Additional formatting stuff

Your C compiler's documentation should contain a list of additional `printf()` formatting information, bonus characters that can be used in conjunction with the conversion characters to additionally format `printf()`'s output. That information is too complex and detailed to list here for every compiler. Instead, look up `printf` in your online reference manual and note these formatting sections:

- Flags
- Width specifiers
- Precision specifiers
- Input-size modifiers

You don't need this information now for understanding the C programming language. However, it comes in handy as you begin working with numbers or require a little fancier output than what you have done with `printf()` in this chapter.

In addition to the conversion characters in Table 24-2, three other characters exist: `%p`, `%n` and `%%`. The `%p` and `%n` are advanced conversion characters, beyond the scope of this book. The `%%` merely prints a `%` on the screen.

- As with the escape sequences, the conversion characters are something you use often but never remember. I advise tacking a sticky note to this page for future reference.
- The `%x`, `%e`, and `%g` conversion characters also have uppercase equivalents: `%X`, `%E` and `%G`. By using capital letters rather than lowercase, you ensure that any letters in the output are all listed as uppercase. For example, `%x` would display a hexadecimal value as `1ba2`, but `%X` would display `1BA2`. Otherwise, the conversion character's behavior is the same.
- `%p` is used to print a value as a pointer. Pointers are covered in this book's companion, *C All-in-One Desk Reference For Dummies* (Wiley).