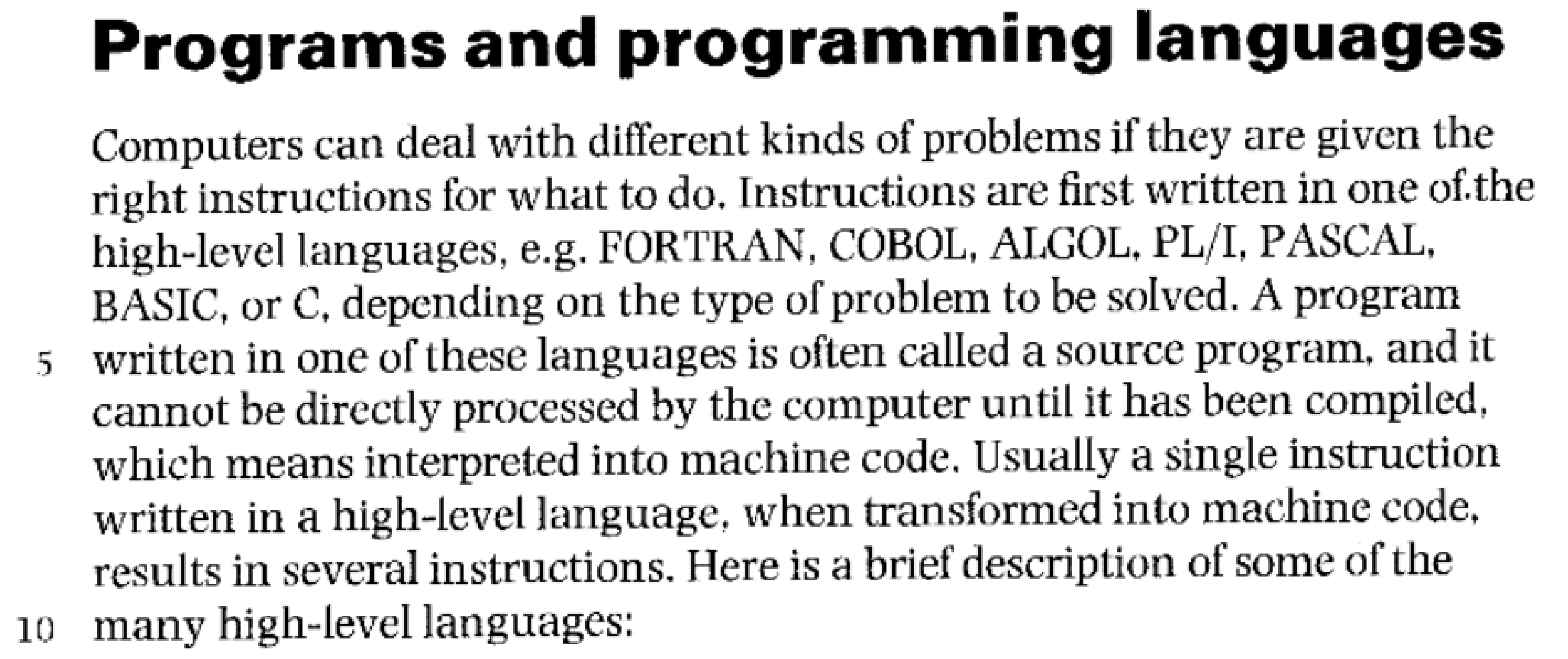
Technical English

for Computer Science

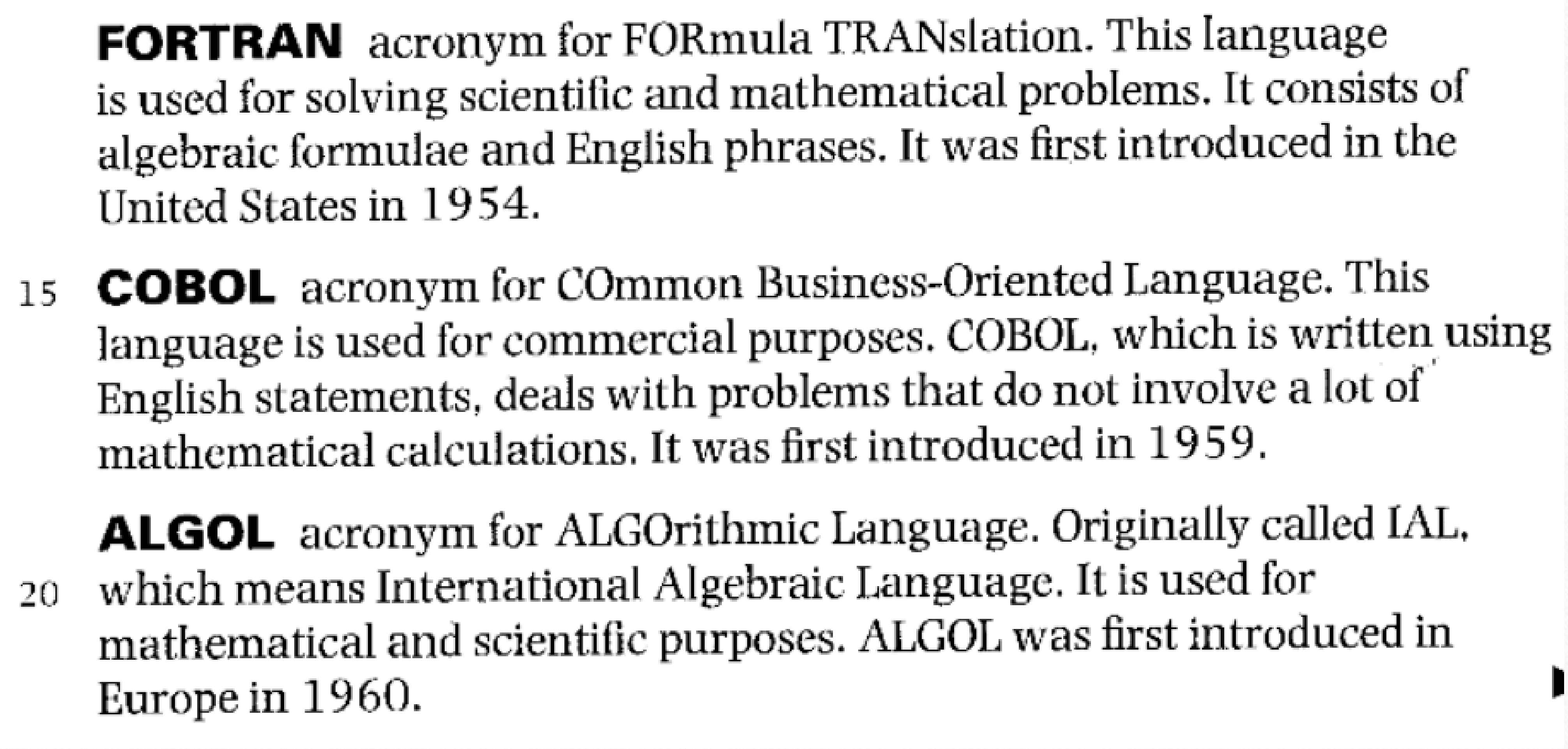
Week 3

Asst. Prof. Deniz DURAL BALTA



Programs and programming languages

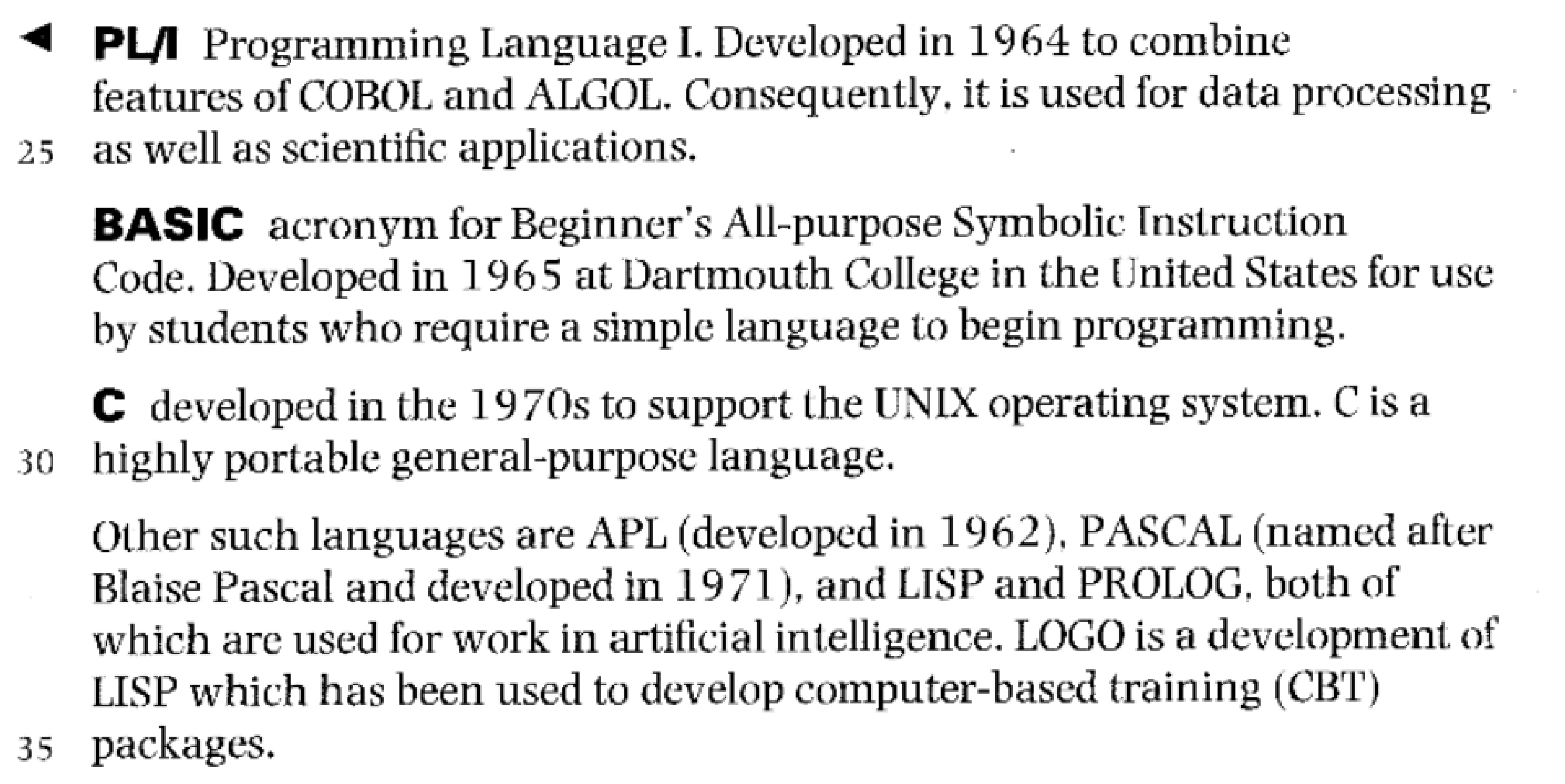
Computers can deal with different kinds of problems if they are given the right instructions for what to do. Instructions are first written in one of.the high-level languages, e.g. FORTRAN, COBOL, ALGOL, PL/I, PASCAL, BASIC, or C, depending on the type of problem to be solved. A program written in one of these languages is often called a source program, and it cannot be directly processed by the computer until it has been compiled, which means interpreted into machine code. Usually a single instruction written in a high-level language, when transformed into machine code, results in several instructions. Here is a brief description of some of the many high-level languages:



FORTRAN acronym for FORmula TRANslation. This language is used for solving scientific and mathematical problems. It consists of algebraic formulae and English phrases. It was first introduced in the United States in 1954.

COBOL acronym for COmmon Business-Oriented Language. This language is used for commercial purposes. COBOL, which is written using English statements, deals with problems that do not involve a lot of mathematical calculations. It was first introduced in 1959.

ALGOL acronym for ALGOrithmic Language. Originally called IAL, which means International Algebraic Language. It is used for mathematical and scientific purposes. ALGOL was first introduced in Europe in 1960.

