**Chapter 1**

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

Computers can do such a wide variety of things because they can be programmed. This  
means that computers are not designed to do just one job, but any job that their programs  
tell them to do.

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

* The central processing unit (CPU)  
  •Main memory  
  •Secondary storage devices  
  •Input devices  
  •Output devices

1.3Internally, the CPU consists of what two units?

The *control unit* and the *arithmetic and logic unit (ALU)*

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

*Fetch* The CPU’s control unit fetches, from main memory, the next instruction in the sequence of program instructions.  
*Decode* The instruction is encoded in the form of a number. The control unit  
decodes the instruction and generates an electronic signal.  
*Execute* The signal is routed to the appropriate component of the computer  
(such as the ALU, a disk drive, or some other device). The signal causes  
the component to perform an operation.

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

Each byte is assigned a unique number known as an *address*. The addresses are ordered  
from lowest to highest. A byte is identified by its address in much the same way a post  
office box is identified by an address.

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

Frequently used programs are stored in secondary memory and loaded into main memory as needed

1.7What are the two general categories of software?

System Software and Application Software

1.8What fundamental set of programs control the internal operations of the  
computer’s hardware? Binary, it’s encoded magnetically into the hard drive.

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

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

1.11What is an algorithm?

An algorithm is a set of well-defined  
steps for performing a task or solving a problem.

1.12Why were computer programming languages invented?

*Programming languages*, which use words instead of numbers, were invented to ease the task  
of programming. A program can be written in a programming language, such as C++, which  
is much easier to understand than machine language.

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

A low-level language is close to the level of the computer, which means it resembles  
the numeric machine language of the computer more than the natural language of humans.  
The easiest languages for people to learn are *high-level languages*.

They are called “highlevel” because they are closer to the level of human-readability than computer-readability.

1.14What does *portability* mean?

This means that a C++ program can be written on one type of  
computer and then run on many other types of systems. This usually requires the program  
to be recompiled on each type of system, but the program itself may need little or no change.

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

During the first phase of this process, a program called the*preprocessor* reads  
the source code. The preprocessor searches for special lines that begin with the# symbol.  
These lines contain commands that cause the preprocessor to modify the source code in  
some way.

During the next phase the*compiler* steps through the preprocessed source code,  
translating each source code instruction into the appropriate machine language instruction.

During the last phase of the translation process, another program  
called the*linker* combines the object file with the necessary library routines. Once the  
linker has finished with this step, an*executable file* is created.

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

The statements written by the programmer are called*source code*, and the file they are saved in is  
called the*source file*.

If the program is free of syntax errors, the compiler stores the translated machine language instructions, which are called*object code*, in an*object file*.

. The executable file contains  
machine language instructions, or*executable code*, and is ready to run on the computer

1.17What is an integrated development environment?

These environments consist of a text editor, compiler, debugger, and other utilities integrated into a package with a single set of menus.  
Preprocessing, compiling, linking, and even executing a program is done with a single click  
of a button, or by selecting a single item from a menu.

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

Key Words

Words that have a special meaning. Key words may only be used for  
their intended purpose. Key words are also known as reserved words.

Programmer-Defined Identifiers  
Words or names defined by the programmer. They are symbolic names  
that refer to variables or programming routines.

1.19Describe the difference between operators and punctuation symbols.

Operators: Operators perform operations on one or more operands. An operand is  
usually a piece of data, like a number.  
Punctuation: Punctuation characters that mark the beginning or ending of a statement,  
or separate items in a list.

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

A “line” is just that—a single line as it appears in the body of a program.

A statement is a complete instruction that causes the computer to perform some action.

1.21Why are variables called “variable”?

A variable is a named storage location in the computer’s memory for holding a piece of  
information. The information stored in variables may change while the program is running   
(hence the name “variable”).

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

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

1.24What are the three primary activities of a program?

Input, processing, and output.

Algorithm Workbench   
Draw hierarchy charts or flowcharts that depict the programs described below. (See   
Appendix D for instructions on creating flowcharts.)   
30. Available Credit   
The following steps should be followed in a program that calculates a customer’s   
available credit:   
1. Display the message “Enter the customer’s maximum credit.”   
2. Wait for the user to enter the customer’s maximum credit.   
3. Display the message “Enter the amount of credit used by the customer.”   
4. Wait for the user to enter the customer’s credit used.   
5. Subtract the used credit from the maximum credit to get the customer’s available   
credit.   
6. Display a message that shows the customer’s available credit.

Int main(){

Int maxCredit, creditUsed;  
  
cout >> “Enter the customer’s maximum credit.”;  
cin<<maxCredit;  
cout>>“Enter the amount of credit used by the customer.”;  
cin<<creditUsed;  
Int available = maxCredit \* creditUsed;  
cout>> available;

}  
  
31. Sales Tax   
Design a hierarchy chart or flowchart for a program that calculates the total of a   
retail sale. The program should ask the user for:   
– The retail price of the item being purchased   
– The sales tax rate   
Once these items have been entered, the program should calculate and display:   
– The sales tax for the purchase   
– The total of the sale   
  
  
  
32. Account Balance   
Design a hierarchy chart or flowchart for a program that calculates the current   
balance in a savings account. The program must ask the user for:   
– The starting balance   
– The total dollar amount of deposits made   
– The total dollar amount of withdrawals made   
– The monthly interest rate   
Once the program calculates the current balance, it should be displayed on the screen.   
  
Predict the Result   
Questions 33–35 are programs expressed as English statements. What would each display   
on the screen if they were actual programs?   
33. The variable x starts with the value 0.   
The variable y starts with the value 5.   
Add 1 to x.   
Add 1 to y.   
Add x and y, and store the result in y.   
Display the value in y on the screen.  
  
X = 0;  
Y = 5;  
X2 = X + 1;  
Y2 = Y + 1;  
X2 + Y2;  
  
34. The variable j starts with the value 10.   
The variable k starts with the value 2.   
The variable l starts with the value 4.   
Store the value of j times k in j.   
Store the value of k times l in l.   
Add j and l, and store the result in k.   
Display the value in k on the screen.   
  
  
  
35. The variable a starts with the value 1.   
The variable b starts with the value 10.   
The variable c starts with the value 100.   
The variable x starts with the value 0.   
Store the value of c times 3 in x.   
Add the value of b times 6 to the value already in x.   
Add the value of a times 5 to the value already in x.   
Display the value in x on the screen.   
  
  
  
36. The following pseudocode algorithm has an error. The program is supposed to ask   
the user for the length and width of a rectangular room, and then display the room’s   
area. The program must multiply the width by the length in order to determine the   
area. Find the error.  
  
Display “What is the room’s width?”.  
Input width.  
Display “What is the room’s length?”.  
Input length.  
Area =  width ×  length.  
Display area.

**Chapter 2**

**Checkpoint**

2.1The following C++ program will not compile because the lines have been mixed  
up.  
int main()  
}  
// A crazy mixed up program  
return 0;  
#include <iostream>  
cout << "In 1492 Columbus sailed the ocean blue.";  
{  
using namespace std;

When the lines are properly arranged the program should display the following  
on the screen:  
In 1492 Columbus sailed the ocean blue.

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

// A crazy mixed up program  
#include <iostream>  
using namespace std;

int main()  
**{**

cout << "In 1492 Columbus sailed the ocean blue.";

**return 0;**

**}**

2.2The following C++ program will not compile because the lines have been mixed up.  
cout << "Success\n";  
cout << " Success\n\n";  
int main()  
cout << "Success";  
}  
  
using namespace std;  
// It's a mad, mad program  
#include <iostream>  
cout << "Success\n";  
{  
return 0;

When the lines are properly arranged the program should display the following  
on the screen:  
**Program Output**  
Success  
Success Success  
Success  
Rearrange the lines in the correct order. Test the program by entering it on the  
computer, compiling it, and running it.

// It's a mad, mad program  
#include <iostream>  
using namespace std;

int main()  
{

cout << "Success\n";  
cout << "Success";  
cout << "Success\n";  
cout << " Success\n\n";  
return 0;}

2.3Study the following program and show what it will print on the screen.  
// The Works of Wolfgang  
#include <iostream>  
using namespace std;  
int main()  
{  
cout << "The works of Wolfgang\ninclude the following";  
cout << "\nThe Turkish March" << endl;  
cout << "and Symphony No. 40 ";  
cout << "in G minor." << endl;  
return 0;  
}

The works of Wolfgang

include the following

The Turkish March

and Symphony No.40

in G minor.

2.4On paper, write a program that will display your name on the first line, your street  
address on the second line, your city, state, and ZIP code on the third line, and  
your telephone number on the fourth line. Place a comment with today’s date at  
the top of the program. Test your program by entering, compiling, and running it.

2.5Examine the following program.  
// This program uses variables and literals.  
#include <iostream>  
using namespace std;  
int main()  
{  
int little;  
int big;  
little = 2;  
big = 2000;  
cout << "The little number is " << little << endl;  
cout << "The big number is " << big << endl;  
return 0;  
}  
List all the variables and literals that appear in the program.

