# CHAPTER 9

# *Programming Languages*

## Review Questions

1. Machine language is the only language understood by a computer (1s and 0s). A symbolic language uses symbols to represent the various machine language instructions.
2. A special program called an assembler must be used to translate a symbolic language into machine language. Because symbolic languages need to be 'assembled' into machine language, they are also known as assembly languages.
3. High level languages are portable to many different computers and allow the programmer to focus on the problem being solved rather than the intricacies of the computer for which the program is being written.
4. Some high-level languages are BASIC, Pascal, Ada, C, C++, and Java.
5. A natural language is a language spoken by humans, like English, French, or Japanese.
6. A source file is the program as it is written by the programmer using a text editor of some sort and is the input to the compiler. An executable file is the program once it has been compiled and is a collection of machine language that is executable by the computer.
7. A compiler is a program that translates the source file into machine language that a computer can then execute.
8. A linker assembles the output of the compiler (the object module) and other object modules including library subroutines into a final executable program.
9. The five categories of computer languages are: procedural (FORTRAN, COBOL, Pascal, C, and Ada), object-oriented (C++ and Java), functional (LISP and Scheme), declarative (Prolog), and special (HTML, PERL, and SQL).
10. Java is based on C and C++.
11. C has all the high-level instructions a structured high-level language should have. C has some low-level instructions that make it a good language for system programmers. C is a very efficient and concise language. C has been standardized by ANSI and ISO.

12. Ada was created by the Department of Defense to be the uniform language used by all DoD contractors.
13. In procedural programming, the data items are separate from the instructions that act on them. In an object oriented language, the data is encapsulated with the functions that act on that data.
14. Encapsulation, inheritance, and polymorphism.
15. A functional programming language is one that looks at programs as a mathematical function. It allows a programmer to combine predefined primitive functions to create new functions.
16. A declarative language uses the principle of logical reasoning to answer queries.
17. Special languages are languages that are either a mixture of the other models or languages that can be applied to a specific task.
18. 'int', 'char', and 'float'
19. An initializer provides an initial value for a variable when it is declared and defined.
20. A literal constant is an unnamed value that is used in a program as is, such as a number or a string literal like “Hello”. A named constant is a value that we store in memory that the program may not change during its execution. A symbolic constant is defined at the beginning of a program and allows the compiler to replace all of the references to that constant during preprocessing.
21. Variables are names for memory locations.
22. These functions can be included in library files that can be compiled separately, included in a program, and used as if the functions were a part of the program.
23. <, <=, >, >=, ==, and !=
24. !, &&, and ||
25. =, +=, -=, *=, /=, and %=
26. One and only one function in a C program must be called 'main'. The execution of the program always starts and ends with 'main'.
27. The function header defines the return type, the name of the function, and the number and type of the required parameters. The function body contains the instructions which comprise the function, including a return statement. The function call invokes the function, supplying any required parameters and perhaps saving the return value of the function.
28. A selection statement is used to conditionally execute a statement or block of code.
29. The condition tested by an if-else statement can be anything that evaluates to a boolean (true or false) value. The expression used to determine the program flow in a switch statement must evaluate to one of the C integral types.
30. The 'while' loop is a pretest loop and is the main repetition construct used in the C language. It tests the value of a testing expression. The 'for' loop is also a pretest loop used for counter controlled loops. The 'do-while' loop is a posttest loop and is used like a 'while' loop when at least 1 iteration is required.

## Multiple-Choice Questions

31. a
32. b
33. d
34. c
35. b
36. c
37. c
38. b
39. a
40. a
41. a
42. a
43. d
44. b
45. d
46. a
47. c
48. b
49. a
50. c
51. a
52. d
53. b
54. c
55. b
56. d
57. c
58. a
59. b

## Exercises

60. Research question
61. Research question
62. Research question
63. Research question
64. Research question
65. Research question
66. Research question

67. Research question
68.
    a. Invalid - Starts with a number
    b. Valid
    c. Invalid - Contains space
    d. Invalid - Starts with a number
69.
    a. int i;
    b. float f;
    c. char c;
70.
    a. int i = 12;
    b. float f = 12.32;
    c. char c = 'A'
71. scanf ("%d %d", &int_1, &int_2);
72. printf ("%d %d", int_1, int_2);
73.
    a. z = x * y;
    b. x++;
    c. y--;
    d. if (x == y)...
74.

```
int i;
for (i = 0 ; i < 10 ; i++ )
{
    printf ( "Hello World!\n" );
}
```

75.

```
int i = 0;
while ( i < 10 )
{
    printf ( "Hello World!\n" );
    i++;
}
```

76.

```
int i = 0;
do
{
    printf ( "Hello World!\n" );
```

```
        i++;
    } while ( i < 10 );
```

77. ( (car(cdr(cdr list))) (car(cdr(cdr(cdr list)))))