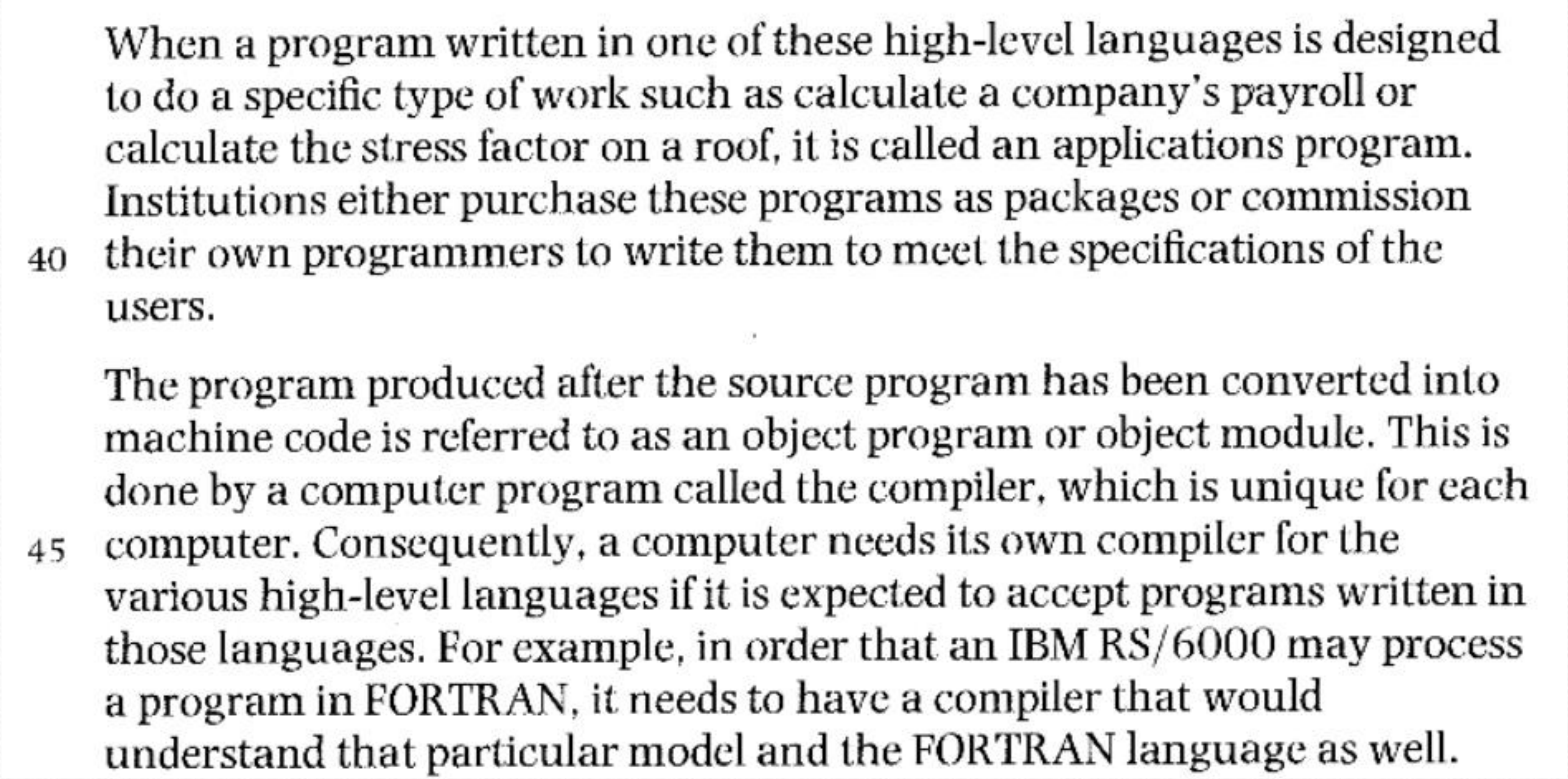


◄ **PL/I** Programming Language I. Developed in 1964 to combine features of COBOL and ALGOL. Consequently, it is used for data processing as well as scientific applications.

BASIC acronym for Beginner's All-purpose Symbolic Instruction Code. Developed in 1965 at Dartmouth College in the United States for use by students who require a simple language to begin programming.

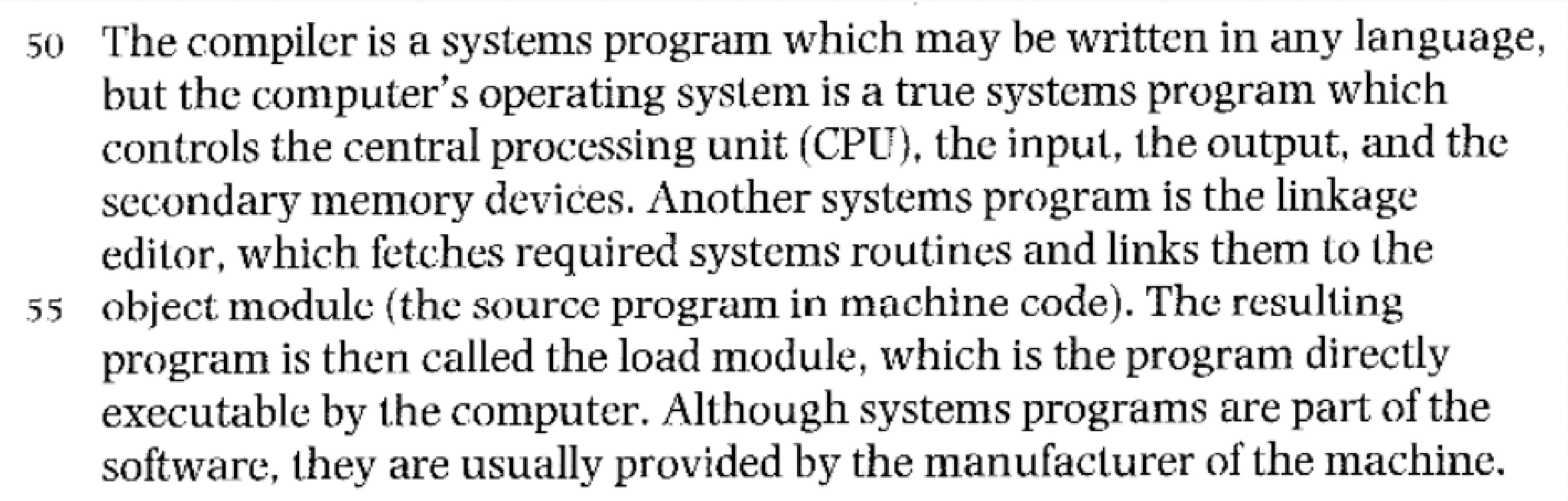
C developed in the 1970s to support the UNIX operating system. C is a highly portable general-purpose language.

Other such languages are APL (developed in 1962), PASCAL (named after Blaise Pascal and developed in 1971), and LISP and PROLOG, both of which are used for work in artificial intelligence. LOGO is a development of LISP which has been used to develop computer-based training (CBT) packages.

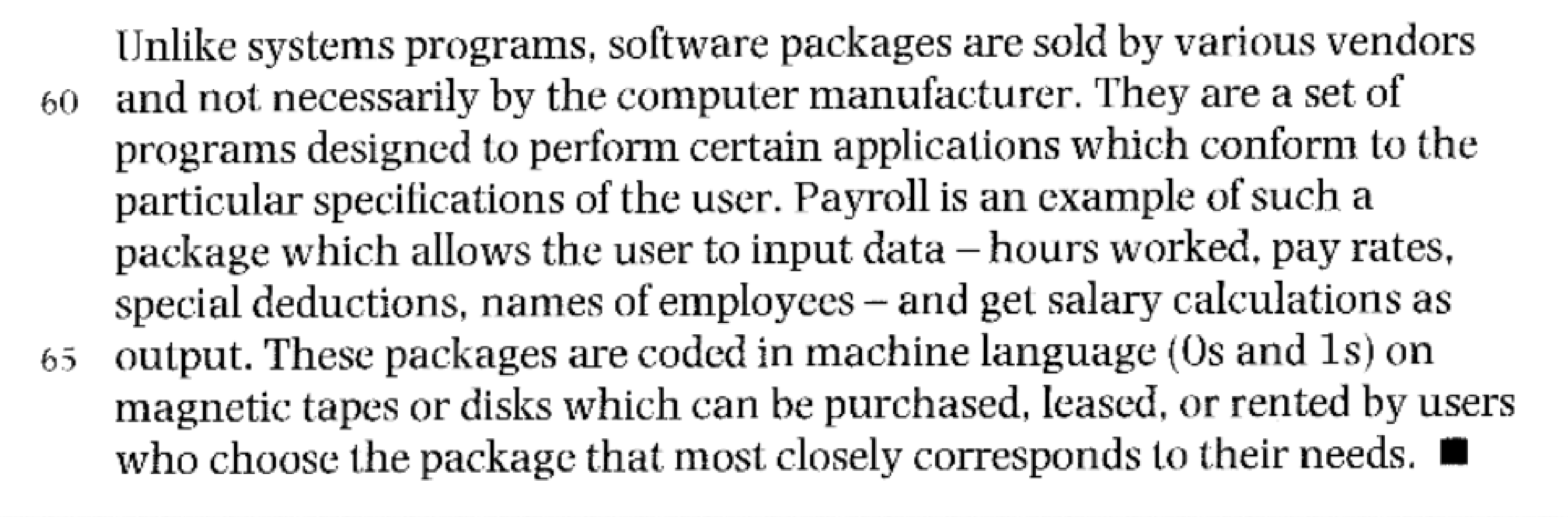


When a program written in one of these high-level languages is designed to do a specific type of work such as calculate a company's payroll or calculate the stress factor on a roof, it is called an applications program. Institutions either purchase these programs as packages or commission their own programmers to write them to meet the specifications of the users.

The program produced after the source program has been converted into machine code is referred to as an object program or object module. This is done by a computer program called the compiler, which is unique for each computer. Consequently, a computer needs its own compiler for the various high-level languages if it is expected to accept programs written in those languages. For example, in order that an IBM RS/6000 may process a program in FORTRAN, it needs to have a compiler that would understand that particular model and the FORTRAN language as well.

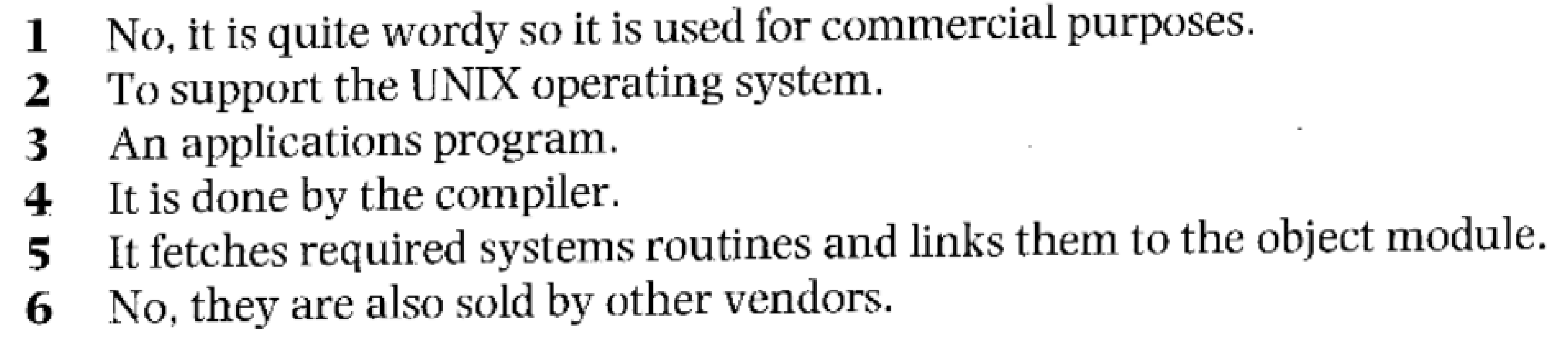


The compiler is a systems program which may be written in any language, but the computer's operating system is a true systems program which controls the central processing unit (CPU), the input, the output, and the secondary memory devices. Another systems program is the linkage editor, which fetches required systems routines and links them to the object module (the source program in machine code). The resulting program is then called the load module, which is the program directly executable by the computer. Although systems programs are part of the software, they are usually provided by the manufacturer of the machine.

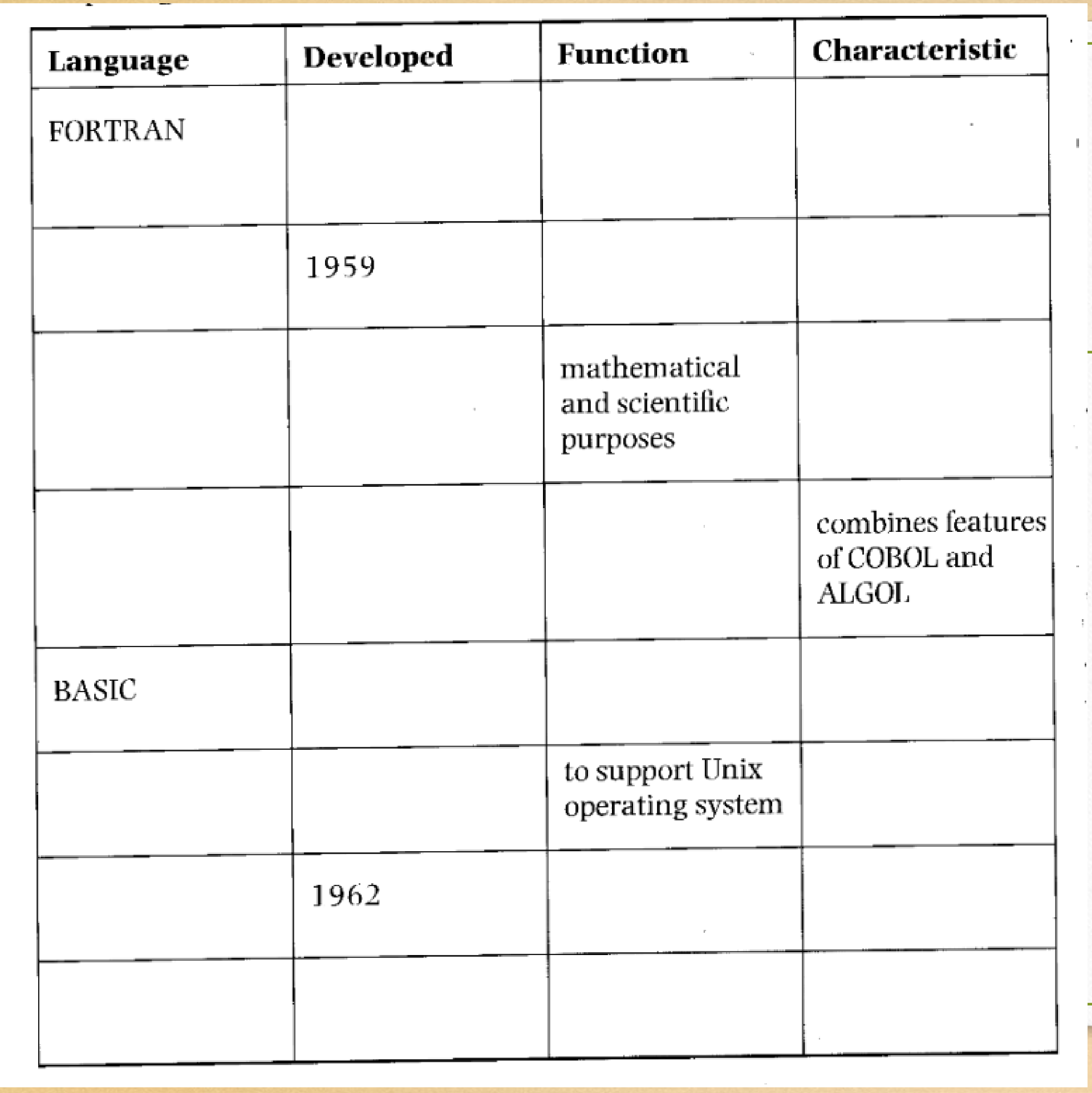


Unlike systems programs, software packages are sold by various vendors and not necessarily by the computer manufacturer. They are a set of programs designed to perform certain applications which conform to the particular specifications of the user. Payroll is an example of such a package which allows the user to input data – hours worked, pay rates, special deductions, names of employees – and get salary calculations as output. These packages are coded in machine language (0s and 1s) on magnetic tapes or disks which can be purchased, leased, or rented by users who choose the package that most closely corresponds to their needs. ■

These are answers the questions about the text. Write their questions.



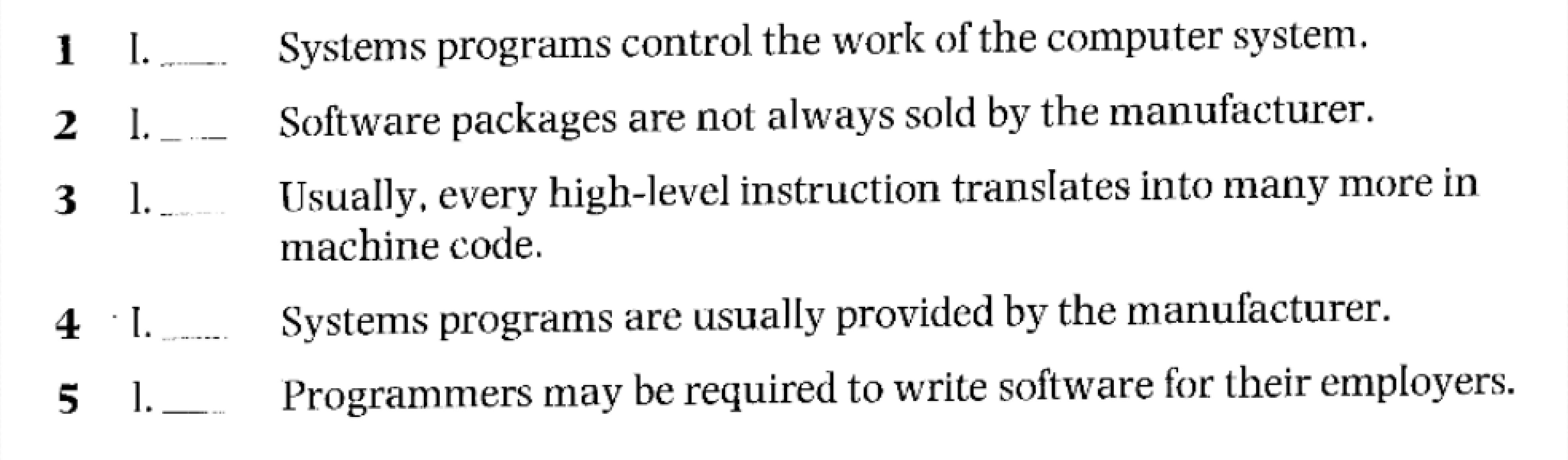
1. No, it is quite wordy so it is used for commercial purposes.
2. To support the UNIX operating system.
3. An applications program.
4. It is done by the compiler.
5. It fetches required systems routines and links them to the object module.
6. No, they are also sold by other vendors.



