Study questions for topic 1 (Computer hardware)

Multiple-choice or T/F:

1. Which instruction below does not change the value in the accumulator? (a) load; (b) store; (c) add; (d) sub

2. Which is not a peripheral? (a) the microprocessor; (b) the monitor; (c) the keyboard; (d) the printer; (e) a disk drive

3. 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

4. Input/output is data that moves between RAM and (a) the program counter; (b) data registers; (c) the instruction register; (d) RAM; (e) peripherals

5. 0011₂ + 0010₂ = (a) 0101₂; (b) 1010₂; (c) 1000₂; (d) 0111₂; (e) 1011₂

6. (T-F) A yes-or-no answer to a question must be stored in at least three bytes of memory.

7. 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

8. Electronic storage composed of silicon chips is (a) RAM; (b) CDROM; (c) hard disk; (d) keyboard; (e) monitor

9. Microprocessors have on them: (a) a disk; (b) a screen; (c) registers; (d) high-level code; (e) documentation

10. (T-F) A microprocessor has its own special assembler language.

11. A language expressed in binary notation is (a) machine language; (b) assembler language; (c) C++; (d) all of the above; (e) none of the above

12. 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

13. A NOT gate is (a) (a) software building block; (b) hardware building block; (c) design tool; (d) algorithm; (e) C++ operator

14. (T-F) A black-box diagram shows both what a component does and how.

15. 3 = (a) 111₂; (b) 11₁₂; (c) 10₂; (d) 11000₂; (e) none of these

16. (T-F) The fetch-execute cycle runs until the user inputs a Quit instruction.

17. A disk drive is a (a) software component; (b) register; (c) peripheral; (d) logic gate; (e) silicon-based storage device

18. (T-F) The operating system runs concurrently as applications execute.

19. 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

20. The hexadecimal system uses base (a) 2; (b) 4; (c) 8; (d) 16; (e) 60

21. 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

22. 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

23. 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

24. (T-F) A bitwise AND operation corresponds to a simple arithmetic procedure taught in grade school.

25. (T-F) The bitwise complement operation (~) corresponds to subtracting a number from zero.

26. 11₁₂ + 1011₂ = (a) 0011₂; (b) 0100₂; (c) 1000₂; (d) 0111₂; (e) 1110₂

27. 1110₂ + 101₂ = (a) 0011₂; (b) 1010₂; (c) 1001₂; (d) 0111₂; (e) 1011₂

28. Machine language is (a) easier to read than C or C++; (b) the language of the executable program; (c) the language of the source code; (d) the compiler’s input; (e) none of these

29. The architecture of a computer is its (a) brand name; (b) application programs; (c) operating system; (d) hardware organization; (e) software design

30. Place values are used in (a) memory; (b) software design; (c) decimal numbers only; (d) binary, decimal, and hexadecimal numerals; (e) assembler programs

31. 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

32. 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

33. 1001₂ + 0011₂ = (a) 1011₂; (b) 0100₂; (c) 0101₂; (d) 1000₂; (e) 0011₂

34. Negative binary values are stored using (a) overflow; (b) accumulator; (c) carry; (d) borrow; (e) a sign bit

35. When a computation produces a value greater than the hardware’s capacity to store it, what is the result? (a) caching; (b) overflow and loss of data; (c) program termination; (d) complementing; (e) compression
36. Floating-point representation adds ____ to what is found in integer storage. (a) a sign bit; (b) an exponent; (c) a binary point; (d) floatation; (e) speed

37. “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

38. A mnemonic (a) specifies a data location; (b) specifies a data value; (c) specifies a constant; (d) specifies an action; (e) is a language

39. (T-F) The language of the microprocessor is assembler.

40. (T-F) The language of a microprocessor is expressed in 0’s and 1’s.

Short-answer:
1. What is a low-level language, associated with a particular microprocessor, that uses binary code to represent instructions and data?

2. Convert 10 (decimal) to binary notation.

3. Convert the value 1001₂ to decimal notation.

4. What information must a microprocessor have about a data item in RAM in order to access it?

5. A circuit’s output, given a certain combination of inputs, is diagrammed by a(n) ________

6. Shifting left a binary number by one bit produces what result?

7. What is a data location that stores intermediate values while calculating?

8. What is a data location that stores the current instruction being executed?

9. What is a data location that stores the address of the next instruction that will execute?

10. Convert to decimal: 1011₂

11. Convert to decimal: 0101₂

12. Convert to decimal: 0110₂

13. Convert to decimal: 1001₂

14. Convert to decimal: 1100₂

15. Convert to binary: 15₁₀

16. Convert to binary: 6₁₀

17. 100₁₆ + 001₁₂ =

18. 011₁₂ + 0100₃ =

19. 1000₂ – 011₀₂ =

20. 011₀₂ – 001₀₂ =

21. What storage technique is used to store negative values?

22. What is the result when a computation produces a value that exceeds the hardware’s storage capacity?

23. What is base-16 notation called?

24. What is a low-level language that uses labels and mnemonics to express programs?

25. Name the loop that a microprocessor is running at all times.

26. Name the part of an assembler-language statement that specifies what action is to be taken.

27. Name the part of an assembler-language statement that specifies the location of the data to be acted upon.

28. Name the part of an assembler-language statement that specifies the name of the statement’s address.

29. How many different values can be represented in 16 bits?

30. What information must a microprocessor have about a data item in RAM in order to access it?

31. Name four peripheral devices.

32. 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?

33. What is the common name for primary storage?

34. 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?

35. How much data is required to store the information as to whether or not a number is greater than zero?

36. Do each of the following number base conversions:
   (a) 23 = ________₂
   (b) 12 = ________₂
   (c) 1011001₂ = ________₁₀
   (d) 10₁₂ = ________₁₀
   (e) 1101₁₁₂ = ________₁₀
   (f) 101011₁₁₂ = ________₁₀
   (g) 35 = ______₂
   (h) 77 = ______₂

37. 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

38. What decimal values are represented by the following stored binary integer values?
   (a) 1000000000011101
   (b) 010000011111000
   (c) 1111111111111111
   (d) 011111110000000

39. 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?

40. What information must a machine-language program have about a data item in a computer in order to access it?

41. How much data is required to store the answer to a yes/no question? __________

42. What range of values can be stored in 16 bits? __________

43. Do these conversions:
   (a) 6 = ______₂
   (b) 100₂ = ______₁₀
   (c) 11₂ = ______₂ (d) 10101₂ = ______₁₀

44. Find the binary results:
   (a) 10₂ + 10₁₂ = ______
   (b) 1011₁₂ + 110₁₂ = ______
   (c) 1000₂ – 111₁₀₂ = ______
45. Express each of these in exponential notation:
   (a) 0.47
   (b) 392.1
46. Write the truth table of the NAND gate:

47. Show result of these bitwise operations:
   (a) $0101_2 \ |\ 1110_2 = \_
   (b) $1010_2 >> 1 = \_
48. 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.

49.  

49. 

49. 

49. 

49. 

49. 

49. 

49. 

In the program above, what is the effect of the line
   a) load fee
   b) sub discount
   c) store bill
   d) fee data 100

Longer answer:
1. Write a program for the model processor’s assembly language that accepts two values and displays the smaller one.
2. Describe the fetch/execute cycle in one or two paragraphs.
3. In what ways do the human brain and a computer resemble each other? Differ?
4. Why is the binary number system appropriate for digital computers?
5. What is the advantage of using a special twos-complement format for negative numbers instead of normal binary format?
Answers to study questions for topic 1

Multiple-choice or T/F:

1. b. The store instruction copies the value in the accumulator to a memory location; the others change the value there.
2. a. The microprocessor is the central component of a computer system; the input/output devices are peripherals.
3. e. A bit is an on-off switch, corresponding to the nature of the hardware.
4. b. A register like an accumulator stores data loaded from memory. The user and peripherals are not on the main circuit board. The program counter is for addresses and the instruction register is for instructions.
5. c. 3 + 2 = 5
6. f. A single bit may store the answer to a yes-or-no question, 0 or 1.
7. b. A data statement begins with a label, which is a name for the memory location that stores a data value.
8. a. Random-access memory is composed of silicon chips to store data.
9. c. Registers are high-speed storage locations on the microprocessor, which is about the size of a silver dollar.
10. 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.
11. a. The language accepted by microprocessors, machine language, consists of 0's and 1's only.
12. d. More than one program can run on a general-purpose computer.
13. b. A NOT gate is a logic component built from transistors.
14. f. A black box hides implementation details.
15. b. 3 = 1 × 2 + 1 × 1 = 11₂
16. f. The fetch-execute cycle runs until a stop instruction is encountered.
17. c. Input/output devices are peripherals.
18. t. The operating system manages resources for the user and for application software.
19. a. A server is a PC that handles data communication in a network and makes applications and data available to network workstations.
20. d. Base-16 notation records four bits per hexadecimal digit.
21. e. The OR gate yields a 0 only if both its inputs are 0.
22. d. The AND gate outputs 1 if both inputs are 1.
23. a. The NOT gate outputs the logical negation of its single input.
24. f. Bitwise AND performs logical conjunction on each bit.
25. f. The bitwise complement of a number is the same number with each bit logically negated, from 1 to 0 or 0 to 1.
26. e. 11 + 1011 = 3 + 11 = 14 = 1110
27. –
28. –
29. –
30. –
31. –
32. –
33. –
34. –
35. –
36. –
37. –
38. –
39. –
40. –

Short-answer:

1. Machine language
2. 1010₂
3. 1001₂ = 9
4. Its address
5. truth table
6. Doubles the number
7. register
8. instruction register
9. program counter
10. 12₁₀
11. 1011₂ = 11₁₀
12. 0101₂ = 5₁₀
Study questions for topic 2 (Program design)

Multiple-choice or T/F:

1. A precise plan to solve a problem in finite time is (a) a module; (b) a function; (c) an algorithm; (d) recursion; (e) none of these
2. (T-F) A recursive definition is one that has just one case to handle.
3. (T-F) At the processor level, a loop entails a backward jump.
4. Which is not a recommended tool for program design? (a) flowcharts; (b) pseudocode; (c) object-oriented analysis; (d) hierarchy charts; (e) use of keywords
5. Which is not a control structure? (a) sequence; (b) branch; (c) loop; (d) the bubble sort; (e) all are control structures
6. (T-F) The loop is a control structure.
7. The code fragment,

```
jump 0
end
print sum
end stop
```

implies a (a) sequence; (b) branch; (c) loop; (d) subroutine
8. (T-F) Syntax is a major factor in program design.
9. 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
10. (T-F) A flowchart using only three different control structures can diagram a solution to any solvable problem.
11. 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
12. The branch is a (a) hardware item; (b) control structure; (c) data structure; (d) flowchart rectangle; (e) module
13. (T-F) The problem-solving process presented in class places design before coding.
14. (T-F) The body of a top-tested loop will always execute at least once.
15. Object-oriented design focuses problem solving on (a) categories of things; (b) processes; (c) functions; (d) integers
16. Successively more detailed development of an algorithm is called (a) object-oriented design; (b) a module hierarchy; (c) bottom-up design; (d) stepwise refinement; (e) problem specification
17. 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
18. 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
19. (T-F) A branch should not be nested inside a loop.
20. (T-F) A component of a structured flowchart has one entrance and one exit.
21. (T-F) The loop is a data structure.
22. (T-F) Structured programming and modular programming are roughly synonyms.

