**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.

**Fill-in-the-Blank**  
8.Computers can do many different jobs because they can be \_programmed\_.  
9.The job of the \_CPU\_\_\_\_ is to fetch instructions, carry out the operations commanded by the instructions, and produce some outcome or resultant information.  
10.Internally, the CPU consists of the \_control unit\_ and the \_aritmetic and logic unit (ALU)\_.  
11.A(n) \_USB drive\_ is an example of a secondary storage device.  
12.The two general categories of software are \_system\_\_\_ and \_application\_.  
13.A program is a set of \_instruction\_.  
14.Since computers can’t be programmed in natural human language, algorithms must  
be written in a(n) \_machine\_ language.  
15.\_Programming language\_ is the only language computers really process.  
16.\_High-level\_\_ languages are close to the level of humans in terms of readability.  
17.\_Low-level\_ languages are close to the level of the computer.  
18.A program’s ability to run on several different types of computer systems is called  
\_portability\_.  
19.Words that have special meaning in a programming language are called \_object code\_\_.  
20.Words or names defined by the programmer are called \_source code\_.  
21.\_\_\_\_\_\_\_\_\_\_ are characters or symbols that perform operations on one or more  
operands.  
  
22.\_\_\_\_\_\_\_\_\_\_ characters or symbols mark the beginning or ending of programming  
statements, or separate items in a list.  
23.The rules that must be followed when constructing a program are called \_\_\_\_\_\_\_\_\_\_.  
24.A(n) \_\_\_\_\_\_\_\_\_\_ is a named storage location.  
25.A variable must be \_\_\_\_\_\_\_\_\_\_ before it can be used in a program.  
26.The three primary activities of a program are \_\_\_\_\_\_\_\_\_\_, \_\_\_\_\_\_\_\_\_\_, and \_\_\_\_\_\_\_\_\_\_.  
27.\_\_\_\_\_\_\_\_\_\_ is information a program gathers from the outside world.  
28.\_\_\_\_\_\_\_\_\_\_ is information a program sends to the outside world.  
29.A(n) \_\_\_\_\_\_\_\_\_\_ is a diagram that graphically illustrates the structure of a program.

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

Float retailPrice, salesTaxRate;

Cout<<”Enter the retail price and the sales tax rate:”;

Cin>>retailPrice

>>salesTaxRate;

Cout<<”%”;

Float salesTax = retailPrice \* salesTaxRate / 100;

float total = retailPrice + salesTax;  
  
  
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.  
  
#include <iostream>

using namespace std;

int main()

{

int x = 0, y = 5;

x += 1;

y += 1;

y = x + y;

cout << y;

return 0;

}

Result = 7

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.   
  
#include <iostream>

using namespace std;

int main()

{

int j = 10, k = 2, l = 4;

j = j \* k;

l = k \* l;

k = j + l;

cout << k;

return 0;

}

Result = 28

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.

#include <iostream>

using namespace std;

int main()

{

int a = 1, b = 10, c = 100, x = 0;

x = c \* 3;

x = b \* 6 + x;

x = a \* 5 + x;

cout << x;

return 0;

}  
  
Result = 365  
  
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.