

FIGURE 2-18 Conversion Specification

Type	Size ^a	Code	Example
char	None	c	%c
short int	h	d	%hd
int	None	d	%d
long int	None	d	%ld
long long int	ll	d	%lld
float	None	f	%f
double	None	f	%f
long double	L	f	%Lf
a. Size is discussed in the next section.			

Table 2-10 Format Codes for Output

Flag Type	Flag Code	Formatting
Justification	None	right justified
	–	left justified
Padding	None	space padding
	0	zero padding
Sign	None	positive value: no sign negative value: –
	+	positive value: + negative value: –
	Space	positive value: space negative value: –

Table 2-11 Flag Formatting Options

Output Examples

1. `printf("%d%c%f", 23, 'z', 4.1);`

```
23z4.100000
```

2. `printf("%d %c %f", 23, 'z', 4.1);`

```
23 z 4.100000
```

3. `int num1 = 23;`
`char zee = 'z';`
`float num2 = 4.1;`
`printf("%d %c %f", num1, zee, num2);`

```
23 z 4.100000
```

Output Examples

```
4. printf("%d\t%c\t%5.1f\n", 23, 'z', 14.2);  
   printf("%d\t%c\t%5.1f\n", 107, 'A', 53.6);  
   printf("%d\t%c\t%5.1f\n", 1754, 'F', 122.0);  
   printf("%d\t%c\t%5.1f\n", 3, 'P', 0.1);
```

```
23      z      14.2  
107     A      53.6  
1754    F      122.0  
3       P      0.1
```

```
5. printf("The number%d is my favorite.", 23);
```

```
The number23 is my favorite.
```

```
6. printf("The number is %6d", 23);
```

```
The number is      23  
^^^^^^^^^^^^^^^^
```

Output Examples

7. `printf("The tax is %6.2f this year.", 233.12);`

```
The tax is 233.12 this year.
```

8. `printf("The tax is %8.2f this year.", 233.12);`

```
The tax is    233.12 this year.  
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

9. `printf("The tax is %08.2f this year.", 233.12);`

```
The tax is 00233.12 this year.
```

10. `printf("\">%8c %d\"", 'h', 23);`

```
"          h    23"  
^^^^^^^^^^^^^^^^
```

Output Examples

```
11. printf("This line disappears.\r...A new  
line\n");  
printf("This is the bell character \a\n");  
printf("A null character\0kills the rest of the  
line\n");  
printf("\nThis is \'it\' in single quotes\n");  
printf("This is \"it\" in double quotes\n");  
printf("This is \\ the escape character it  
self\n");
```

```
...A new line  
This is the bell character  
A null character  
This is 'it' in single quotes  
This is "it" in double quotes  
This is \ the escape character it self
```

Output Examples

```
12. printf(" |%+8.2f| |%0+8.2f| | %-0+8.2f|", 1.2,  
2.3, 3.4);
```

```
|+1.20      | |+0002.30| | +3.40      |  
^^^^^^^^^^| ^^^^^^^^^^^| ^^^^^^^^^^^|
```

Common Output Errors

1. `printf("%d %d %d\n", 44, 55);`