Short-answer:

1. The word otherwise might be appropriate in pseudocode for which control structure?
2. In a module hierarchy, what is the location of a module used directly by module $x$?
3. What is an algorithm?
4. Name the three fundamental control structures recommended in structured programming.
5. In software development, what steps are recommended before coding?
6. Name four chief tools used for writing down a program design before coding.
7. What is the output of the program diagrammed by the flowchart below?

```
Begin
  qly := 2
  qly := qly + 97
  T
  Display qly
  Add 3 to qly
  End
```

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

```
Begin
  Set pnt0 to 0
  Set pnt2 to 0
  Input argument
  argument = product
  Y
  N
  Double product
  Add 1 to pnt
  Display pnt
  End
```

9. What output would be produced by a program based on this pseudo-code, on inputs of 70 and 80?

```
Begin
  Input $a$, $b$
  $a$ := $a$
  $b$ := $b$ ?
  $b$ := $b$ ?
  Add 1 to result
  display result
  End
```

10. What kind of programming stresses use of three control structures and modular decomposition?
11. What kind of programming stresses defining categories of things and their behaviors?
12. The word *repeat* might be appropriate in pseudocode for which control structure?

13. Counting from 5 to 15 would require which control structure?

14. What is the computer-science term for a precise plan to solve a problem in a finite amount of time?

15. What is pseudocode?


17. In the software development process, what steps are recommended before coding a program? After coding?

18. 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?

19. How many times will a counter-controlled loop iterate? A sentinel-controlled loop?

20. Name two variants of the loop control structure.

21. Consider the flowchart below. For each of the input pairs (A,B), shown below, show the resulting output.

22. Put these phases or sub-phases of the problem-solving process in chronological order, number the first “1”, the second “2”, etc.

23. Consider the flowchart below. For each of the input pairs (A,B), shown below, show the resulting output.

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

   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.

**Longer-answer questions:**

1. Is the pseudocode below an example of a structured design? Why or why not?

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

3. 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

4. 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:) \[ g(a,b) = \begin{cases} 
   0 & \text{if } a = 0 \\
   b + g(a/2, 2b) & \text{if } a \text{ is odd} \\
   g(a/2, 2b) & \text{if } a \text{ is even} 
   \end{cases} \]

5. 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:**

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>749.95</td>
</tr>
<tr>
<td>20</td>
<td>199</td>
</tr>
<tr>
<td>10</td>
<td>190</td>
</tr>
</tbody>
</table>

6. 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

7. 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:**

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>-38</td>
<td>300</td>
</tr>
<tr>
<td>300</td>
<td>77</td>
</tr>
<tr>
<td>1300</td>
<td>300</td>
</tr>
</tbody>
</table>

8. 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 \( A - B \)”.

9. 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.

10. 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.

11. 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.)

12. (Challenge) Design a program to input two non-negative integers \((A, B)\) and a positive integer (C). Compute \( A \times B \div C \) and 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?

13. (Challenge) 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.

14. 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 

\[ 7 + 21 + 63 + 189 + 567 = 847. \]

15. 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 \( n \).

16. (Challenge) 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.

17. Write pseudocode to compute the base-2 integer logarithm of an input integer. (See flowchart below.)

![Flowchart]
18. 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.

19. 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’.

20. 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.
Answers to study questions for topic 2

Multiple-choice or T/F:

1. c. An algorithm must be specific, detailed, and not take forever.
2. f. A recursive definition, like "My ancestors are my parents and their ancestors," involves a base case ("my parents") and a recursive case ("their ancestors").
3. t. To repeat an instruction the microprocessor must jump to an earlier program instruction.
4. e. Stream output is a tool in coding C++ programs.
5. d. The bubble sort is a full algorithm, composed of several control structures.
6. t. Iteration, or repetition, is the loop control structure.
7. b. A jump forward implements the branch control structure.
8. f. Program design is independent of languages and their grammar rules.
9. d. Object-oriented design looks at things that we deal with and how they behave.
10. t. The three control structures of structured programming are the sequence, the branch, and the loop.
11. d. Pseudocode is considered useful in program design.
12. b. The branch is the selection control structure.
13. t. Program coding is based on a thought-out design.
14. f. If the test fails the first time, the body will not execute.
15. a. An object is defined by attributes and behaviors.
16. d. The design may be refined or improved step by step.
17. c. A multiway branch corresponds to a series of cascading flowchart diamonds.
18. c. Modular decomposition is a divide-and-conquer strategy for problem solving.
19. f. Any control structure may be nested inside another.
20. t. There should be one way into and one way out of any part of an algorithm.
21. f. The loop is a control structure.

Short-answer:

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 steps
4. Sequence, branch, loop
5. 1. Problem analysis or specification; 2. design
6. Pseudocode; flowcharts; module hierarchy diagrams
7. 2 5 8
8. 5432154321
9. Tall
10. structured programming
11. object-oriented programming
12. loop
13. loop
14. algorithm
Study questions for topic 3 (C and C++ basics)

Multiple-choice or T/F:

1. (T-F) The compiler ignores white space
2. (T-F) C++ has two kinds of statement: declarations and executable statements.
3. (T-F) C++ is higher-level than assembler language.
4. (T-F) Semantics deals with structure, syntax with meaning;
5. The #include statement (a) treats a file as if it were part of the source file to be compiled; (b) opens a file for input; (c) opens a file for output; (d) is required for all type declarations
6. (T-F) A library is a kind of program documentation.
7. << is (a) an input operator; (b) an output operator; (c) the assignment operator; (d) a relational operator
8. (T-F) C++ programs use streams to handle input and output.
9. cout is (a) a function; (b) a keyword; (c) a data item; (d) stream input; (e) none of these
10. 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
11. (T-F) A machine-language instruction usually performs a more complex task than a C++ statement.
12. The compiler produces (a) high-level code; (b) machine code; (c) documentation; (d) keyboard input
13. C and C++ syntax requires that every program have: (a) a main function; (b) a variable; (c) input; (d) output; (e) comments
14. How many reserved words does C++ have? (a) none; (b) a few; (c) dozens; (d) hundreds; (e) thousands
15. (T-F) C++ is an extension of C.
16. (T-F) Any C++ program is nearly certain to compile on a C compiler.
17. (T-F) The compiler translates from machine language to C++.
18. (T-F) if is a valid statement in C and C++.
19. A variable does not have a (a) name; (b) address; (c) type; (d) precedence
20. To document our source code we are encouraged to use (a) short variable names; (b) comments; (c) operators; (d) manuals
21. A global data item that might be of value in documenting code is a (a) named constant; (b) comment; (c) literal; (d) variable
22. Syntax is (a) documentation; (b) meaning; (c) grammar rules; (d) good-programming guidelines; (e) recursion
23. The C or C++ 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
24. (T-F) A variable is a memory location with a name and a data type.
25. (T-F) The compiler treats int the same as INT.
26. (T-F) An identifier may start with a digit.
27. Which is not a valid identifier: (a) in1; (b) my_function; (c) 3p2; (d) t4
28. Which is not a valid identifier: (a) xyz; (b) my_variable; (c) xi2345; (d) FooBar
29. Machine language is (a) easier to understand than C or C++; (b) the language of the executable program; (c) the language of the source code; (d) what the compiler starts with; (e) none of these
30. (T-F) Identifiers usually stand for memory locations.
31. Variable declarations in a list following a type name are separated by (a) periods; (b) commas; (c) semicolons; (d) colons; (e) parentheses
32. (T-F) The same kind of program elements can be on the left and right side of "=".
33. const int x = 3; is (a) a syntax error; (b) an assignment statement; (c) the declaration of a named constant; (d) an output statement
34. For input in C++, we use (a) the assignment operator; (b) data statements; (c) read(); (d) cin; (e) cout
35. 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
36. (T-F) The compiler reads machine code and outputs high-level code.
37. (T-F) A data type is a memory location with space allocated for a value.
38. 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
39. Which is not a valid C/C++ identifier: (a) 2days; (b) myvar; (c) Data12; (d) FooBar; (e) x
40. (T-F) I/O takes the form of streams.

Short-answer:

1. Write a C or C++ statement that declares an integer variable, sum, and initializes it with the sum of x and 2.
2. Write a C++ statement that prompts for an integer variable, quantity, with appropriate labelling.
3. What may the programmer do in a C or C++ program to make sure his or her intention is clear?
4. Name three ways to give a value to a variable in C or C++.
5. Write a C or C++-language statement that displays the label “The quantity is” and the value of the integer variable, quantity.
6. Give two reasons for using named constants in a program.
7. How does a variable differ from a literal?
8. How is C++ like English? How is it different?
9. What should every program have that the compiler will ignore?
10. Write a C++ statement to prompt the user for a quantity of items.
11. Name a C++ grammar rule that is recursive.
12. What is wrong with this definition of a named constant?
   ```c++
   const int YEAR;
   ```
13. Write a C/C++ statement that declares a variable that will store the number of people in a team.
14. What is wrong with this code?
   ```c++
   int x;
   cin >> "Enter an integer: ";
   >> x;
   ```
15. What is risky about this code?
   ```c++
   const int MINUTES = 180;
   int num_persons, share;
   cout << "How many persons? ";
   cin >> num_persons;
   share = MINUTES / num_persons;
   ```
16. What is the output of this code?
   ```c++
   int x1, x2 3;
   cout << 3 * x1;
   ```
17. Write three C++ statements to declare, prompt for, and input an integer variable that expresses the number of years in the term of a contract.
18. Name two different forms of documentation that you can write in your source code to make the program easier to understand.
19. Which of the items below are valid C or C++ identifiers?
   (a) A12; (b) cat; (c) 39G; (d) output;
   (e) A3D; (f) employee-salary;
   (g) employee_salary
20. Will these two lines of source code compile?
   ```c++
   cout << "This is a long sentence continued on a second line";
   If not, what message is generated? Correct the error.
   ```
21. Declare a named constant to represent the number of days in a week.
22. Correct the error in this code.
   ```c++
   int quantity, price;
   cout << "Enter quantity <" and price; 
   cin >> quantity, price;
   cout << "Amount due: "
   << quantity * price << endl;
   ```
23. Label each term below with the letter of its appropriate definition:
   - _____ assignment statement
   - _____ compiler
   - _____ compound statement
   - _____ documentation
   - _____ fourth-generation language
   - _____ function definition
   - _____ low-level language
   - _____ keyword
   - _____ identifier
   - _____ named constant
   - _____ stream
   - _____ third-generation language
   - _____ syntax
   - _____ statement
   - _____ source code
   - _____ variable declaration
   - _____ white space
24. Longer answer:
   1. Write a well-documented C or C++ program that prompts for a unit price of an item and a quantity, and displays the total price to be charged.
   2. Write a C++ program that accepts three weights and displays their average.
   3. Write a well-documented C or C++ program that prompts for three quantities and displays their sum.
4. Write a C++ 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.)

