```
>> e=13
e =
    13
```

13 is assigned to e (numerical variable).

The display of the value of a numerical variable is indented.

Several symbolic variables can be created in one command by using the syms command, which has the form:

```
syms variable_name variable_name variable_name
```

The command creates symbolic objects that have the same names as the symbolic variables. For example, the variables *y*, *z*, and *d* can all be created as symbolic variables in one command by typing:

```
>> syms y z d
>> y
y =
y
```

The variables created by the syms command are not displayed automatically. Typing the name of the variable shows that the variable was created.

When the syms command is executed, the variables it creates are not displayed automatically—even if a semicolon is not typed at the end of the command.

11.1.2 Creating Symbolic Expressions

Symbolic expressions are mathematical expressions written in terms of symbolic variables. Once symbolic variables are created, they can be used for creating symbolic expressions. The symbolic expression is a symbolic object (the display is not indented). The form for creating a symbolic expression is:

```
Expression_name = Mathematical expression
```

A few examples are:

```
>> syms a b c x y
>> f=a*x^2+b*x + c
f =
a*x^2 + b*x + c
```

Define a, b, c, x, and y as symbolic variables.

Create the symbolic expression $ax^2 + bx + c$ and assign it to f.

The display of the symbolic expression is not indented.

When a symbolic expression, which includes mathematical operations that can be executed (addition, subtraction, multiplication, and division), is entered, MATLAB executes the operations as the expression is created. For example:

```
>> g=2*a/3+4*a/7-6.5*x+x/3+4*5/3-1.5
```

$\frac{2a}{3} + \frac{4a}{7} - 6.5x + \frac{x}{3} + 4 \cdot \frac{5}{3} - 1.5$ is entered.

```
g =
(26*a)/21 - (37*x)/6 + 31/6
```

$\frac{26a}{21} - \frac{37x}{6} + \frac{31}{6}$ is displayed.

Notice that all the calculations are carried out exactly, with no numerical approximation. In the last example, $\frac{2a}{3}$ and $\frac{4a}{7}$ were added by MATLAB to give $\frac{26a}{21}$, and $-6.5x + \frac{x}{3}$ was added to $\frac{37x}{6}$. The operations with the terms that contain only numbers in the symbolic expression are carried out exactly. In the last example, $4 \cdot \frac{5}{3} + 1.5$ is replaced by $\frac{31}{6}$.

The difference between exact and approximate calculations is demonstrated in the following example, where the same mathematical operations are carried out—once with symbolic variables and once with numerical variables.

```
>> a=sym(3); b=sym(5);
>> e=b/a+sqrt(2)
e =
2^(1/2) + 5/3
>> c=3; d=5;
>> f=d/c+sqrt(2)
f =
    3.0809
```

Define a and b as symbolic 3 and 5, respectively.

Create an expression that includes a and b.

An exact value of e is displayed as a symbolic object (the display is not indented).

Define c and d as numerical 3 and 5, respectively.

Create an expression that includes c and d.

An approximated value of f is displayed as a number (the display is indented).

An expression that is created can include both symbolic objects and numerical variables. However, if an expression includes a symbolic object (or several), all the mathematical operations will be carried out exactly. For example, if c is replaced by a in the last expression, the result is exact, as it was in the first example.

```
>> g=d/a+sqrt(2)
g =
2^(1/2) + 5/3
```

Additional facts about symbolic expressions and symbolic objects:

- Symbolic expressions can include numerical variables that have been obtained from the execution of numerical expressions. When these variables are inserted in symbolic expressions their exact value is used, even if the variable was displayed before with an approximated value. For example:

```
>> h=10/3
h =
    3.3333
```

h is defined to be 10/3 (a numerical variable).

An approximated value of h (numerical variable) is displayed.

```
>> k=sym(5); m=sym(7);    Define k and m as symbolic 5 and 7, respectively.
>> p=k/m+h                h, k, and m are used in an expression.
p =                       The exact value of h is used in the determination of p.
85/21                     An exact value of p (symbolic object) is displayed.
```

- The double(S) command can be used to convert a symbolic expression (object) S that is written in an exact form to numerical form. (The name "double" comes from the fact that the command returns a double-precision floating-point number representing the value of S.) Two examples are shown. In the first, the p from the last example is converted into numerical form. In the second, a symbolic object is created and then converted into numerical form.

```
>> pN=double(p)           p is converted to numerical form (assigned to pN).
pN =
    4.0476
>> y=sym(10)*cos(5*pi/6)  Create a symbolic expression y.
y =                       Exact value of y is displayed.
-5*3^(1/2)
>> yN=double(y)           y is converted to numerical form (assigned to yN).
yN =
   -8.6603
```

- A symbolic object that is created can also be a symbolic expression written in terms of variables that were not first created as symbolic objects. For example, the quadratic expression $ax^2 + bx + c$ can be created as a symbolic object named f by using the sym command:

```
>> f=sym('a*x^2+b*x+c')
f =
a*x^2 + b*x +c
```

It is important to understand that in this case, the variables a, b, c, and x included in the object do not exist individually as independent symbolic objects (the whole expression is one object). This means that it is impossible to perform symbolic math operations associated with the individual variables in the object. For example, it will not be possible to differentiate f with respect to x. This is different from the way in which the quadratic expression was created in the first example in this section, where the individual variables are first created as symbolic objects and then used in the quadratic expression.

- Existing symbolic expressions can be used to create new symbolic expressions. This is done by simply using the name of the existing expression in the new expression. For example: