previous question, where the third entity represents a joining of one instance of each of the other two entities, design three-table databases for the following:
    (a) job applicants being hired for jobs, so that a sheet can be generated with information about all the new hires for a given period of time;
    (b) poems being published in journal issues, so that a sheet can be generated listing all the poems published for a given journal issue;
    (c) attendees signing up for workshops at a research conference, so that a report can be generated listing all the attendees at a given workshop;
    (d) home buyers visiting open houses, so that a list can be generated of all the home buyers who visited a given house;
    (e) customers buying CDs at music stores, so that a sales slip can be generated;
    (f) CD vendors selling shipments of music to music stores, so that a vendor can list all the shipments made to a certain music store.
11. Why are there:
    (a) database management systems
    (b) database design principles
    (c) primary keys

Study questions on multiple topics

1. One language used for design of interactive systems is (a) HTML; (b) machine language; (c) database query language; (d) UML; (e) none of these
2.

Name _____

D. Keil
63.152: Computer Science I
Framingham State College Fall 2009