5. Consider the program:
```cpp
#include <iostream.h>

void main()
{
    cout << "Enter 2 #s" << "to add: " ;
    int input1, input2, sum;
    cin >> input1;
    cin >> input2;
    sum = input1 + input2;
    cout << 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?

6. Modify program hello.cpp, below, so that the output of the cout statement goes to a text file called hello.out on your work disk.
```cpp
#include <iostream.h>

void main() {
    cout << "Hello";
}
```

7. Write, compile, and run a C++ 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.

8. Will the two-line statement:
```cpp
cout << "This continues on a second line";
```
compile? If not, what message is generated? Correct the error and explain the general rule that applies.

9. Debug this code:
```cpp
int quantity, price;
cout << "Enter quantity " << "and price: ";
cin >> quantity, price;
cout << "Amount due: " << quantity * price << endl;
```

10. Debug this code, briefly explaining the syntax error:
```cpp
int top&bottom = 3;
cout << "top&bottom = " << top&bottom << endl;
```

11. Write a well-documented C++ 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 study questions for topic 3

Multiple-choice or T/F:

1. t. The use of spaces, tab characters and newline characters generally has no effect on program compilation.
2. t. Executable statements include assignments and function calls; declarations include variable declarations and type definitions.
3. t. The vocabulary of C and C++ contains powerful statements like if and while not duplicated in assembler language.
4. f. Semantics deals with meaning, syntax with structure.
5. a. The #include causes the named file to be read and compiled with the rest of the file to be compiled.
6. f. A library is a coding resource, often precompiled.
7. b. The insertion operator << is used with cout and other output streams to insert characters into the data stream.
8. t. Examples: cout, cin.
9. c. cout is an instance of an output stream type.
10. d. All low-level input and output routines, for example, are put into the executable program file by the compiler as part of a runtime package.
11. f. A C or C++ assignment, for example, executes at least a load and a store instruction.
12. b. A compiler translates from a language like C++ to an assembler or machine language of a particular machine.
13. a. Variables, input, and output are nearly universal but optional. The function called main is mandatory.
14. c. C++ as defined by Stroustrup has 48 reserved words.
15. t. C++ contains all the syntax of C and some additional rules.
16. f. The converse is true, but many C++ programs contain statements that a C compiler would reject, or even comments, such as those beginning "//".
17. f. The compiler translates from the higher-level language to machine language.
18. t. The braces enclose a compound statement; the ones above enclose the null statement.
19. d. Variables have identifiers, memory locations, and data types.
20. b. Comments should be used to clarify a programmer’s intention.
21. a. A named constant documents itself.
22. c. Syntax rules specify the proper order of language elements in a program.
23. a. The assignment statement copies a value but leaves it intact at its source.
24. t. A variable is a cell or series of cells that may be assigned a value. To know how much space to assign the variable, the compiler must know its data type.
25. f. The C++ compiler is case sensitive.
26. f. An identifier must start with a letter or an underscore.
27. c. An identifier may not begin with a digit.
28. b. An identifier may not contain a space.
29. b. Executable machine code is in binary form.
30. t. For example, a variable is a name for a data location in RAM.
31. b. Example: int a,b,sum;
32. f. Any expression may be on the right, but only variable names may be on the left.
33. c. A named constant is the declaration and initialization of a variable whose value may not be changed.
34. d. The stream object, cin, permits input from the keyboard; cout permits screen output.
35. d. For a program to be maintainable, to fix errors and add features, the programmer must be able to understand it.
36. f. The compiler reads high-level code and outputs machine code.
37. a. A data type specifies the meaning of a variable; a variable is a data item with a memory location and a value.
38. b. The compiler will be unable to compile a program that has syntax errors.

Short-answer:

1. int sum = x + 5;
2. cout << “Enter quantity: “; cin >> quantity;
3. Write comments or use meaningful identifiers
4. 1. Initialize in declaration
2. Assign with assignment operator (=)
3. Input with cin or scanf
5. cout << “The quantity is “ << quantity << endl; printf(“The quantity is %d”, quantity);
6. (1) To document meaning of a value used; (2) To permit updating in one step a constant used multiple times
7. A variable has an address and its value can change.
8. Like: Has grammar rules and vocabulary
   Different: English has more rules, is ambiguous
9. Comments, white space
10. int quantity;
    cout << “Quantity: “; cin >> quantity;
11.
12.
13.
14.
15.
16.
17. cout << “Number of years: “; int num_years;
    cin >> num_years;
18.
Study questions for topic 4a (Numeric data)

Multiple-choice or T/F:

1. (T-F) The compiler performs multiplication before addition in the absence of parentheses.
2. * is (a) a sign; (b) a string; (c) a function; (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) assigned a value; (b) input; (c) processed; (d) used in an expression
5. Before use in a statement, a variable 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 C and C++, \( 15 / 12 = (a) 0; (b) 1; (c) 1.25; (d) 12; (e) 15 \)
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. Convert to C or C++:
   \[ \frac{b + c}{d + e} \]
   \( a; (b) abcd; (c) (b+c)/(d+e); (d) b+c/d+e; (e) b/d+c/e \)
11. (T-F) An item of type int occupies fewer bits than a char item.
12. A category that defines the meaning or interpretation of a pattern of bits is (a) an algorithm; (b) a register; (c) a function; (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 data type, of the following? (a) short int; (b) int; (c) unsigned int; (d) long int; (e) none of these
20. What is the largest capacity data type, of the following? (a) short int; (b) int; (c) unsigned int; (d) long int; (e) none of these
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 can be between 0 and 1.
27. (T-F) Floating-point storage entails representationsal error.
28. (T-F) Since \((6 - 3 - 1)\) is 2 rather than 4, therefore we know that the subtraction operator is right associative.
29. A mantissa, or fraction, component appears in type (a) double; (b) int; (c) char; (d) unsigned; (e) string
30. (T-F) Floating-point storage uses an organization concept similar to scientific notation.
31. What is necessarily double about the type double? (a) magnitude; (b) storage; (c) precision; (d) meaning; (e) sign
32. Standard functions abs, sin, exp, and sqrt are defined in the library file (a) iostream.h; (b) ctype.h; (c) stdio.h; (d) math.h; (e) stdlib.h
33. Type casting in C/C++ (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
34. To format output, we can use (a) a math library; (b) iostream.h; (c) manipulators; (d) cin; (e) >>
35. An item of type char occupies how many bits? (a) 0; (b) 1; (c) 4; (d) 8; (e) 16
36. With what special tool can you explicitly specify the width of part of your output? (a) cout; (b) cin; (c) setw(); (d) ios; (e) setiosflags()
37. What is the data type of 3.14? (a) char; (b) int; (c) double; (d) char[]; (e) void
38. What is the data type of 2.0? (a) char; (b) int; (c) double; (d) char[]; (e) void
39. To convert an int value to a float in C++ (a) is possible with just an assignment; (b) requires function calls; (c) requires writing an algorithm; (d) is not possible
40. If \( PI \) is 3.1416, and \( n \) is declared as an integer, then \( n = 2 * PI; \) \( cout << n << endl; \)
41. (T-F) If \( x \) is an integer variable and \( y \) is a double, then "\( x = y; \)" will compile.

Short-answer:

1. Evaluate the C/C++ expression, \( 1 + 7 * 4 + 15 / 3 \).
2. What is the output of this code? \( \text{int } n = 25.9; \text{ cout } << n << \text{ endl; } \)
3. Write a statement in C++ that will declare and assign the value 70.4 to a variable named, \( \text{height. } \)
4. What is the relationship between a data type and a variable?
5. Write an expression that converts the value 8.42 to int and stores the result in an int variable, n.
6. Write a declaration for a variable, cost, that can store a fractional value.
7. Name two data types whose instances can store fractional values.
8. Name two data types whose instances can store whole-number values.
9. List the the arithmetic operators in order of precedence.
10. Write an expression whose value is the original value
    increments a variable n by 1 and whose value is the incremented value.
11. Write an expression that decrements a variable n by 1 and whose value is the decremented value.
12. What header file is used to access the functions abs, sqrt and sin?
13. Write a C/C++ expression equivalent to 5^3
14. Write a shorter version of this statement: quantity = quantity + input;
15. What operator assigns to its left operand the product of its left and right operands?
16. What operator assigns to its left operand the sum of its left and right operands?
17. What operator assigns to its left operand the difference between its left and right operands?
18. What operator assigns to its left operand the quotient of its left and right operands?
19. Write an expression that increments a variable n by 1 and whose value is the original value of the variable.
20. Write an expression that decrements a variable n by 1 and whose value is the original value of the variable.
21. Write an expression that increments a variable n by 1 and whose value is the incremented value.
22. Write an expression that decrements a variable n by 1 and whose value is the decremented value.
23. Name an integer data type that may store a larger range of values than an int.
24. Name an integer data type that may store a smaller range of values than an int.
25. Name a numeric data type that stores only values greater than or equal to 0.
26. Evaluate: 2 – 6 / 2 + 4
27. What standard library defines the sqrt function?
28. What is the common term for coercion of data types?
29. Write an expression that type casts the double variable amt to type int.
30. When a value assigned to a variable is too large for the variable’s storage capacity, it is a case of __________.
31. What is the output of:
   int a = 12.8;
   cout << a;
   
32. What is the output of:
   int a = 9.6;
   cout << a;
33. What is the standard library used in formatting numeric output?
34. List from smaller to larger (in storage space occupied) the following types: int; float; short; unsigned; double; char.
35. Write an appropriate constant declaration allowing you to replace the numeral 0.05 in the following C++ statement with a meaningful identifier.
   tax = 0.05 * sales amt;
36. Will the following program compile? If not, what error message will you get?
   void main()
   {
      const int num_bldgs = 4;
      num_bldg = 5;
   }
37. Evaluate:
   sqrt (4)____
   exp(0)____
   pow(3,2)____
   abs (–60)____
   log(10000)____
   log(0.1)____
   pow(2,3)____
   fabs(30.33)____
   abs (–20.33)____
38. 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?
39. These declarations will be used to compute the area of a circle from an input radius value:
   float a,r;
40. Rewrite the declarations, using meaningful identifiers.
41. Is a real number a kind of rational number? Explain with reference to the Venn diagram in Section 3.2.
42. 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?
43. What is the risk of using cin to input data to int or float variables?
44. What is the difference among the C++ constants ‘5’, “5”, 5, and 5.0?
45. Write a cout statement that displays the float variable length, of value 2.54, as “2.5”.
46. Write a cout statement that displays the total price of 127 items, at $0.326 apiece, rounded to the nearest cent.
47. What is wrong with these named-constant definitions?
   (a) const float LB_PER_OZ = 1/16;
   (b) const TAX_RATE = 0.05;
48. What is the value of pounds after these two lines execute?
   (Careful…)
   int ounces = 8;
   float pounds = ounces / 16;
49. Using the C/C++ precedence rules, evaluate each of the following expressions:

50. (a) 12 / 4 – 2 ________
    (b) 3 + 4 * 5 ________
    (c) 7 % 3 + 1 ________
    (d) 20 / (4 + 3 * -2) * 3 ________

51. Problem: 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 C or C++ 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.

52. What are the outputs of each of these statements?

(a) cout << 1 / 3;  (b) cout << (8 / 2) / 3;
(c) cout << (8 / 2.0) / 3;  (d) cout << 26 % 4;
(e) cout << 7.5 / 2.5;  (f) cout << 2.5 * 4;

53. Write an appropriate constant declaration that would allow you to replace the numeral 0.05 (for sales tax) in the following C++ statement with a meaningful identifier.

tax_amt = 0.05 * sales_amt;

54. Write a C or C++ statement that calculates and displays the corresponding integer y, for any integer input value x, applying the formula:

\[ y = \frac{3x^2 - 2x + 5}{4} \]

55. Using type casting, write a statement that assigns the value 2/3 (the mathematical quotient of 2 and 3) to a double variable.

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

___ initialization
___ endl
___ float
___ type conversion
___ iomanip.h
___ setw
___ manipulator

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 C++ 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 functions.
i) Standard function that determines field width for output.
j) Any of a set of standard functions that helps format floating-point output.

**Longer answer problems:**

1. Write a C or C++ 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.

2. 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 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.

   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

8. 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

9. 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 = \( \pi \times \text{radius}^2 \).

   Display the area correct to 2 decimal places.

   Sample I/O:
   Enter radius: 10.2
   Area: 326.9

10. Write a program that computes the radius of the circle whose area is input from the keyboard. Use the \( M\_PI \) constant and \( \sqrt{x} \) function. Display the radius correct to 2 decimal places.

11. 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.

12. 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 \( \sin \) function for one of the equal angles; the \( \sin \) function returns the ratio of the length of the leg opposite the angle to the length of the hypotenuse. The (angle) argument for the \( \sin \) function must be expressed in radians. One degree equals (\( \pi / 180 \)) radians.

13. Find the base-2 logarithm of any input value, using the \( \log_2 \) function and the fact that for any value \( A \), \( \log_2(A) = \log_2(A) \times \log_2(A) \). It may help you to know that the \( C++ \) \( \log \) function returns \( \log_e \) of its argument, where \( e \) is approximately 2.71828.

14. 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

   \[
   P = 26 \times 26 \times 26 \times 26 \times 26 \times 26 \times 26 \times 26
   \]

   where \( P \) is a declared float variable, give the same result? If not, which is correct? Can you explain?

15. 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.

16. 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
17. Debug this code, whose output is 0 on input of 2 lb., 3 oz.:
   ```cpp
   const KILOS_PER_OUNCE = 0.0283495;
   cout << "Enter pounds and ounces: ";
   float pounds, ounces;
   cin >> pounds >> ounces;
   float total_ounces = 16 * pounds + ounces;
   float kilos = total_ounces * KILOS_PER_OUNCE;
   cout << "Kilograms: " << kilos << endl;
   ```

18. Show, through compiler error messages or successful compilations, what the syntax rules of C++ are regarding the order in which the type qualifiers `const`, `unsigned`, `short`, `long`, and `double` may appear with the types `int` and `float`.

19. Write a C++ 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.

20. 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 C++ 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.

21. 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:
   
<table>
<thead>
<tr>
<th>n</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>2^n</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>16</td>
<td>32</td>
<td>64</td>
<td>128</td>
<td>256</td>
</tr>
</tbody>
</table>

22. Write a C or C++ 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
Answers to study questions for topic 4a

Multiple-choice or T/F:
1. t. The order of precedence of arithmetic operators is: unary negation, multiplication/division, addition/subtraction.
2. d. Among other things, * is the multiplication operator.
3. a. An assignment loads data into a register, then copies it to a named memory location.
4. a. Outputting an unassigned variable will yield unpredictable results.
5. b. Syntax requires declaration of all variables, but no other
6. d. The multiplication operator is applied before the addition operator.
7. b. Integer division yields an integer quotient.
8. d. The modulo operator yields the remainder in the division of the first operand by the second.
9. c. The modulo operator yields the remainder in the division of the first operand by the second.
10. c. Parentheses are necessary to override priority of operations given to division over addition
11. f. A char occupies 8 bits (one byte); an int is usually 16 or 32 bits.
12. d. A variable is an instance of a data type; the data type is a category or description.
13. c. The modulo operator % produces the remainder of an integer division.
14. f. Unary operators are one-operand operators.
15. d. The extended assignment operator for addition, +=, adds a value and produces a value.
16. c. The post or pre-increment operator, ++, adds 1 to a variable.
17. c. An expression formed with the pre-increment operator is evaluated after the incrementing.
18. c. An expression formed with the post-increment operator is evaluated before the incrementing.
19. a. Type short never takes more storage than type int.
20. d. Type long never takes less storage than type int.
21. b. Floating-point data is stored in a format with a sign bit, a mantissa (fraction), and an exponent.
22. t. The type int is signed.
23. f. An unsigned integer is always greater than or equal to 0.
24. t. Both floating-point types, float and double, have a sign bit.
25. t. Fractional values are managed by float and double.
26. f. Type int stores whole numbers only.
27. t. For values like 1/3, round-off is necessary in floating-point storage.
