**Checkpoint**

1.1 Why is the computer used by so many different people, in so many different professions?

1.2 List the five major hardware components of a computer system.

1.3 Internally, the CPU consists of what two units?

1.4 Describe the steps in the fetch/decode/execute cycle.

1.5 What is a memory address? What is its purpose?

1.6 Explain why computers have both main memory and secondary storage.

1.7 What are the two general categories of software?

1.8 What fundamental set of programs control the internal operations of the computer s hardware?

1.9 What do you call a program that performs a specialized task, such as a virus scanner, a le-compression program, or a data-backup program?

1.10 Word processing programs, spreadsheet programs, e-mail programs, Web browsers, and game programs belong to what category of software?

1.11 What is an algorithm?

1.12 Why were computer programming languages invented?

1.13 What is the difference between a high-level language and a low-level language?

1.14 What does *portability* mean?

1.15 Explain the operations carried out by the preprocessor, compiler, and linker.

1.16 Explain what is stored in a source file, an object file, and an executable file.

1.17 What is an integrated development environment?

1.18 Describe the difference between a key word and a programmer-defined identifier.

1.19 Describe the difference between operators and punctuation symbols.

1.20 Describe the difference between a program line and a statement.

1.21 Why are variables called “variable”?

1.22 What happens to a variable’s current contents when a new value is stored there?

1.23 What must take place in a program before a variable is used?

1.24 What are the three primary activities of a program?

1.25 What four items should you identify when defining what a program is to do?

1.26 What does it mean to “visualize a program running”? What is the value of such an activity?

1.27 What is a hierarchy chart?

1.28 Describe the process of desk-checking.

1.29 Describe what a compiler does with a program’s source code.

1.30 What is a run-time error?

1.31 Is a syntax error (such as misspelling a key word) found by the compiler or when the program is running?

1.32 What is the purpose of testing a program with sample data or input?

1.33 Briefly describe the difference between procedural and object-oriented programming.

2.9 Refer to the data types listed in Table 2-6 for these questions.

A) If a variable needs to hold numbers in the range 32 to 6,000, what data type would be best? Short or Unsigned short

B) If a variable needs to hold numbers in the range 40,000 to +40,000, what data type would be best? Integer

C) Which of the following literals uses more memory? 20 or 20L they both use the same amount of memory.

2.10 On any computer, which data type uses more memory, an integer or an unsigned integer? They both use the same amount of memory.

2.22 How would you consolidate the following definitions into one statement?

int x = 7;

int y = 16;

int z = 28;

2.24 Is the following an example of integer division or floating-point division? What value will be stored in portion?

portion = 70 / 3;

2.25 Write statements using the const qualifier to create named constants for the following literal values:

**Literal Value Description**

2.71828 Euler s number (known in mathematics as *e*)

5.256E5 Number of minutes in a year

32.2 The gravitational acceleration constant (in feet per second2)

9.8 The gravitational acceleration constant (in meters per second2)

1609 Number of meters in a mile

3. How may the int variables months, days, and years be defined in one statement, with months initialized to 2 and years initialized to 3?

int months = 2, days, years = 3;

5. Is the following comment written using single-line or multi-line comment symbols?

/\* This program was written by M. A. Codewriter\*/

5. Multi-line comment

9. Every complete statement ends with a

A) period

B) # symbol

**C) semicolon**

D) ending brace

11. Every C++ program must have a

A) cout statement.

**B) function main.**

C) #include statement.

13. The following data

72

'A'

"Hello World"

2.8712

are all examples of

A) Variables

**B) Literals or constants**

C) Strings

D) None of the above

15. Which of the following are *not* valid assignment statements? (Circle all that apply.)

A) total = 9;

**B) 72 = amount;**

**C) profit = 129**

D) letter = 'W';

19. The negation operator is Unary

21. When do preprocessor directives execute?

**A) Before the compiler compiles your program**

B) After the compiler compiles your program

C) At the same time as the compiler compiles your program

D) None of the above

23. T ***F*** Variable names may begin with a number

25. ***T*** F A left brace in a C++ program should always be followed by a right brace later in the program.

3.1 What header le must be included in programs using cin?

3.2 TRUE or FALSE: cin requires the user to press the **[Enter]** key when finished entering data. True

3.11 Assume the following variable de nitions:

int a = 5, b = 12;

double x = 3.4, z = 9.1;

What are the values of the following expressions?

A) b / a

B) x \* a

C) static\_cast<double>(b / a)

D) static\_cast<double>(b) / a

E) b / static\_cast<double>(a)

F) static\_cast<double>(b) / static\_cast<double>(a)

G) b / static\_cast<int>(x)

H) static\_cast<int>(x) \* static\_cast<int>(z)

I) static\_cast<int>(x \* z)

3.15 Write statements using combined assignment operators to perform the following:

A) Add 6 to x.

B) Subtract 4 from amount.

C) Multiply y by 4.

D) Divide total by 27.

E) Store in x the remainder of x divided by 7.

3.18 The following program will not compile because the lines have been mixed up.

#include <iomanip>

}

cout << person << endl;

string person = "Wolfgang Smith";

int main()

cout << person << endl;

{

#include <iostream>

return 0;

cout << left;

using namespace std;

cout << setw(20);

cout << right;

When the lines are properly arranged the program should display the following:

Wolfgang Smith

Wolfgang Smith

Rearrange the lines in the correct order. Test the program by entering it on the computer, compiling it, and running it.

3.19 The following program skeleton asks for an angle in degrees and converts it to

radians. The formatting of the nal output is left to you.

#include <iostream>

#include <iomanip>

using namespace std;

int main()

{

const double PI = 3.14159;

double degrees, radians;

cout << "Enter an angle in degrees and I will convert it\n";

cout << "to radians for you: ";

cin >> degrees;

radians = degrees \* PI / 180;

// Display the value in radians left justified, in fixed

// point notation, with 4 places of precision, in a field

// 5 spaces wide, making sure the decimal point is always

// displayed.

return 0;

}