C PROGRAMMING FOR ENGINEERS

Input and Output Functions

printf Function

Prints values and text to the computer screen.

(has two arguments)

printf(" ",);

Control String region (must be enclosed by double quotation marks).
Can contain both text and conversion specifications (printing format).

Variable list region (variables are separated by commas).

Primary conversion specifiers for output:

%i	prints an integer value
%f	prints a floating-point value
%e	prints in exponent format
%c	prints character data

Output Examples

Example #1

```
int x=7;
printf("The x value is %i", x);
```

What appears on the computer screen:

```
The x value is 7
```

Example #2

```
double y=3.5;
printf("Y has been assigned %f", y);
```

What appears on the computer screen:

```
Y has been assigned 3.500000
```

Field Width and Precision

- Field width represents the number of horizontal spaces that are needed when printing a number, and would include blank spaces.

Example:

```
int x=7;
printf("The x value is %6i", x);
```

What appears on the computer screen:

```
The x value is     7
```

Notice that the printed number is right justified in the field. Thus, 5 blank spaces are inserted in front of the number.

The number 6 is the field width (meaning 6 print spaces are allocated to print the number).

Field Width and Precision Continued

- By default, the Field Width will increase if the number to be printed is larger than the specified number of print spaces, and the number is still right justified. However, to allow the default setting to control output would be unwise.

- For left justification a minus sign is placed in the specifier as illustrated below.

```
printf("The x value is %-6i",x);
```

A minus sign controls left justification.

Field Width and Precision Continued

- When dealing with integer numbers precision is not a real issue, but it is for floating-point numbers.

- Below are some examples using pi of 3.14159:

Specification	Value Printed	
%f	3.141590	(Defaults to 6 digits of precision.)
%6.4f	3.1416	(Field width of 6 with 4 digits of precision.)
%4.1f	␣3.1	(Field width of 4 with 1 digit of precision.)
	A blank space is added.	
%2.1f	3.1	(Field width of 2 with 1 digit of precision. The field width will increase to print the value.)

Escape Character

- Used for vertical control when printing.
- The backslash \ is the **Escape Character** (found above the Enter key).
- Used in the printf and fprintf functions.
- Has special meaning when attached to other characters.
- Example:

```
printf("\n\n");
```

When this print statement is executed, two blank lines are printed to the computer screen.

Commonly used sequences:

\n	new line (vertical spacing)
\\	used for accessing input and output files

scanf Function

- Allows the user to enter values from the keyboard during program execution.
- Example:

```
scanf("%i",&year);
```

This statement will read an integer number from the keyboard.

Conversion specifier (integer).

Variable

The & symbol is called the address operator. It determines the memory address of the variable. It is required for the scanf function to read a value.

The above statement will read an integer value from the keyboard. Notice that you do not place a field width number in front of the letter "i".

scanf Function (cont.)

- More than one number can be read from the keyboard at the same time, and the numbers can be different data types.
- Example:

```
scanf(" %i %lf",&year, &taxrate);
```

The above scanf function will allow the user to enter an integer number and a double type floating-point number.

The basic Conversion Specifiers for entering numbers are:

%i	used to enter integer data type
%f	used to enter float data type
%lf	used to enter double data type
%c	used to enter character data