28. f. Left associative operators are applied from left to right.
29. a. Floating-point storage includes a binary fraction, which is multiplied by two raised to the power of the exponent component of the stored value.
30. t. Scientific notation, like floating-point storage, uses a fraction and an exponent.
31. c. The type double is double-precision floating point; that is, the mantissa may have twice as many bits as a float’s mantissa.
32. d. The C/C++ mathematics library is in header file math.h.
33. b. Since integer division produces an integer result, it is necessary to type cast an operand to a floating-point type in order to produce a fraction, as in (float)2 / 3.
34. c. The manipulators defined iniomanip.h include setprecision, setw, and setiosflags.
35. d. char takes up one byte and may have any of 256 different values.
36. c. The setw stream function accepts a field width as a parameter and may be used to the left of a data item in a stream output statement.
37. c. A numeral with a decimal point is considered to be a floating-point value.
38. c. A numeral with a decimal point is a floating-point value in C++, even if it happens also to be an integer mathematically.
39. a. For example, a floating-point number assigned to an integer is truncated by the compiler.
40. c. A real number assigned to an integer variable is converted to int.
41. t. Numeric type conversion is automatic in C++.

Short-answer:
1. \(1 + (7 \times 4) + (15 / 3) = 1 + 28 + 5 = 34\)
2. 25
3. `scanf("%f", &height); cin >> height;`
4. A variable is an instance of a data type.
5. \(\text{int } n = 8.42;\)
6. float cost
7. float; double
8. int; char; long int; short int
9. \(/, \%, *, +, -\)
10. 7 \% 2
11. math.h
12. 5 * x * x * x
13. quantity += input;
14. *= assigns to its left operand the product of its left and right operands?
15. += assigns to its left operand the sum of its left and right operands?
16. -= assigns to its left operand the difference between its left and right operands?
17. /= assigns to its left operand the quotient of its left and right operands?
18. `%=` assigns to its left operand the remainder when its left operand is divided by its right?
19. `n++` increments `n` by 1 and its value is the original value of `n`.
20. `n--` decrements `n` by 1 and its value is the original value of `n`.
21. `++n` increments `n` by 1 and its value is the incremented value.
22. `−−n` decrements `n` by 1 and its value is the decremented value.
23. Type `long` may store a larger range of values than an `int`.
24. Type `short` may store a smaller range of values than an `int`.
25. `unsigned int` stores only values greater than or equal to 0.
26. $2 - 6 / 2 + 4 = 3$
27. The standard library `math.h` defines the `sqrt` function.
28. Type casting is the common term for coercion of data types?
29. `(int)amt` type casts the `double` variable `amt` to type `int`.
30. When a value assigned to a variable is too large for the variable’s storage capacity, it is a case of overflow.
31. 12
32. 9.6
33. The standard library used in formatting numeric output is `iomanip.h`. 

Study questions for topic 4b (Character and string data)

Multiple-choice or T/F:

1. (T-F) "cout << 'a';" displays the characters 'a' and 'n'.
2. (T-F) "cin >> x;" will be valid for x of any of several data types.
3. 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
4. What is the data type of 'n'? (a) char; (b) int; (c) float; (d) char[]; (e) void
5. What is the data type of 4? (a) char; (b) int; (c) float; (d) char[]; (e) void
6. What is the data type of '\n'? (a) char; (b) int; (c) float; (d) char[]; (e) void
7. What is the data type of "Hello"? (a) char; (b) int; (c) float; (d) char[]; (e) void
8. char is (a) a keyword; (b) a user-defined identifier; (c) an expression; (d) a function; (e) an operator
9. How is the null character expressed in C and C++? (a) "0"; (b) 'n'; (c) '0'; (d) endl; (e) none of these
10. Which data type is used to perform file input? (a) float; (b) ostream; (c) ifstream; (d) char; (e) user-defined types, usually
11. (T-F) The function eof() is used to close a file.
12. To prepare a file for input, we use (a) cout; (b) cin; (c) open(); (d) the assignment operator; (e) None of these
13. In the standard C input and output functions, the first parameter is (a) the first value to be input or output; (b) a format string; (c) a variable; (d) an address; (e) none of these
14. (T-F) Every printable character has a numeric encoding from 0 to 127.
15. (T-F) A decimal digit may be expressed as a value of type char.
16. The newline character is (a) 'n'; (b) 'n'; (c) 't'; (d) '0'; (e) 'r'
17. The null character is (a) 'n'; (b) 'n'; (c) 't'; (d) '0'; (e) 'r'
18. (T-F) The character '0' has ASCII code 0.
19. The standard library for many functions that manipulate characters, such as toupper, is (a) iostream.h; (b) stdlib.h; (c) toupper.h; (d) math.h; (e) ctype.h
20. (T-F) The toupper function is defined in ctype.h.
21. Which is not defined in ctype.h? (a) printf; (b) tolower; (c) isdigit; (d) isalpha; (e) toupper
22. (T-F) The toupper function returns a true/false value.
23. (T-F) The isdigit function returns a true/false value.
24. Character literals are expressed in (a) capital letters; (b) italics; (c) single quotes; (d) double quotes; (e) parentheses
25. (T-F) To find the ASCII code of a character, typecast the character to int and display it.
26. (T-F) To find the character that corresponds to an ASCII code, typecast the code to int.
27. (T-F) A '0' terminates a C-style string.
28. (T-F) A '0' terminates a C-style string.
29. char s[40]; declares a (a) character variable; (b) character constant; (c) string variable; (d) string constant; (e) string literal
30. The standard library for C-style strings is (a) stdlib.h; (b) stdio.h; (c) string.h; (d) string; (e) string.h
31. (T-F) For C-style string variable s, the following statement produces a 4-character string: s = "tree".
32. (T-F) The first character of C-style string variable s is s[0].
33. (T-F) The first character of C-style string variable s is s[1].
34. (T-F) strlen(s) returns the number of characters most recently copied into s.
35. (T-F) strcpy(x1,x2) copies the contents of x2 to x1.
36. (T-F) strcpy(x1,x2) copies the contents of x1 to x2.
37. (T-F) strcat(x1,x2) copies the contents of x2 to x1.
38. (T-F) strcat(x1,x2) concatenates the string x2 to the end of x1.
39. (T-F) To use the assignment operator to change the value of a C-style string, we must work with one character at a time.
40. (T-F) >> can input just one character.
41. (T-F) An instance of class string can be given a value with the assignment operator.
42. (T-F) A C-style string can be given a value with the assignment operator.
43. What operator concatenates C++ strings? (a) +; (b) −; (c) =; (d) cat; (e) none
44. How many bytes in memory are occupied by C-style string constant “m”? (a) 0; (b) 1; (c) 2; (d) 80; (e) an indeterminate number
45. A stream is a(n) (a) algorithm; (b) device; (c) data item; (d) function; (e) simple data type
46. cin and cout are (a) integers; (b) functions; (c) strings; (d) streams; (e) disks
47. (T-F) A string may be the source or destination for stream I/O.
48. (T-F) A file stream object may replace cin for input.
49. How is the newline character expressed in C and C++? (a) "0"; (b) 'n'; (c) '0'; (d) endl; (e) none of these

Short-answer:

1. Write a declaration for a variable of the type most appropriate to store the value '9'.
2. How are the newline, tab, and null character constants expressed in C++?
3. Write a C language statement that waits for the user to enter an integer and stores the result in an integer variable, input.
4. Name the data type of ‘4’ when it appears in a program.
5. Name the data type of “4.5” when it appears in a program.
6. What char value is found in every C-style string?
7. Write a statement to assign the value “day” to a pre-existing C-style string variable, shift.
8. What is the header file used to access the functions strlen, strcpy, and strcat?
9. Declare a character variable and initialize it with the null character.
10. What is the character constant for the newline?
11. What is the name of the table that specifies the numeric encodings of characters?
12. Name four functions defined in the standard library header file string.h.
13. What operation occurs when + or += are used with C++ strings?
14. Name a C++ standard data type used to store sequences of characters.
15. Name the standard header file that defines the C++ class, string.
16. What is a stream?
17. What is a sequence of characters coming from or going to a device?
18. Name a data type for an object used to open and read a file.
19. Name the standard header file used when opening disk files.
20. What is the standard C-language function for reading keyboard input?
21. Write a statement in C to read user input into an integer variable, n.
22. Write a statement in C to read user input into a float variable, n.
23. Write a statement in C to read user input into a character variable, c.
24. Write a statement in C to read user input into a string variable, s.
25. Name the standard C-language header file for input/output.
26. Write a statement in C to display a string, name, with the label, “Name: “ in front of it.
27. Use the ASCII table to show the output generated by the C++ statement:
   cout << char (123) << char(65) << char(62) << char(65) << char(125);  
28. Show the 8-bit binary value that is stored when a user types R [Enter] in response to the following C++ statement, where ch is of type char: cin >> ch;  
29. What is the data type of the parameters to toupper and tolower? What is the data type of the values these functions return?
30. How are the newline, tab, and null-character constants expressed in C++?
31. How can you convert an integer to a character value? A character to an integer?
32. What are an advantage and a disadvantage of C++ ’s automatic type conversion?
33. 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  
34. Declare and initialize a variable, letter, to store the character value ‘Q’.
35. Declare and initialize a variable, team, to store the string value “Rams”.
36. Declare a string variable, town, with a storage capacity of 40 bytes.
37. What is the data type and memory allocation, in bits and bytes, of a single component of a string data item?
38. Declare the integer variables height and weight; initialize them to 67 and 125, respectively; and write a C-style (not cout) output statement to display their values, with appropriate labels. Name the four standard identifiers associated with the C and C++ 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 function (F).
39. Define:
   toupper  
   ctype.h type cast  
   escape sequence  
   null character  
   atoi  
   ANSI C-style string  
   stdio.h printf  
   scanf
40. How many bytes of memory are allocated by this declaration:  
   char prompt[] = "Enter name: ";
41. Declare and initialize a variable, letter, to store the character value ‘Q’.
42. Declare and initialize a variable, team, to store the string value “Rams”.
43. Declare a string variable, town, with a storage capacity of 40 bytes.
44. What is the data type and memory allocation, in bits and bytes, of a single component of a string data item?
45. Declare the integer variables height and weight; initialize them to 67 and 125, respectively; and write a C-style (not cout) output statement to display their values, with appropriate labels.
46. Name the four standard identifiers associated with the C and C++ 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 function (F).
47. Correct the error in this C-
language code.
    int input1;
    scanf("%d", input1);
48. Label each term below with the
letter of its appropriate definition:
    _____ char
    _____ ASCII
    _____ character string
    _____ fstream.h
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.

**Long-answer problems**

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.
2. One ASCII character produces
   nonvisual output. Write a program
   that outputs *char (7)* and tell what
   its result is.
3. Write a C++ 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
4. Write a program that will print out
   a five-character ASCII string
   generated by five integer input
   values.
5. 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.
6. Find out experimentally what the
   return values of the functions
   *isalpha* and *isdigit* are for
   character parameters ‘e’, ‘Q’,
   ‘&’, and ‘4’.
7. Debug this code:
   ofstream outfile("x.out")
   int x,y;
   cout << *x=“ << x << “ y= “ << y;
8. Show your initials by assigning
   ASCII values to character
   elements of a string variable one
   at a time and displaying the string.
Multiple-choice or T/F:

1. f. '\n' is a special expression denoting the newline character.
2. t. The statement would work, for example, if \( x \) were an integer, a character, a floating-point variable, or a string variable.
3. a. A character in single quotes has the \texttt{char} data type in C++.
4. a. A character in single quotes has the \texttt{char} data type in C++.
5. b. A numeral without a decimal point is taken to be an integer.
6. a. The backslash is a meta-character that may precede some characters to denote another character, in this case, a single backslash.
7. d. An expression in double quotes is a string in C++.
8. a. \texttt{char} is the data type name for characters.
9. c. The null character literal is in single quotes and requires a backslash for the escape sequence.
10. c. \texttt{ifstream} is defined in the standard include file \texttt{fstream.h}.
11. f. \texttt{close()} closes a file; \texttt{eof()} tests whether it is still open.
12. c. The stream function \texttt{open()} opens a file for input.
13. b. The \texttt{printf} and \texttt{scanf} functions each take a first parameter that is a string constant the specifies the format of the output.
14. t.
15. t.
16. b.
17. d.
18. f.
19. e.
20. t.
21. a.

15. \texttt{string}
16. A sequence of characters coming from or going to a device.
17. stream
18. \texttt{ifstream}
19. \texttt{fstream.h}
20. \texttt{scanf}
21. \texttt{scanf("%d",&n);}
22. \texttt{scanf("%f",&n);}
23. \texttt{scanf("%c",&c);}
24. \texttt{scanf("%s",s);}
25. \texttt{stdio.h}
26. \texttt{printf("Name: %s",name)}
27.
28.
29.
30.
31.
32.
33.
34. \texttt{char letter = 'Q';}
35. \texttt{char team[] = "Rams";}
36. \texttt{char town[40];}
37. char; one byte (8 bits)
38. \texttt{int height = 67,weight = 125; printf("height: %i\nweight: %d",height,weight);}
39.
40. \texttt{stdio.h: printf (F), scanf (F), iostream.h: cin (D), cout (D)}

Short-answer:

1. \texttt{char digit;}
2. '\n' (newline), '\t' (tab), '\0' (null)
3. \texttt{scanf("%i", &input);}
4. \texttt{char}
5. \texttt{char*} or \texttt{string}
6. The null terminator, '\0'
7. \texttt{strcpy(shift,"day");}
8. \texttt{string.h}
9. char x = '\0';
10. '\n'
11. ASCII
12. \texttt{strcat, strcpy, strlen, strcmp; string.h}
13. concatenation
14. \texttt{string}
Study questions for topic 5a (Branch statements)

**Multiple-choice or T/F:**

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 C++, we write; (a) !; (b) &&; (c) ||; (d) ==; (e) "oar"
3. (T-F) The switch statement is a way to avoid multiple nested if's.
4. Which is not a kind of statement? (a) assignment; (b) declaration; (c) input; (d) branch; (e) case label
5. The break statement is used with almost all (a) if statements; (b) loops; (c) switch statements; (d) functions; (e) none of these
6. ! represents (a) disjunction; (b) conjunction; (c) logical negation; (d) abstraction; (e) encapsulation
7. Evaluate $a > b \&\& b == c$, where $a = 2, b = 1, and c = 3$? (a) $a; (b) 0; (c) 1; (d) 2; (e) none of these
8. Evaluate $a \neq b \&\& a < c$, where $a = 2, b = 1, and c = 3$? (a) $a; (b) 0; (c) 1; (d) 2; (e) none of these
9. The case keyword is used in which statement? (a) while; (b) function call; (c) assignment; (d) if; (e) switch
10. && is (a) a logical operator; (b) a relational operator; (c) an arithmetic operator; (d) a Boolean expression; (e) none of these
11. (T-F) $3 > 2 || 1 == 0$
12. (T-F) $4 > 1 || 2 != 3$
13. What C++ token always follows the if keyword? (a) statement; (b) expression; (c) left parenthesis; (d) identifier; (e) then
14. The relational operator that tests for equality is (a) =; (b) >; (c) = =; (d) !==; (e) <>

15. The relational operator that tests for inequality is (a) =; (b) >; (c) = =; (d) !==; (e) <>
16. (T-F) The keyword else may begin a statement in C++
17. The if statement implements which control structure? (a) sequence; (b) branch; (c) loop; (d) no control structure; (e) function call
18. What is the value of $a == b || b > c$, where $a = 1, b = 2, and c = 3$? (a) $a; (b) 0; (c) 1; (d) 2; (e) none of these
19. || is (a) a logical operator; (b) a relational operator; (c) an arithmetic operator; (d) a Boolean expression; (e) none of these
20. != is (a) a logical operator; (b) a relational operator; (c) an arithmetic operator; (d) a Boolean expression; (e) none of these
21. (T-F) A variable is accessible from any statement that appears after the variable’s declaration in the same block.
22. (T-F) A block is enclosed by braces.
23. (T-F) A block defines a scope of access
24. (T-F) A block is enclosed in parentheses.
25. (T-F) = is a relational operator.
26. (T-F) > is a logical operator
27. It is normally reasonable to compare items of type _______ for equality. (a) C-style string; (b) int; (c) float; (d) void; (e) none of these
28. Which is not a relational operator? (a) >; (b) <; (c) >=; (d) ==; (e) =
29. Which is not a logical operator? (a) !; (b) ||; (c) <; (d) &&; (e) none is a logical operator
30. A C++ type with two possible values is (a) int; (b) double; (c) bool; (d) char; (e) string
31. (T-F) The switch keyword implements the loop control structure.
32. (T-F) The switch statement is used for multiway branches.
33. (T-F) An alternative within a switch statement may have multiple case labels.

**Short-answer:**

1. Name three C/C++ statements that implement the decision control structure.
2. What does || mean in C and C++?
3. What statement in C or C++ takes one of exactly two different courses of action depending on the result of a test of a value?
4. What keyword is used to prevent falling through from one case label to the next in a switch statement?
5. Write an expression in C/C++ that tests whether the following is true: either height is greater than 72 or age is not less than 30
6. 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 C/C++ statement would be most recommended?
7. Write a statement that shows an error message if the value of numeric variable quantity is negative.
8. Write an expression in C/C++ whose value is true if height is in the range of 60 to 72, otherwise false.
9. What is the output of the poorly indented code below? ________

```cpp
int x = 1, y = 3;
if (x < 2)
  if (y > 4)
    cout << "A";
  else
    cout << "B";
  cout << "C";
```
10. What keyword is a case label for the condition where no other case label is matched?
11. What are the C++ logical operators discussed in this topic?
12. List the C++ relational operators.
13. What keyword is used once for almost every case label in a `switch` statement?

14. 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 `&&`?

15. 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.

16. Is there a set of braces in every `if` statement? Every `switch`?

17. 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 \( B \).)

18. 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’.

19. Name the syntax error in this statement:

   `if (X > Y) cout << X else cout << Y;`

20. What does the following program output? Indent the nested `if` statements properly.

   ```cpp
   // notgreat.cpp
   #include <iostream.h>
   void main()
   {
     int p = 1, q = 0;
     if (p) if (!q)
       cout << "Y";
     else cout << "N";
     else cout << "N";
   }
   ```

21. 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 C++ token always immediately follows the `if` keyword?

22. Distinguish “=’” from “==”.

23. Is “a > b > c” a syntax error? What rule does it follow or violate?

24. What C++ statement corresponds to the flowchart below?

25. What category of C++ statements would correspond to the flowchart below if it were filled in?

26. When would you not use `break` in a `switch` statement?

27. Why would a single set of test data be insufficient to thoroughly test a program containing an `if` statement?

28. Does \( 2 < 3 \& \& 4 > 1 \) mean the same as \(!(!2 < 3) || !(4 > 1)\)? Explain.

29. 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)`

30. Debug this error-ridden code:

   ```cpp
   cout << "Do you drink soda pop?"; char input;
   cin >> input;
   switch (input)
   {
     'y': cout << 'Recycle your bottle\n';
     'n': cout << 'Enjoy your juice\n';
     default: cout << 'Y' or 'N' please\n;
   }
   ```

31. What is wrong with this statement?

   ```cpp
   if (ltr == 'a' || 'b' || 'c')
     cout << "Letter is early in the alphabet\n";
   ```

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

   ```cpp
   int n = 3;
   if (n = 2) cout << '2';
   else cout << "not 2";
   ```

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

   ```cpp
   char name[40];
   cout << "Name? ";
   cin >> name;
   if (name == ")
     cout << "Is blank\n";
   ```

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

   ```cpp
   if (x == 2)
     y = 1;
   cout << "same";
   else cout << "different";
   ```

35. What do indents mean in C++ syntax?

36. 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`
37. List and identify (a) the C/C++ relational operators; (b) the logical operators.

38. Write one C or C++ 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.

39. Write a statement that tests whether a variable, quantity, is over 10, and if so reduces the value of another variable, prior, by 5% and displays the message “discounted”

40. Evaluate these C/C++ expressions:
   \[ 1 + 1 == 2 && 4 != 5 \]
   \[ 3 < 2 || 9 > 5 \]

**Longer answer problems:**

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 function 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:

<table>
<thead>
<tr>
<th>Letter</th>
<th>Dial digit</th>
</tr>
</thead>
<tbody>
<tr>
<td>'A'..'C'</td>
<td>2</td>
</tr>
<tr>
<td>'D'..'F'</td>
<td>3</td>
</tr>
<tr>
<td>'G'..'I'</td>
<td>4</td>
</tr>
<tr>
<td>'J'..'L'</td>
<td>5</td>
</tr>
<tr>
<td>'M'..'O'</td>
<td>6</td>
</tr>
<tr>
<td>'P'..'R'..'S'</td>
<td>7</td>
</tr>
<tr>
<td>'T'..'V'</td>
<td>8</td>
</tr>
<tr>
<td>'W'..'Y'</td>
<td>9</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Height</th>
<th>Weight</th>
<th>Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt; 72 in.</td>
<td>&gt; 190 lb.</td>
<td>Tall and heavy</td>
</tr>
<tr>
<td>&gt; 72 in.</td>
<td>≤ 190 lb.</td>
<td>Tall and light</td>
</tr>
<tr>
<td>≤ 72 in.</td>
<td>&gt; 170 lb.</td>
<td>Short, heavy</td>
</tr>
<tr>
<td>≤ 72 in.</td>
<td>≤ 170 lb.</td>
<td>Short, light</td>
</tr>
</tbody>
</table>

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.

