which not only declares `i` to be a variable of type `int`, but defines `i` as well, by causing the compiler to set aside space for `i`. To declare `i` without defining it, we must put the keyword `extern` at the beginning of its declaration:

extern keyword ➤ 18.2

```
extern int i;   /* declares i without defining it */
```

`extern` informs the compiler that `i` is defined elsewhere in the program (most likely in a different source file), so there's no need to allocate space for it.

`extern` works with variables of all types. When we use it in the declaration of an array, we can omit the length of the array:

Q&A

```
extern int a[];
```

Since the compiler doesn't allocate space for `a` at this time, there's no need for it to know `a`'s length.

To share a variable `i` among several source files, we first put a definition of `i` in one file:

```
int i;
```

If `i` needs to be initialized, the initializer would go here. When this file is compiled, the compiler will allocate storage for `i`. The other files will contain declarations of `i`:

```
extern int i;
```

By declaring `i` in each file, it becomes possible to access and/or modify `i` within those files. Because of the word `extern`, however, the compiler doesn't allocate additional storage for `i` each time one of the files is compiled.

When a variable is shared among files, we'll face a challenge similar to one that we had with shared functions: ensuring that all declarations of a variable agree with the definition of the variable.

When declarations of the same variable appear in different files, the compiler can't check that the declarations match the variable's definition. For example, one file may contain the definition

```
int i;
```

while another file contains the declaration

```
extern long i;
```

An error of this kind can cause the program to behave unpredictably.

To avoid inconsistency, declarations of shared variables are usually put in header files. A source file that needs access to a particular variable can then include the appropriate header file. In addition, each header file that contains a