Language	Developed	Function	Characteristic
FORTRAN			
	1959		
		mathematical and scientific purposes	
			combines features of COBOL and ALGOL
BASIC			
		to support Unix operating system	
	1962		

- **Summarize the information on different high-level computer languages by completing the table below.**

Find the passages in the text where the following ideas are expressed. Give the line references.



1 l. ____ Systems programs control the work of the computer system.

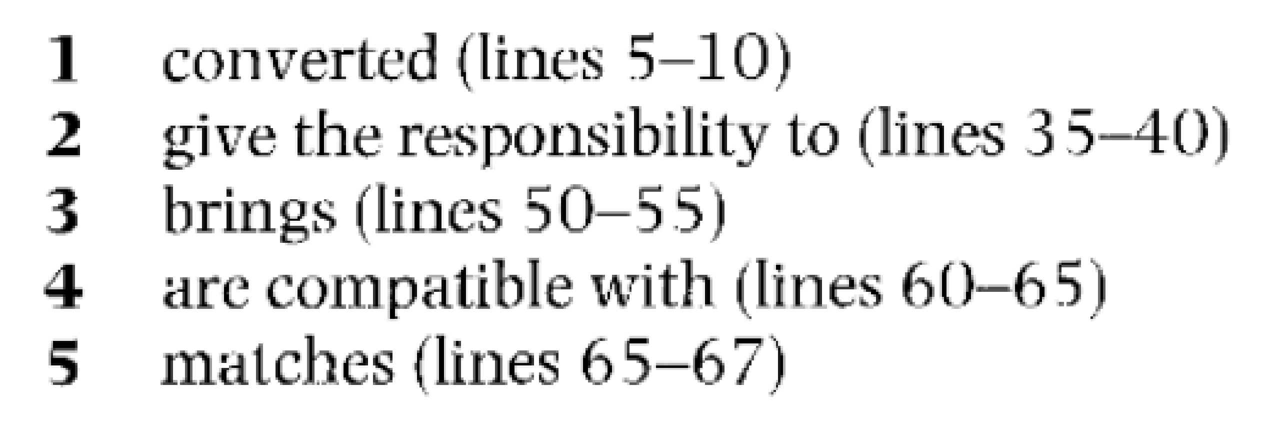
2 l. ____ Software packages are not always sold by the manufacturer.

3 l. ____ Usually, every high-level instruction translates into many more in machine code.

4 l. ____ Systems programs are usually provided by the manufacturer.

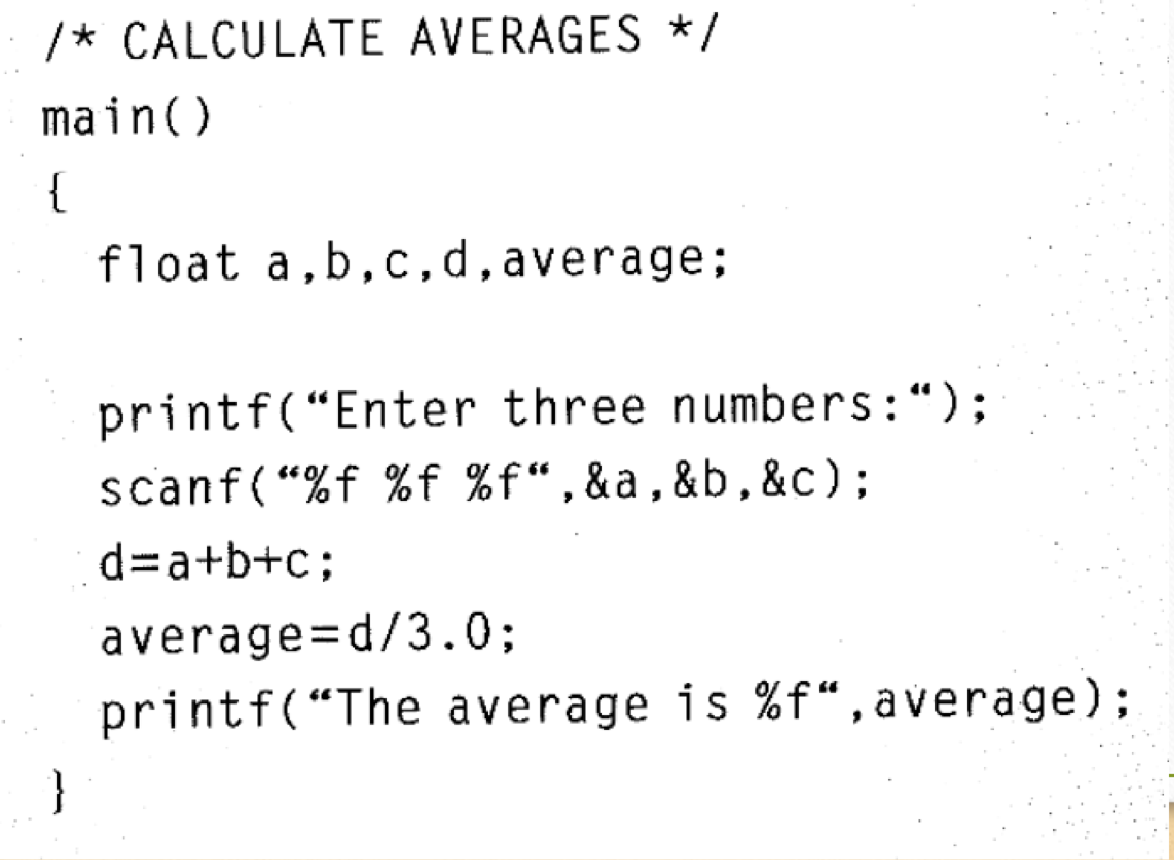
5 l. ____ Programmers may be required to write software for their employers.