Variable : little, big.

Literal : 2, 2000, "The little number is ", "The big number is ".

2.6 What will the following program display on the screen?  
#include <iostream>  
using namespace std;  
int main()  
{  
int number;  
number = 712;  
cout << "The value is " << "number" << endl;  
return 0;  
}

The value is 712

2.7 Which of the following are illegal variable names, and why?  
X

99bottles = no, because there’s number at the beginning  
july97  
theSalesFigureForFiscalYear98  
r&d = no, because other than caps, lowercase, and underscore, other symbol is not allowed.  
grade\_report

2.8 Is the variable name Sales the same as sales ? Why or why not?

It’s not the same Uppercase and lowercase characters are distinct.

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? unsigned short int

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

C)Which of the following literals uses more memory? 20 or20L

20L

2.10 On any computer, which data type uses more memory, an integer or an unsigned  
integer? They have same size.

number 65 is the code for A, 66 is the code for B, and so on

2.11What are the ASCII codes for the following characters? (Refer to Appendix B )  
C = 67  
F = 70  
W =

2.12Which of the following is a character literal?  
'B' = true  
"B" = false

2.13Assuming the char data type uses 1 byte of memory, how many bytes do the  
following literals use?  
'Q' = 1 byte  
"Q" = 2 byte  
"Sales" = 6 byte  
'\n' = 2 byte

2.14Write a program that has the following character variables: first, middle ,  
and last. Store your initials in these variables and then display them on the  
screen.

#include <iostream>

using namespace std;

int main()

{

char first, middle, last;

cin>>first

>>middle

>>last;

cout<<first<<”.”<<middle<<”.”<<last;

return 0;

}

2.15What is wrong with the following program statement?  
char letter = "Z";

It shouldn’t be “ ”, but ‘ ’.

2.16What header file must you include in order to use string objects?

#include <string>

2.17Write a program that stores your name, address, and phone number in three  
separate string objects. Display the contents of the string objects on the  
screen.

#include <iostream>

#include <string>

using namespace std;

int main()

{

string name, address, phone;

cin>>name

>>address

>>phone;

cout<<name<<endl<<address<<endl<<phone;

return 0;

}

2.18Yes or No: Is there an unsigned floating point data type? If so, what is it?

No.  
2.19How would the following number in scientific notation be represented in E notation?  
6.31 1017  
6.31E17

2.20Write a program that defines an integer variable namedage and a float  
variable namedweight. Store your age and weight, as literals, in the variables.  
The program should display these values on the screen in a manner similar to  
the following:  
**Program Output**  
My age is 26 and my weight is 180 pounds.  
(Feel free to lie to the computer about your age and your weight—  
it’ll never know!)

int namedage;

float namedweight;

cin>>namedage

>>namedweight;

Cout<<”My age is “<<namedage<<” and my weight is“<<namedweight<<” pounds.\n  
(Feel free to lie to the computer about your age and your weight—\n  
it’ll never know!)”;

return 0;

2.21Is the following assignment statement valid or invalid? If it is invalid, why?  
72 = amount;

It should be

amount = 72;

2.22How would you consolidate the following definitions into one statement?  
int x = 7, y = 16, z = 28;

2.23What is wrong with the following program? How would you correct it?  
#include <iostream>  
using namespace std;  
int main()  
{  
double number;

number = 62.7;  
//double number;  
cout << number << endl;  
return 0;  
}

2.24Is the following an example of integer division or floating-point division? What  
value will be stored in portion ?  
portion = 70 / 3;

Floating-point division

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

**Literal ValueDescription**  
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

const double E = 2.71828;

const double MIN\_IN\_YEAR = 5.256E5 ;

const double GRAVITY\_FEET\_PER\_SECOND2 = 32.2;

const double GRAVITY\_METER\_PER\_SECOND2 = 9.8;

const int METER\_IN\_MILE = 1609;

**Review Questions and Exercises**  
**Short Answer**

1.How many operands does each of the following types of operators require?  
\_single operand\_Unary  
\_two operand\_\_\_Binary  
\_three operand\_\_Ternary

2.How may the double variables temp, weight, and age be defined in one statement?

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

Two.  
4.Write assignment statements that perform the following operations with the variables  
a, b, and c .

A) Adds 2 to a and stores the result in b .

Int a = 2;

int b = a;  
B)Multiplies b times 4 and stores the result in a .

int a = b \* 4;

C)Divides a by 3.14 and stores the result in b .

D)Subtracts 8 from b and stores the result in a .

Int a = b / 8;  
E)Stores the value 27 in a.

int a = 27;

F)Stores the character ‘K’ in c .