```
44 55 1638280
```

2. `printf("%d %d\n", 44, 55, 66);`

```
44 55
```

3. `float x = 123.45;`
`printf("The data are: %d\n", x);`

```
The data are: -1073741824
```

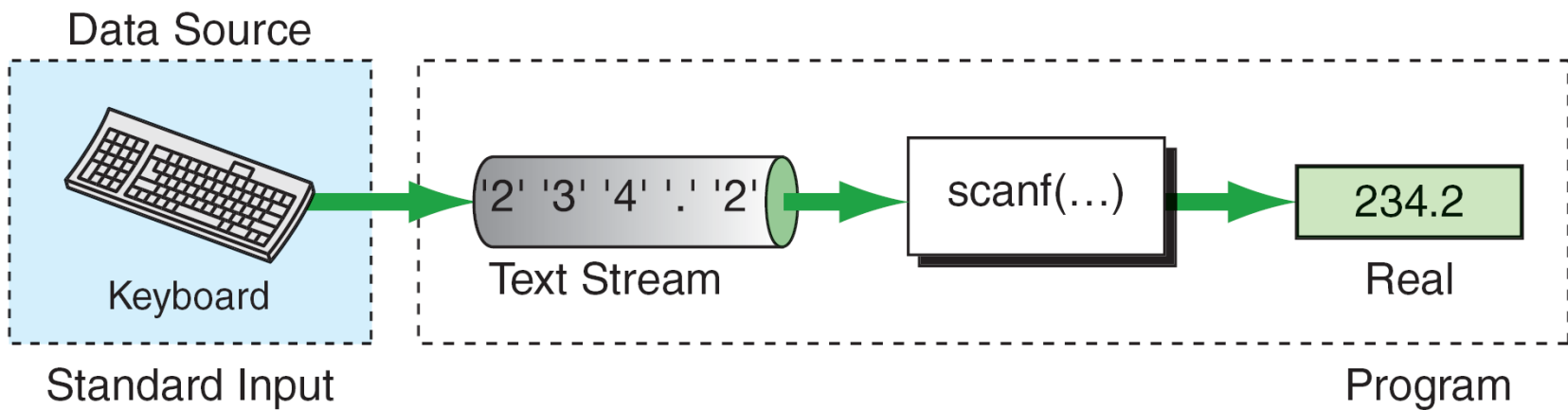
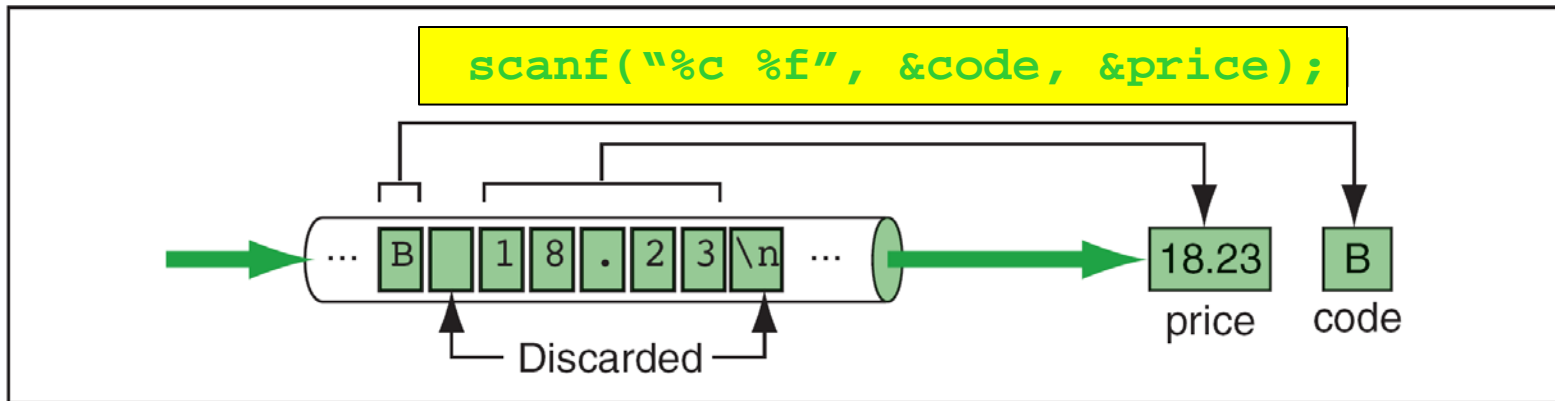
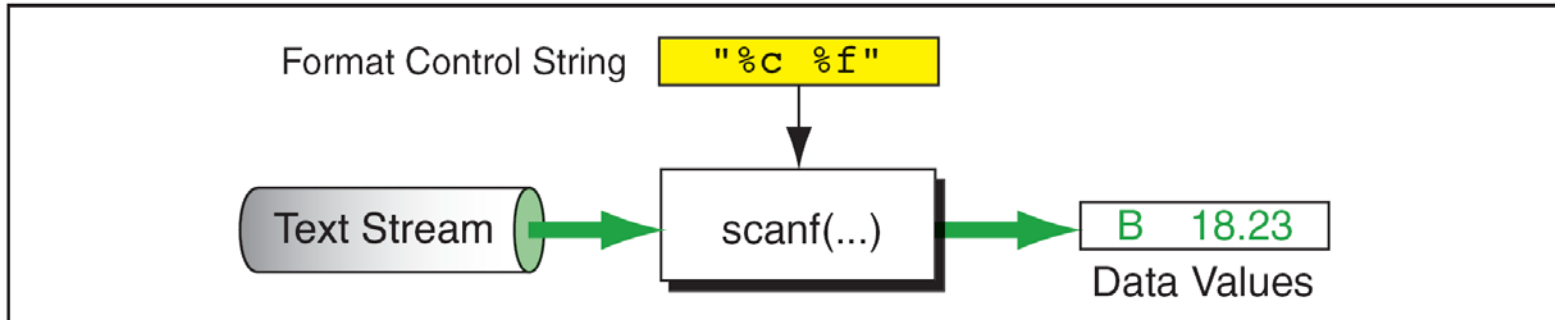


FIGURE 2-19 Formatting Text from an Input Stream

(a) Basic Concept



(b) Implementation

FIGURE 2-20 Input Stream Formatting Example



FIGURE 2-21 Conversion Specification

Note

***scanf* requires variable addresses in the address list.**

1. The conversion operation processes until:
 - a. End of file is reached.
 - b. The maximum number of characters has been processed.
 - c. A whitespace character is found after a digit in a numeric specification.
 - d. An error is detected.
2. There must be a conversion specification for each variable to be read.
3. There must be a variable address of the proper type for each conversion specification.
4. Any character in the format string other than whitespace or a conversion specification must be exactly matched by the user during input. If the input stream does not match the character specified, an error is signaled and *scanf* stops.
5. It is a fatal error to end the format string with a whitespace character. Your program will not run correctly if you do.

Table 2-12 *scanf* Rules

Input Examples

1-1. 214 156 14z

```
scanf("%d%d%d%c", &a, &b, &c, &d);
```

1-2. 214 156 14 z

```
scanf("%d%d%d %c", &a, &b, &c, &d);
```

2. 2314 15 2.14

```
scanf("%d %d %f", &a, &b, &c);
```

Input Examples

3. 14/26 25/66

```
scanf("%2d/%2d %2d/%2d", &num1, &den1, &num2,  
&den2);
```

4. 11-25-56

```
scanf("%d-%d-%d", &a, &b, &c);
```

Common Input Errors

```
1. int a=0;  
   scanf ("%d", a);  
   printf ("%d\n", a);
```

```
234      (Input)  
0        (Output)
```

```
2. float a = 2.1;  
   scanf ("%5.2f", &a);  
   printf ("%5.2f", a);
```

```
74.35      (Input)  
2.10      (Output)
```

Common Input Errors

```
3. int a;  
   int b;  
   scanf ("%d%d%d", &a, &b);  
   printf ("%d %d\n", a, b);
```

```
5 10          (Input)  
5 10          (Output)
```

```
4. int a = 1;  
   int b = 2;  
   int c = 3;  
   scanf ("%d%d",      &a, &b, &c);  
   printf ("%d %d\n", a, b, c);
```

```
5 10 15      (Input)  
5 10 3       (Output)
```