Using the line references given, refer back to the text and find words or phrases that have a similar meanings to:



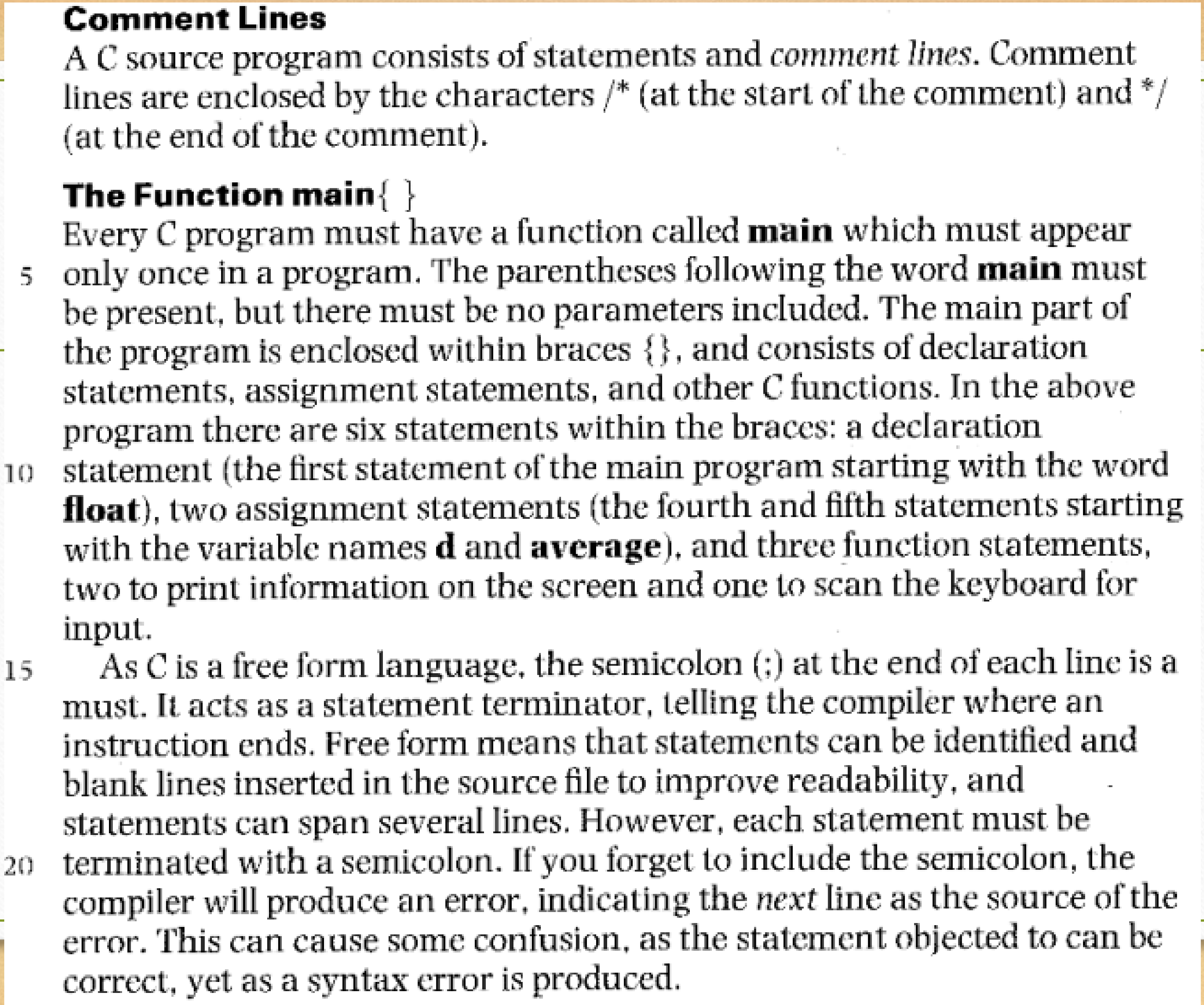
1. converted (lines 5–10)
2. give the responsibility to (lines 35–40)
3. brings (lines 50–55)
4. are compatible with (lines 60–65)
5. matches (lines 65–67)

Read the program and text, then complete the sentences.



```
/* CALCULATE AVERAGES */
main()
{
  float a,b,c,d,average;

  printf(“Enter three numbers:“);
  scanf(“%f %f %f“,&a,&b,&c);
  d=a+b+c;
  average=d/3.0;
  printf(“The average is %f“,average);
}
```



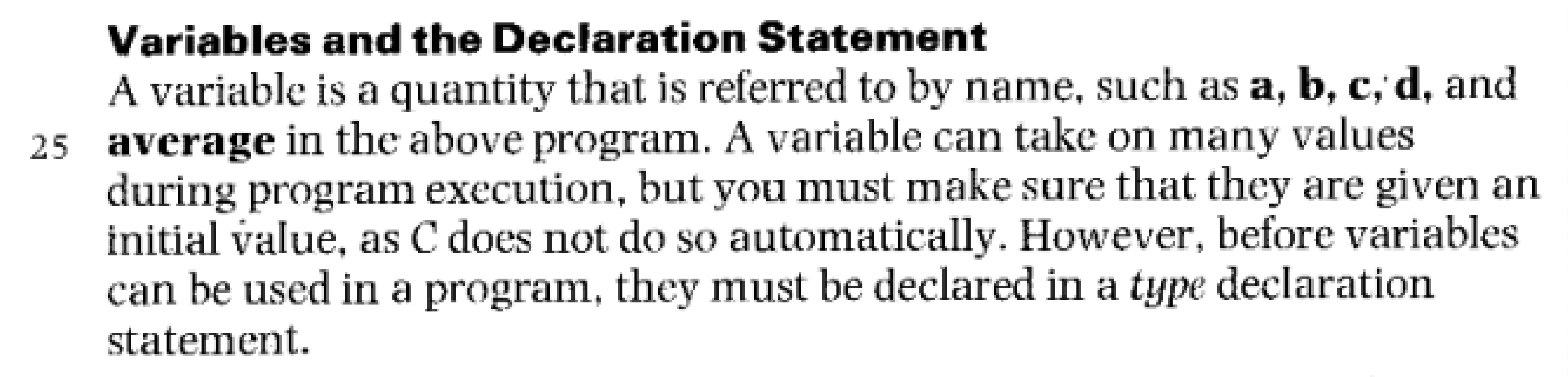
Comment Lines

A C source program consists of statements and *comment lines*. Comment lines are enclosed by the characters /* (at the start of the comment) and */ (at the end of the comment).