```
#include <iostream.h>

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

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:

<table>
<thead>
<tr>
<th>Letter</th>
<th>Dial</th>
</tr>
</thead>
<tbody>
<tr>
<td>A B C</td>
<td>2</td>
</tr>
<tr>
<td>D E F</td>
<td>3</td>
</tr>
<tr>
<td>G H I</td>
<td>4</td>
</tr>
<tr>
<td>J K L</td>
<td>5</td>
</tr>
<tr>
<td>M N O</td>
<td>6</td>
</tr>
<tr>
<td>P R S</td>
<td>7</td>
</tr>
<tr>
<td>T U V</td>
<td>8</td>
</tr>
<tr>
<td>W X Y</td>
<td>9</td>
</tr>
</tbody>
</table>

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. Simplify this code:

```
char input;
    cin >> input;
    switch(input)
    {
        case '0':
            cout << "digit"; break;
        case '1':
            cout << "digit"; break;
        case '2':
            cout << "digit"; break;
        case '3':
            cout << "digit"; break;
        case '4':
            cout << "digit"; break;
        case '5':
            cout << "digit"; break;
        case '6':
            cout << "digit"; break;
        case '7':
            cout << "digit"; break;
        case '8':
            cout << "digit"; break;
        case '9':
            cout << "digit"; break;
    }
```
Multiple-choice or T/F:
1. d. The other relational operators are <, ==, <=, >, !=.
2. c. The OR logical operator takes two logical operands.
3. t. A cascade of if's is replaced in switch by a list of case labels and associated statements.
4. e. A case label is part of a switch (branch) statement.
5. 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.
6. c. The C++ not operator is "!", which turns all zero values to 1 and all nonzero values to zero.
7. b. The expression is false because b(1) is not equal to c(3). False evaluates to 0.
8. c.
9. e. A switch statement should contain case labels, followed by statements that will execute when a match to the switch selector is found.
10. a. The && operator is the logical operator of conjunction, AND.
11. t. 3 is greater than 2, and the OR operator returns true if either operand is true.
12. t.
13. c. Statements and expressions are not tokens; the left parenthesis is mandatory.
14. c. The two equal signs are a single relational operator. The answer is not (a) because the = operator is the assignment operator.
15. d.
16. f. The keyword else is part of the if statement and should be preceded by a semicolon.
17. b. The if statement is a selection statement.
18. b. The test fails on both terms.
19. a. || is the logical operator of disjunction; OR
20. b. != is the relational operator for inequality.

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)
   cout << "Invalid value\n";
8. height >= 60 && height <= 72
9. BC
10. default
11. !, ||, & &
12. <, >, <=, >=, !=, ==
13. break
14. 0 (false); 1 (true); 0 (false)
15. 0 (false); 1 (true); 0 (false)
16. 0 (false); 1 (true); 0 (false)
17. switch (input)
   {
   case 'a':
      cout << "Hello";
      break;
   case 'b':
      cout << "One moment";
      break;
   case 'c':
      cout << "Bye";
      break;
   }
18. 0 (false); 1 (true); 0 (false)
Study questions for topic 5b (Loop statements)

Multiple-choice or T/F:

1. The most appropriate statement for reading data from a file is (a) if; (b) while; (c) do; (d) switch; (e) assignment
2. The while statement is (a) counter driven; (b) bottom tested; (c) recursive; (d) top tested; (e) to be avoided
3. 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
4. ++ 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
5. 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
6. (T-F) In a bottom-tested loop the body is always executed at least once.
7. What appears after the do keyword is (a) a Boolean expression; (b) always a compound statement; (c) a statement; (d) a parenthesis
8. The for statement implements (a) a counted loop; (b) a top-tested loop; (c) a multi-way branch; (d) a recursive function
9. 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
10. (T-F) It is possible to use any of the three C++ loop statements to solve a given problem that involves repetition.

Short-answer:

1. Write a C or C++ statement that displays the numbers from 100 to 200.
2. Name three C++ loop statements.
3. How many asterisks would the code below display?
   ```c
   for(int i=0; i < 40; ++i)
      cout << "*";
   ```
4. What is the output of this code?
   ```c
   for(int i=10; i < 16; ++i)
      if(i % 5 != 0)
         printf("%s", i);
   ```
5. How many asterisks are output by this code?
   ```c
   for(int i=0; i < n; ++i)
      cout << "*";
   ```
6. Which C/C++ 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 C/C++ 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 C++ loop statements, using keywords and brief (two-word) descriptions.
11. How many statements comprise the loop body in each C++ loop statement?
12. What is the C++ code in parentheses after while for?
13. What does the C++ code after the second semicolon in parentheses in the for statement do?
14. Which C++ 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:
   ```c
   for (int i=0; i < m; i++)
      cout << setw(i * i) << '*';
   ```
16. 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?
17. 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.
18. How many iterations do nested 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?
19. 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?
20. What predefined C++ function would you use to simulate a process that produces a sequence of unpredictable numbers, such as a list of winning lottery numbers?
21. 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.
   ```c
   int a = 1;
   while (a < 10)
   {
      cout << a << endl;
      a = a + 1;
   }
   ```
22. What is the output of the following program code?
```c
int x = 10;
while (x > 0)
  x = x - 3;
cout << x << endl;
```
23. 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.
```c
int n;
cin << n;
if (n > 0)
{
  int sum = 0;
  int i = n - 1;
  while (i > 0)
  {
    sum = sum + i;
    cout << "The sum up to " 
    << n << " is " << sum 
    << endl;
  }
}
```
24. What is the risk in using a `do...while` loop to read data items from a text file?
25. What is wrong with the following `main` function, intended to compute $2^{10}$?
```c
void main()
{
  int power;
  while (power < 1000)
  {
    cout << power << endl;
    power = power * 2;
  }
}
```
26. (Challenge Exercise) Construct a flowchart of the logic of the `for` loop.
27. Debug this code, which compiles to an error message, “Multiple declarations of ‘m’”:
```c
for (int m = 0, m < 10, ++m)
  cout << m;
```
28. Debug this code:
```c
int num = 5, i;
for (i=0, i < num, ++num)
  cout << i;
```
29. How many stars does the code below output? How many if the braces are removed?
```c
for (int i=0, i < 5, ++i)
{
  cout << '*';
  for (int j=0; j < 10; ++j)
    cout << '*';
}
```
30. 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
   - ifstream::eof
   - ifstream::get
   - infinite loop
   - iteration
   - nested loop
   - ofstream::put
   - return
   - top tested
   - while

   a) repetition
   b) loop that iterates a predeterminated number of times
   c) top-tested C++ loop statement
   d) loop without working exit condition
   e) single-character file input routine
   f) single-character file output routine
   g) state of affairs when all available characters have been read
   h) function used to tell when to exit file-reading loop
   i) system value used to keep track of where next access should occur
   j) loop whose exit condition is before body
   k) loop whose body is before exit condition
   l) bottom-tested C++ loop statement
   m) data item used to count a loop’s iterations or otherwise set exit condition
   n) unstructured statement enabling exit from loop but not function
   o) unstructured statement enabling exit from function
   p) loop within a loop
   q) C++ keyword for counted loop
31. Name the three C++ loop statements, using keywords and brief (two-word) descriptions.
32. What does the C++ 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 function, 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?
```c
int a = 1;
while (a < 10);
  a = a + 1;
cout << a << endl;
```
35. Name the three C++ loop statements, using keywords and brief (two-word) descriptions.
36. What does the C++ code after the second semicolon in parentheses in the `for` statement usually do?
1. 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 function, etc

**Longer answer problems:**

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 C or C++ program:

```
Prompt for price, depreciation rate, expected life
value < price years remaining < expected life F
T

value < price

value - price

value - price

value - price

value - price

value - price

T

Display value
```
14. Implement this flowchart as a C or C++ program:

```
Prompt for price, depreciation rate, expected life

value < price years remaining < expected life

value < price

T

Display value
```
15. Write a program that displays a logarithm table, showing the natural logarithms of integer values from 1 to 100.
16. 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.
17. Input pairs of real numbers (e.g., \( r_1 \) and \( r_2 \)). Display the absolute value of the difference, \( |r_1 - r_2| \), for each input pair. Exit from the loop if \( r_1 < 0 \).

