

Programming Languages

(Solutions to Review Questions and Problems)

Review Questions

- Q9-1.** A machine language uses only 0s and 1s for instructions and addresses. An assembly language uses symbols to represent instructions and addresses.
- Q9-3.** The machine language is the only language understood by the computer hardware.
- Q9-5.** The four steps are lexical analysis, syntax analysis, semantic analysis, and code generation.
- Q9-7.** In the procedural paradigm, a program is an active agent that manipulates passive objects (data). In an object-oriented paradigm, data are designed as active objects. The action to be performed on these objects are included in the object.
- Q9-9.** In the functional paradigm a program is designed like a mathematical function. It allows the programmer to combine predefined primitive functions to create new functions.

Problems

P9-1.

```
int count;  
int index;  
int level;
```

P9-3.

```
const char name = 'A';  
const int count = 1;  
const float height = 1.82;
```

P9-5. The statement is executed twice (once when $A = 5$ and the second time when $A = 7$). When A becomes 9, the loop is terminated.

P9-7. The statement is executed eight times ($i = 5, 7, 9, 11, 13, 15, 17, 19$). Note that in each iteration the value of i is incremented twice: the first time inside the header ($i++$), the second time in the body of the loop ($i = i + 1$).

P9-9. Table 9.1 shows the solution.

Table 9.1 *Solution to P9-9*

```
A = 5;
do
{
    statement;
    A = A - 2;
} while (A < 8);
```

P9-11. Table 9.2 shows the solution

Table 9.2 *Solution to P9-11*

```
i = 5;
while (i < 20)
{
    statement;
    i = i + 2;
}
```

P9-13. Table 9.3 shows the solution.

Table 9.3 *Solution to P9-13*

```
for (int A = 5; A < 8; A = A - 2)
{
    statement;
}
```

P9-15. This is not possible because in a *do-while* loop, the body of the loop is executed at least once.

P9-17. Table 9.4 shows one possible solution

Table 9.4 *Solution to P9-17*

```
while (true)
{
    statement;
}
```

P9-19. Table 9.5 shows one possible solution.

Table 9.5 *Solution to P9-19*

```
for (; true ; )
{
    statement;
}
```

P9-21. *Hello* is the variable, "Hello" is the literal.

P9-22. Table 9.6 shows one solution. Note that we don't need the last *break* statement, but it is normally included to make each case the same.

Table 9.6 *Solution to P9-22*

```
switch (A)
{
    case 4:    statement 1;
              break;
    case 6:    statement 2;
              break;
    case 8:    statement 3;
              break;
}
```

P9-23. A and B should be passed by value, S and P by reference.

P9-25.

- a. It should be by reference if we can allow the subprogram change the value of A in the main program. The following shows the statement:

```
cube (A);
```

- b. Alternatively, we can pass A by value and let the function **cube** return the cube of A. In this case, the original value of A remains untouched in the main program. The following shows the statement:

```
result ← cube (A);
```

P9-27. It can be passed either by value or by reference, but it is normally passed by value to keep the value of the variable in the main untouched.