The Function main{ }

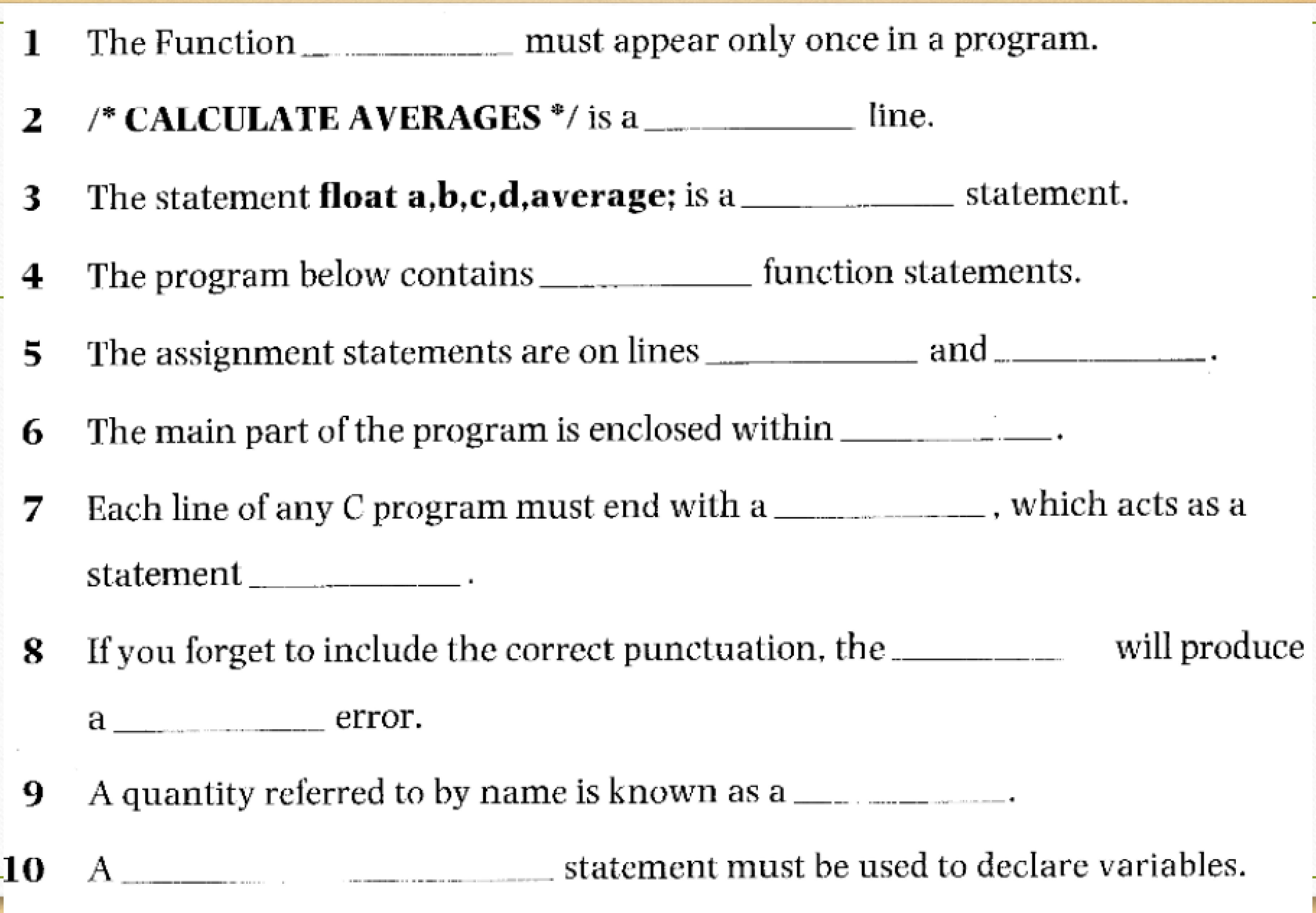
Every C program must have a function called **main** which must appear only once in a program. The parentheses following the word **main** must be present, but there must be no parameters included. The main part of the program is enclosed within braces {}, and consists of declaration statements, assignment statements, and other C functions. In the above program there are six statements within the braces: a declaration statement (the first statement of the main program starting with the word **float**), two assignment statements (the fourth and fifth statements starting with the variable names **d** and **average**), and three function statements, two to print information on the screen and one to scan the keyboard for input.

As C is a free form language, the semicolon (;) at the end of each line is a must. It acts as a statement terminator, telling the compiler where an instruction ends. Free form means that statements can be identified and blank lines inserted in the source file to improve readability, and statements can span several lines. However, each statement must be terminated with a semicolon. If you forget to include the semicolon, the compiler will produce an error, indicating the *next* line as the source of the error. This can cause some confusion, as the statement objected to can be correct, yet as a syntax error is produced.



Variables and the Declaration Statement

A variable is a quantity that is referred to by name, such as **a, b, c,' d,** and **average** in the above program. A variable can take on many values during program execution, but you must make sure that they are given an initial value, as C does not do so automatically. However, before variables can be used in a program, they must be declared in a *type* declaration statement.



1 The Function ____________ must appear only once in a program.

2 /* **CALCULATE AVERAGES** */ is a ____________ line.

3 The statement **float a,b,c,d,average;** is a ____________ statement.

4 The program below contains ____________ function statements.

5 The assignment statements are on lines ____________ and ____________.

6 The main part of the program is enclosed within ____________.

7 Each line of any C program must end with a ____________, which acts as a statement ____________.

8 If you forget to include the correct punctuation, the ____________ will produce a ____________ error.

9 A quantity referred to by name is known as a ____________.

10 A ____________ ____________ statement must be used to declare variables.