**Sample output:**
```
r_1 r_2 |r_1 - r_2|
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 \( (a+b) \neq 0 \) 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
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 for topic 5b

Multiple-choice or T/F:

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.

Short-answer:

1. for (int i = 100; i <= 200; ++i) cout << i << " ";
2. while, do...while, for
3. 400 asterisks
4. ****
5. 50 asterisks
6. do...while
7. A for loop has in parentheses an initialization, an exit/continuation test, and an update.
8. The loop must have a compound statement as its body.
9. The loop will be infinite.
Study questions for topic 6 (Subprograms)

Multiple-choice or T/F:
1. (T-F) A function body must contain the name of the function.
2. (T-F) The void data type is used when it is not known what the user will input.
3. (T-F) A function declaration specifies what the function does in detail.
4. (T-F) Procedural abstraction consists of breaking down a problem or program into simpler parts.
5. How many definitions must a function have in a program? (a) 0; (b) exactly 1; (c) 2; (d) at least one
6. 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 function definitions
7. The top-down approach 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 functions; (d) originated with object-oriented programming
8. (T-F) According to the text, a single function should perform a wide variety of tasks.
9. A function prototype (a) introduces the function name to the program; (b) is a declaration; (c) must occur before the function is used; (d) all of the above; (e) none of the above
10. Reasons to define functions 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
11. We are urged to (a) avoid local variables; (b) avoid global variables; (c) maximize side effects; (d) avoid global constants; (e) none of these
12. Local variables are located (a) within the function's machine code; (b) in the same area as global variables; (c) on the stack; (d) in the microprocessor
13. A function definition (a) is a function declaration; (b) has a header and a body; (c) focuses on minimizing the number of functions; (d) originated with object-oriented programming
14. A value may be passed out of a function to the calling statement with a (a) value parameter; (b) reference parameter; (c) variable parameter; (d) goto statement
15. (T-F) All parameters and variables declared by name are passed using the stack.
16. (T-F) A formal parameter appears in parentheses in a function call.
17. With a value parameter, what is passed to the function? (a) the address; (b) the value of an expression; (c) the full text of the expression; (d) nothing; (e) a request for information
18. A function 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
19. Which would you use to cause a function to prompt for two quantities and pass them back to the calling statement? (a) a loop; (b) value parameters; (c) reference parameters; (d) return values; (e) a class
20. In a function call we might find (a) a function definition; (b) a function header; (c) a formal parameter; (d) an actual parameter; (e) none of these
21. Which operator is used to declare a parameter by reference? (a) none; (b) ref; (c) &; (d) *; (e) !
22. According to the text, the best way to share temporary data items between functions is (a) class members; (b) global variables; (c) local variables; (d) parameters; (e) all of the above
23. Parameter values are stored (a) with the calling function's machine code; (b) with the called function's machine code; (c) in the calling function's activation record on the stack; (d) in the called function's activation record on the stack; (e) in a program's variable memory
24. The stack is like a (a) bulletin board; (b) rope; (c) bookshelf; (d) chain; (e) none of these
25. (T-F) A given identifier may be used in only one function declaration.
26. (T-F) The values of all parameter expressions are copied from the calling function into the called function.
27. A function that returns a value (a) must do so with a parameter; (b) should specify the data type of that value in the function header. (c) does so with an assignment statement; (d) does so automatically
28. (T-F) A recursive function should have an if or switch statement in it.
29. (T-F) A recursive function is one that calls itself.
30. Which is not a kind of loop statement in C++? (a) repeat; (b) counter-driven; (c) top-tested; (d) bottom-tested; (e) recursive
31. 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
32. (T-F) A function call should contain a data type name.
33. (T-F) A function prototype should contain a data type name.
34. (T-F) A function definition should contain a data type name.

Short-answer:
1. What are two ways that a function can pass a data value back to the statement that called it?
2. Write the prototype for a function that accepts two float parameters and returns their integer sum.
3. Write an appropriate declaration for a function that accepts two integers, returns no value, and draws a figure. You need not spell out everything that the function does.
4. In the module hierarchy diagram below, (a) name functions that call show_results; (b) name functions 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?

5. What tools does an application programmer have in addition to the keywords and operators of the C++ language?
6. Make up names for three functions 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.
7. What is missing from this function definition?
   void fetch_price()
   cout << "Enter price: " cin >> price;
8. Give two reasons for using a modular approach to program development.
9. What is the difference between a function declaration and a function definition?
10. What must be true of calls to functions with the void type designation?
11. Name a technique used in debugging to pinpoint the source of an incorrect output value.
12. What is one way to share data among functions without using parameters or return values? What are its disadvantages?
13. Put these phrases or subphrases of the problem-solving process in chronological order, numbering the first “1”, the second “2”, etc.
14. _____ code program
15. _____ desk check
16. _____ write a design
17. _____ get problem specifications
18. _____ test program
19. _____ debug code
20. 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() {}
21. Do the C++ statements in a function always execute at least once when the program containing the function executes? Justify your answer.
22. Find the syntax or other errors in the programs below
   (a) // hello2.cpp
      #include <iostream.h>
      void main() { say_hello(); } void main() { } void say_hello() {
   (d) // hello5.cpp
      #include <iostream.h>
      void say_hello();
      void main() { say_hello(); } void main() { } void say_hello() {
   (e) // hello5.cpp
      #include <iostream.h>
      void say_hello();
      void main() {
        // says hello
        say_hello();
      }
      void say_hello() {
        // This function is empty
      }
22. Could a void function 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)
   { cout << x << endl;
   }
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 \( \text{age} - 2 \) be a reference parameter? Explain.

29. What can you conclude about two variables or parameters, used in two different functions, if the data items have the same identifier?

30. When a reference parameter of type \texttt{char} is passed, how many bytes of data are copied to the stack and communicated to the called function?

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?

