There are many other specifiers besides %d, %e, %f, and %g. I'll gradually introduce many of them in subsequent chapters. For the full list, and for a complete explanation of the other capabilities of conversion specifications, consult Section 22.3.

PROGRAM   Using `printf` to Format Numbers

The following program illustrates the use of `printf` to print integers and floating-point numbers in various formats.

tprintf.c

```
/* Prints int and float values in various formats */

#include <stdio.h>

int main(void)
{
  int i;
  float x;

  i = 40;
  x = 839.21f;

  printf("|%d|%5d|%-5d|%5.3d|\n", i, i, i, i);
  printf("|%10.3f|%10.3e|%-10g|\n", x, x, x);

  return 0;
}
```

The | characters in the `printf` format strings are there merely to help show how much space each number occupies when printed; unlike % or \, the | character has no special significance to `printf`. The output of this program is:

```
|40|   40|40   |  040|
|   839.210| 8.392e+02|839.21    |
```

Let's take a closer look at the conversion specifications used in this program:

- %d — Displays i in decimal form, using a minimum amount of space.

- %5d — Displays i in decimal form, using a minimum of five characters. Since i requires only two characters, three spaces were added.

- %-5d — Displays i in decimal form, using a minimum of five characters; since the value of i doesn't require five characters, the spaces are added afterward (that is, i is left-justified in a field of length five).

- %5.3d — Displays i in decimal form, using a minimum of five characters overall and a minimum of three digits. Since i is only two digits long, an extra zero was added to guarantee three digits. The resulting number is only three characters long, so two spaces were added, for a total of five characters (i is right-justified).

- %10.3f — Displays x in fixed decimal form, using 10 characters overall,