

Names, Scopes, and Bindings

3.10 Exercises

- 3.23 Assuming a LeBlanc-Cook style symbol table, explain how the compiler finds the symbol table information (e.g., the type) of a complicated reference such as `my_firm^.revenues[1999]`.
- 3.24 Show the contents of a LeBlanc-Cook style symbol table that captures the referencing environment of
 - (a) function F1 in Figure 3.4 (page 126).
 - (b) procedure `pop` in Figure 3.7 (page 136).
- 3.25 Show a trace of the contents of the referencing environment A-list during execution of the program in
 - (a) Figure 3.9 (page 140). Assume that a positive value is read at line 8.
 - (b) Exercise 3.14.
- 3.26 Repeat the previous exercise for a central reference table.
- 3.27 Consider the following tiny program in C:

```
void hello() {  
    printf("Hello, world\n");  
}  
  
int main() {  
    hello();  
}
```

- (a) Split the program into two separately compiled files, `tiny.c` and `hello.c`. Be sure to create a header file `hello.h` and include it correctly in `tiny.c`.

- (b) Reconsider the program as C++ code. Put the `hello` function in a separate namespace, and include an appropriate using declaration in `tiny.c`.
- (c) Rewrite the program in Java, with `main` and `hello` in separate packages.

3.28 Consider the following file from some larger C program:

```
int a;
extern int b;
static int c;

void foo() {
    int a;
    static int b;
    extern int c;
    extern int d;
}

static int b;
extern int c;
```

For each variable declaration, indicate whether the variable has external linkage, internal (file-level) linkage, or no linkage (i.e., is local).

3.29 Modula-2 provides no way to divide the header of a module into a public part and a private part: everything in the header is visible to the users of the module. Is this a major shortcoming? Are there disadvantages to the public/private division (e.g., as in Ada)? (For hints, see Section 9.2.)