```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?

```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 function 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 function declaration:
```cpp
void draw_rectangle(int width, int ht);
```

37. What is wrong with this recursive function?
```cpp
void get_amt()
{
    cout << 'Sales amt: ';
    float amt;
    cin >> amt;
    get_amt();
}
```

38. What is a difference between the body of any \texttt{void} function and the body of a non-\texttt{void} function?

39. Label each term below with the letter of its appropriate definition:
   - debugging
   - trace statement
   - function
   - function declaration
   - function definition
   - global variable
   - intermediate variable
   - local variable
   - module
   - procedural abstraction
   - recursion
   - activation record
   - actual parameter
   - base case
   - formal parameter
   - function call
   - function declaration
   - parameter passing
   - pass-by-reference
   - pass-by-value
   - recursive case
   - return value
   - stack
   - implementation code that allows a programmer to see values of variables while debugging.
   - a data item accessible to all functions in a program.
   - a data item accessible only to function that declares it.
   - a prototype, introducing identifier to a program.
   - a C++ subprogram.
   - use of a parameter that may communicate a new value back to the calling statement
   - mechanism for passing a value to calling statement using the function call as an expression
   - introduction of subprogram name to a program
   - argument as found in function header
   - data item that stores a function invocation in memory
   - nonrecursive branch in recursive function
   - language mechanism for communicating data between functions
   - use of a parameter that communicates data into called function but not out
   - invocation of subprogram
   - argument found in function call
   - attribute of a function that calls itself
   - memory structure for sharing data between functions
   - mechanism for automatically initializing member data items
   - statement in recursive function in which function calls itself
Longer answer problems:

1. Write a program that defines and calls two functions: 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 function 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 function 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 function that accepts an integer parameter, length, and displays a line of that many hyphens.

5. Write the definition of a function that accepts two integer values as arguments and that returns the smaller one.

6. Break down the following program into three functions, input, calculate, and display, each called from main. Use parameters rather than global variables to share data among functions.

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

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. Declare function prototypes, including parameter declarations.

8. Write a program that defines and calls two functions: 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 function 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

9. Write a program that declares, calls, and defines a function that accepts three integer values as parameters and returns the smallest one.

10. Use two functions, one to draw a horizontal line of eight asterisks and the other to draw two asterisks separated by six spaces on the same line.

11. Modify your solution to Exercise 1 to include a new function that calls the other two. Each time the new function is called, a box is drawn.

12. Write a function that repeatedly prompts for and inputs integer values, until the user enters a zero value, and displays the smallest one and the average.

13. Write a program containing a function, 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 function four times to put a meal together. Print the meal file.

14. Here is a structure diagram of a modularized program that adds four input values and shows the sum:

   ![Structure Diagram]

   Implement this design, using a function for each of the three indicated modules. Note that the function 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.

15. State income tax for the State of Panic is computed according to the following formula:

   \[\text{Net income is gross income minus deductions.}\]

   Use a while loop to permit input of multiple sets of data (name, gross sales, and deductions). Use separate functions for (1) data input, (2) tax computation, and (3) a tax report written to the screen.

   Sample Tax Report:

<table>
<thead>
<tr>
<th>Name</th>
<th>Gross Sales</th>
<th>Deductions</th>
<th>Net</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joe Smith</td>
<td>23000.00</td>
<td>7000.00</td>
<td>16000.00</td>
</tr>
<tr>
<td>Linda Tucker</td>
<td>25000.00</td>
<td>670.00</td>
<td>24330.00</td>
</tr>
<tr>
<td>Harry Carey</td>
<td>17000.00</td>
<td>9000.00</td>
<td>8000.00</td>
</tr>
</tbody>
</table>

16. Use separate functions 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 \(\pi\) 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 \(\pi r^2\).
18. 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 function 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:

<table>
<thead>
<tr>
<th>ID</th>
<th>OLD SALARY</th>
<th>PAY RAISE</th>
<th>NEW SALARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>25000.00</td>
<td>1375.00</td>
<td>26375.00</td>
</tr>
<tr>
<td>2</td>
<td>28058.00</td>
<td>1543.19</td>
<td>29601.19</td>
</tr>
<tr>
<td>3</td>
<td>32000.00</td>
<td>1760.00</td>
<td>33760.00</td>
</tr>
<tr>
<td>4</td>
<td>26500.00</td>
<td>145.50</td>
<td>27955.50</td>
</tr>
<tr>
<td>5</td>
<td>31000.00</td>
<td>1705.00</td>
<td>32705.00</td>
</tr>
</tbody>
</table>

TOTALS

<table>
<thead>
<tr>
<th>ID</th>
<th>OLD SALARY</th>
<th>PAY RAISE</th>
<th>NEW SALARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>31000.00</td>
<td>1705.00</td>
<td>32705.00</td>
</tr>
</tbody>
</table>

20. Use a function to compute the sum of the first n terms of the geometric series:

\[ a + ar + ar^2 + ar^3 + \ldots + ar(n-1) \]

where \( a \) (real), \( r \) (real), and \( n \) (integer) are input from the keyboard. The function should accept three parameters. For example, if \( a = 2 \), \( r = 3 \), and \( n = 4 \), then the function should return \( 2 + 2*3 + 2*3*3 + 2*3*3*3) = 80.0. \)

21. 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.

22. 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 function 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

23. Input an unsorted list of integers. Let the user enter a sentinel value of 999 to terminate input. Use a single function 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.

24. (Challenge) Input two positive integers. Use a function 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

25. 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.

26. 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 \( (x_1, x_2, x_3, \ldots, x_n) \) having a mean \( \mu \) is:

\[ \sqrt{(x_1-\mu)^2 + (x_2-\mu)^2 + \ldots + (x_n-\mu)^2} \]
27. Write a program that will compute the mean and standard deviation of four input values. Use separate functions 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

28. 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.

29. 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 the loop if the input value is less than or equal to 1.

30. 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 function that accepts the four specifications as parameters.

31. Write a function that tells whether its integer parameter is odd.

32. (Challenge) What common function presented in calculus courses is computed by this C++ function?
```cpp
int f(float n)
{
    if (n < 2)
        return 0;
    else
        if (n == 2)
            return 1;
        else
            return f(n / 2) + 1;
}
```

33. Write a program containing a Boolean function 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 value.

34. Use a float function 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.

35. Write a program that accepts a person's full name and prints the initials. Include a function that returns the first letter after a space in a string, starting at a given character position.

36. 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 function for each of the first four menu options.

37. Write a function `pos(substring,string)` that returns the position of the first character of `substring` in `string`, or returns −1 if `string` does not contain `substring`.

38. Write a recursive function that returns the product of two positive-integer arguments. The function may not use the '*' operator or any loop statement.

39. Write a recursive function 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

40. Write two Boolean functions 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.

41. Write a function that accepts two strings and an integer. The integer should represent a position in the first string. The function should insert the second string into the first at the position indicated. For example, if string variable `s1` has the value "resion", a function call `insert(s1,"cur",2)` should result in `s1` having a value "recursion".

42. Write a function 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 function should return 2.

43. Write and demonstrate three functions 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 functions in one program. What is the C++ feature that lets you use `prompt_for` as the name of all three functions in the same program?
44. Write a function to return the integer value of a string parameter. For example, if the string parameter is “43”, the function should return the integer 43.
Answers to study questions for topic 6

Multiple-choice or T/F:
1. f. The function name is in the header.
2. f. The user's input is rarely known by the programmer; the void type is used for functions that return no value.
3. f. The function declaration introduces it into a source file; it is the definition that spells out what statements are in it.
4. t. Defining functions in order to modularize a program is called procedural abstraction.
5. b. A function must have one definition and may have many declarations.
6. d. Writing function definitions is a way to break down a program or problem into smaller, easily understood, independent units.
7. a. The solution to each subproblem becomes a module or function.
8. f. A goal of modularity is cohesion within each function: the purpose of the function should be singular and focused.
9. d. A prototype, or declaration, is the same as the function header. Functions are often declared at the top of a program and defined after main.
10. c. Breaking down a program into components and grouping related, repeated sequences of statements are the main goals of writing functions.
11. b. Local variables are preferred to global ones because they guarantee against side effects.
12. c. A local variable is allocated when a function begins to execute and disappears when the function terminates.
13. b. The function header has the same form as a function declaration, without the semicolon; the body is a compound statement.
14. b. A reference parameter is specified with "&" after the type name.
15. t. The stack allows a function to share data and to allocate space that is freed up when the function terminates.
16. f. The formal parameter appears in the function definition; the function 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 function header.
19. c. Reference parameters will work; value parameters cannot pass data back, and return values only pass back a single value.
20. d. Actual parameters occur in function calls, formal parameters in function definitions and declarations.
21. c. The ampersand is used to declare a reference parameter, which passes the address of a variable.
22. d. Members of a class are useful, but should represent relatively long-lasting values. Parameters are better.
23. d. Parameters declared in a function's header disappear when that function call's activation record disappears from the stack.
24. a. The stack, like a bulletin board, allows functions to share information as coworkers do.
25. f. Function names may be overloaded. The same name may be used for several functions or constructors with parameter lists of different lengths and parameters of different data types.
26. f. Reference and array parameters have only their addresses copied.
27. b. A data type is associated with every function.
28. t. The branch should allow the function to process the base case, allowing the function to terminate, or the recursive case, causing the function to call itself.
29. t. A recursive function calls itself and must also have a way to terminate without calling itself.
30. a. C++ has no repeat statement.
31. a. A function that invokes itself is recursive and the result is a loop.
32. f. A function call may be an expression or statement and needs no type name.
33. t. A function prototype must begin with a type name.
34. t. A function definition’s header must begin with the function’s return type.

Short-answer:
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
Multiple-choice or T/F:

1. A data item that 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.

2. (T-F) A structure, unlike an integer, may be compound.

3. (T-F) A member item in a structure type or class is accessible to only one function.

4. Creating new data types is (a) impossible; (b) data abstraction; (c) procedural abstraction; (d) to be discouraged;

5. (T-F) A class is an instance of an object.

6. (T-F) Objects are widely considered a factor in making it harder to write longer programs.

7. (T-F) Functions that are members of a class share access to all data members of an object of that class.

8. (T-F) The only ways for different functions to share data in RAM is by use of global variables or parameter passing.

9. (T-F) A class is a data type as opposed to a data item.

10. In a C++ program, the identifier `part.price` could represent (a) a class; (b) an object type; (c) a function header; (d) a member of an object; (e) none of these.

11. (T-F) A function may be a member of a class.

12. An instance of this class occupies approximately how many bytes of RAM: `class persons
(char name[30]; int age; );` (a) 0; (b) 2; (c) 30; (d) 34; (e) none of these.

13. :: is (a) the scope resolution operator; (b) the delimiter used to access any member of an object; (c) used in all function declarations; (d) a logical operator; (e) not part of C++.

14. (T-F) A structure may be passed as a parameter.

15. 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.

16. (T-F) An identifier declared in braces after `enum` is a sub-component inside an enumerated-type data item.

17. (T-F) With `enum` we supply a full list of possible values of a data type.

18. An enumerated-type value is (a) a compound item; (b) true or false; (c) an `int`; (d) a `float`; (e) a `char`.

19. A compound data type may be created with (a) `struct`; (b) `enum`; (c) `int`; (d) `for`.

20. (T-F) A structure value may be passed as a parameter or returned by a function.

21. (T-F) A structure may be assigned a value only by assigning values to its members one at a time.

22. A structure within a structure is (a) a syntax error; (b) a logic error; (c) a nested structure; (d) a recursive structure.

23. It is often useful to (a) create a structure type especially for a certain function; (b) create a function especially for a certain structure type; (c) declare a structure item without a type name; (d) declare global functions to manipulate global structure variables.

24. (T-F) When a class is declared using the `class` keyword, member items are public, unless specified otherwise.

25. What keyword is used to prevent a member item from being accessed from outside a class's functions? (a) `restricted`; (b) `local`; (c) `private`; (d) `public`; (e) `protected`.

26. (T-F) A constructor should be called only when the programmer is ready to initialize members of an object with new values.

27. An object may have its members initialized in the statement that declares it, using (a) the initialization operator once for each member; (b) a constructor; (c) an access function; (d) `cin`; (e) `cout`.

28. (T-F) A program calls a constructor only if the programmer uses its name in a statement.

29. (T-F) An instance of a class may have its members automatically initialized at the time it is declared.

Short-answer:

1. If `items` is a structure 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. Name a keyword that may be used to declare a data type with which to declare objects.

3. Write a statement that assigns the value 82 to the `season` member of an object, `part`, which is an instance of class, `parts`.

4. Write an enumerated type declaration denoting the four seasons and a declaration for a variable, `season`, of that type.

5. What C++ techniques presented thus far may be used to share data among different functions? Which do the textbook authors prefer?

6. Distinguish between a structure variable and a structure type.

7. Where is a member identifier declared—in a structure variable declaration, in a structure type declaration, or in its own variable declaration?

8. If `employee.id_num` is an integer, what kind of data item is `employee`?
9. 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()
{
    cout << "Name: 
persons person;
    cin >> person.name;
    cout << "Age: 
    cin >> person.age;
    cout << person.name
    << " is 
    << person.age
    << " years old. 
";
}

10. How are structures and character strings alike, as opposed to the data types int, char, and float?

11. Is the following declaration legal? What if anything is wrong with it?
struct employees
{
    char name[40];
    long salary;
    employees supervisor;
};

12. What is the difference between (a) a structure and an object; (b) a member function and a member variable; (c) a class and an object?

13. How is a time capsule similar to an object that encapsulates data and functions?

14. Write a declaration of a car class, naming some of its features and allowing values to be specified for them.

15. Why is the following code invalid?
struct bits
{
    char name[80];
    bits bit;
};

16. Why is the following code invalid?
struct ID
{
    char name[80];
    int size = 10;
};

17. What is the number of bytes occupied by an instance of the following structure type?
struct employees
{
    char name[20];
    long salary;
    int age;
};

18. 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 function or in a free function?

19. Why would a programmer wish to declare a new data type?

20. In a program that declared a class employees and an instance emp of that class, would it be appropriate for function employees::display_info to refer to emp? Why or why not?

21. Declare a class circles with members x, y and radius, and write constructors that initialize a variable of type circles to: (a) a screen location of (320,240) and a radius of 100; (b) a screen location and radius specified in the declaration of the variable.

22. 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:
    char name[40];
    int hours;
};
It has:
(a) How many members? _____
(b) How many member data items? _____
(c) How many constructors?

23. Label each term below with the letter of the appropriate definition:
   ________ class
   ________ data abstraction
   ________ encapsulation
   ________ member function
   ________ member variable
   ________ object
   ________ object-based design
   ________ structure
   ________ structure 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 structure or object.
(c) A named category of data items that may be used to declare a structure.
(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.

Longer-answer:
1. Declare a class to represent a machine part, with a name, an inventory quantity, and a price. Declare and define functions 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 member functions for the class, access functions, and a constructor.

3. Declare a structure type or class for a machine part, with an identification number, an inventory quantity, and a price. Declare, define and call functions 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 functions 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 structures 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.

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

7. Declare some C++ classes and define some of their member functions to represent a registration 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 structure type or class for times of the day that could be expressed on a clock.

10. Write a structure type or class for durations of time using the units of time found in a clock.

11. Write a structure type or class for a date on a calendar.

12. Write a structure 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 functions 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 functions that implement addition, subtraction, multiplication, and division for the structure type rationals by accepting a parameter of type rationals.

15. Write a function 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 for topic 7

Multiple-choice or T/F:

1. c. Types, classes, and control structures are not data items; an integer may not have several attributes.
2. t. A structure may have one or more members.
3. f. A member data item is accessible to all member functions.
4. b. Example: structures and classes (composition).
5. f. An object is an instance of a class.
6. f. Objects are considered a factor in making it easier to write long programs.
7. t. Member functions have automatic access to all data members.
8. f. Two other ways: member functions of a class share data members of that class, and return values pass data back from a function to its calling statement.
9. t. An object is a data item that is an instance of a class. A class is like a template or category.
10. d. The period separates the object name from the member item.
11. t. A member function is declared in the class declaration.
12. c. The values of the constants named in an enum are 0, 1, 2, …
13. a. A structure is a compound item, composed of its members.
14. t. A structure is like a simple-type data item in parameter passing.
15. f. A structure variable may be initialized when declared, with a list of values between braces, or assigned the value of another structure with the assignment operator.
16. c. A structure may have another structure as a member.
17. b. A function that accepts a structure as a parameter in effect implements an operation associated with that structure type.
18. f. Members are private, unless they follow the keyword public.
19. c. A private member is visible only to member functions of its class.
20. f. A constructor is called automatically when an object instance is declared.
21. 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.
22. f. A constructor is executed automatically on declaration of an instance of its class.
23. t. A constructor is called automatically and may initialize member items.
24. t. When a structure type is declared with the struct keyword, member access is public by default.
25. t. A structure variable occupies memory, has a value, and is an instance of a structure type, which does not have a value or occupy memory.
26. f. A constant named in an enumerated type declaration is a possible value of an item, not a component.
27. a. Employee is a structure or object.
28. error: need semicolon after structure type declaration
29. b. Structures and character strings are compound, the other types named are simple.
30. Missing semicolon at end
31. (a) A structure, unlike an object, may not call member functions.

Short-answer:

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
1. (a); (b); (c); (d); (e)
2.