Char c = ‘K’;

G)Stores the ASCII code for ‘B’ in c .

5.Is the following comment written using single-line or multi-line comment symbols?  
/\* This program was written by M. A. Codewriter\*/

Single-line comment

6.Is the following comment written using single-line or multi-line comment symbols?  
// This program was written by M. A. Codewriter

Multi-line comment

7.Modify the following program so it prints two blank lines between each line of text.  
#include <iostream>  
using namespace std;  
int main()  
{  
cout << "Two mandolins like creatures in the\n\n";  
cout << "dark\n\n";  
cout << "Creating the agony of ecstasy.\n\n";  
cout << " - George Barker\n\n";  
return 0;  
}

8.What will the following programs print on the screen?  
A)#include <iostream>  
using namespace std;  
int main()  
{  
int freeze = 32, boil = 212;  
freeze = 0;  
boil = 100;  
cout << freeze << endl << boil << endl;  
return 0;  
}

0

100

B)#include <iostream>  
using namespace std;  
int main()  
{  
int x = 0, y = 2;  
x = y \* 4;  
cout << x << endl << y << endl;  
return 0;  
}

8

2

C)#include <iostream>  
using namespace std;  
int main()  
{  
cout << "I am the incredible";  
cout << "computing\nmachine";  
cout << "\nand I will\namaze\n";  
cout << "you.";  
return 0;  
}

I am the incredible computing

machine

and I will

amaze

you

D)#include <iostream>  
using namespace std;  
int main()  
{  
cout << "Be careful\n";  
cout << "This might/n be a trick ";  
cout << "question\n";  
return 0;  
}

Be careful

This might

be a trick question

E)#include <iostream>  
using namespace std;  
int main()  
{  
int a, x = 23;  
a = x % 2;  
cout << x << endl << a << endl;  
return 0;  
}

23

1

**Multiple Choice**  
9.Every complete statement ends with a  
C)semicolon  
  
10.Which of the following statements is correct?  
C)#include <iostream>  
  
11.Every C++ program must have a  
B)functionmain  
  
12.Preprocessor directives begin with a  
A)#

13.The following data  
72  
'A'  
"Hello World"  
2.8712  
are all examples of

B)Literals or constants

14.A group of statements, such as the contents of a function, is enclosed in  
A)Braces{}  
  
15.Which of the following are*not* valid assignment statements? (Circle all that apply.)  
C)profit = 129

;  
16.Which of the following are*not* validcout statements? (Circle all that apply.)  
B)cout << "Have a nice day"\n;  
D)cout << Programming is great fun;

17.Assumew = 5,x = 4,y = 8, andz = 2. What value will be stored inresult in each of  
the following statements?  
A)result = x + y; = 12  
B)result = z \* 2; = 4  
C)result = y / x; = 2  
D)result = y − z; = 6  
E)result = w % 2; = 1 because this is mod  
18.How would each of the following numbers be represented in E notation?  
A)3.287 ×106  
  
19.The negation operator is  
A) Unary *Only need one parameter*  
  
20.A(n) \_\_\_\_\_\_\_\_\_\_\_ is like a variable, but its value is read-only and cannot be changed  
during the program’s execution.  
C)named constant  
  
21.When do preprocessor directives execute?  
A)Before the compiler compiles your program

**True or False**  
22.T F A variable must be defined before it can be used.  
23.T F Variable names may begin with a number.  
24.T F Variable names may be up to 31 characters long.  
25.T FA left brace in a C++ program should always be followed by a right brace later  
in the program.  
26.T FYou cannot initialize a named constant that is declared with theconst modifier.

**Chapter 16**

16.1What is the difference between a try block and a catch block?

The first part of the construct is the*try block*. This starts with the key wordtry and is followed by a block of code executing any statements that might directly or indirectly cause an  
exception to be thrown. The try block is immediately followed by one or more*catch blocks*,  
which are the exception handlers. A catch block starts with the key wordcatch, followed  
by a set of parentheses containing the definition of an exception parameter. The catch block shown catches the error message in theexceptionString parameter and then displays it withcout.

16.2What happens if an exception is thrown, but not caught?

There are two possible ways for a thrown exception to go uncaught. The first possibility is  
for the try/catch construct to contain no catch blocks with an exception parameter of the  
right data type. The second possibility is for the exception to be thrown from outside a try  
block. In either case, the exception will cause the entire program to abort execution.

16.3If multiple exceptions can be thrown, how does the catch block know which  
exception to catch?

16.4After the catch block has handled the exception, where does program execution resume?

16.5How can an exception pass data back to the exception handler?