

## ▼ Section 1: Numerical Calculations

In this section you will use Maple to do some standard numerical calculations. Maple's ability to produce exact answers in addition to numerical approximations gives you more options in solving problems.

### ▼ **Doing Exact Arithmetic with Maple**

Using Maple to do numerical computations is very straightforward. Just enter the numerical expression and end the line with a semicolon. Pressing **[Enter]** will then execute the line and the result will be displayed in blue in the center of the screen.

#### ▼ **Example 1**

A simple calculation has been entered on the next line. Click anywhere in the red line and press **[Enter]**.

```
> 2+4;
```

```
>
```

```
> 12*34567890;
```

```
>
```

Each red input line is "live" and can be modified at any time.

Change the "4" in the line above to an "8" and press **[Enter]**.

Notice how the blue output is automatically updated to display the new result.

```
>
```

#### ▼ **Example 2**

For our next example let's calculate  $134^{39}$ .

```
> 134^39;
```

```
>
```

Unlike your calculator, Maple gives you the exact answer to this problem, all 83 digits worth!

```
>
```

#### ▼ **Example 3**

Maple can calculate with fractions without converting to decimals:

```
> 3/5 + 5/9 + 7/12;
```

```
>
```

#### ▼ **Example 4**

To enter the square root of a number use **sqrt** :

```
> sqrt(24);
```

```
>
```

Notice that Maple has simplified  $\sqrt{24}$  but has left the answer in exact form. In the next section you will learn how to get a decimal approximation for this number.

>

#### ▼ Example 5

Maple has all of the important mathematical constants built in. To enter  $\pi$  type `Pi`.

Notice that an asterisk `*` is required to indicate multiplication.

> `4*(3+Pi);`

>

Again Maple carries out the calculation but leaves the answer in exact form.

>

#### ▼ Example 6

Unlike your calculator, Maple gives you the exact answer when applying trigonometric functions.

> `sin(5*Pi/3);`

>

> `sec(Pi/4);`

>

To get the inverse sine of a number use the **arcsin** function:

> `arcsin(-1);`

>

If you ask Maple to calculate a value that is *undefined* it will respond with an error message:

> `tan(Pi/2);`

>

#### ▼ Example 7

To enter the natural exponential function  $e^x$  in Maple, type: **exp(x)**.

> `exp(x);`

>

And to get the number  $e$  by itself, type: **exp(1)**.

> `exp(1);`

>

#### ▼ Example 8

To enter the absolute value function  $|x|$  in Maple type: **abs(x)**.

> `abs(x);`

>

For example, this function at  $x = -3$  yields

> `abs(-3);`

>  
Note that Maple gives the correct, exact answer for the following since  $e - \pi < 0$ .

> **abs(exp(1)-Pi);**  
>

#### ▼ Example 9

Maple has many special purpose commands for working with numbers. You will learn these as you need them in your math course. Here is one last example for now. If we have an integer and want to factor it into primes we can use Maple's **ifactor** command. Feel free to experiment by changing the number.

> **ifactor(31722722304);**  
>

#### ▼ Example 10

There may be times when you want to enter more than one command on a single line. This is okay to do in Maple, just be sure to end each command with a semicolon. It also helps to put spaces between the commands. When you press **[Enter]** all of the expressions are executed and the results are listed, in order, in a single output field.

> **sin(Pi/3); cos(Pi/3); tan(Pi/3);**  
>

To display the commands in a single execute group but on separate lines, press **[Shift] + [Enter]**. The result will be the same.

> **sin(Pi/3);**  
**cos(Pi/3);**  
**tan(Pi/3);**  
>

#### ▼ Example 11

To calculate and display a sequence of numbers use the **seq** command. Here we calculate the squares of the first 100 natural numbers.

> **seq(k^2, k=1..100);**  
>

### ▼ Numerical Approximations using the *evalf* command

Recall that in the previous section we asked Maple to add three fractions and the result was also displayed as a fraction. This sort of exact arithmetic is very useful but there are times when we prefer an answer in decimal form. The Maple command **evalf** performs this task for us.

#### ▼ Example 1

Compare the results of the next two lines.

> **3/5+5/9+7/12;**  
>  
> **evalf(3/5+5/9+7/12);**  
>

### Example 2

Assigning a name to the result of a calculation makes it easier to use that result in a subsequent calculation. To assign a name we use a colon followed by an equal sign ( i.e. name := result ; ). On the next line we have assigned the letter k to the original output above. Then we apply **evalf** to k.

```
> k := 3/5+5/9+7/12;  
>  
> evalf(k);  
>
```

Important Maple Note: Maple is case sensitive. So for example Maple considers k and K to be different variables..

```
> k;  
>  
> K;  
>
```

By the way, you can also use words as variable names.

```
> joe := 2^5;  
>  
> sqrt(joe);  
>
```

### Example 3

If we want fewer or more digits of accuracy than the default number which is 10 digits we can add an extra argument to the **evalf** command as shown below.

```
> w := 4*(3+Pi);  
>  
> evalf(w);  
>
```

To see four digits of w:

```
> evalf(w,4);  
>
```

To see forty-five digits of w:

```
> evalf(w,45);  
>
```

### Example 4

If you enter numbers with a decimal point Maple automatically gives decimal results. Compare the results of the two lines below.

```
> sqrt(34);  
>  
> sqrt(34.0);  
>
```

Here is another example:

```

> 4-1/3;
>
> 4.0-1/3;
>

```

### ▼ **Example 5**

We can apply the **evalf** command to a sequence of numbers. Below we first generate the exact square roots of the first 10 natural numbers, then apply evalf to get decimal approximations.

```

> result := seq(sqrt(k), k=1..10);
>
> evalf(result);
>

```

### ▼ **Maple Shortcut: Quick reference to the last chronological output**

There will be many times when using Maple that you will string together a sequence of computations. Rather than giving a name to each result as you go along, you can use the percent sign ( % ) to refer to the chronologically last expression computed by Maple. Note, however, that this device can lead to great confusion, and is not generally recommended.

Here are some examples of how it works.

```

> 3/5+5/9+7/12;
> evalf(%);
>
> Pi;
> evalf(%);
> %+5;
>

```

For more on using the ditto symbol see *Exercise 1.4* below.

Warning: Use of % should be avoided as much as possible, as it may make the worksheet unreadable.

```

>

```

### ▼ **Exercise 1.1**

Use Maple to calculate the number  $37^{43}$ .

#### ▼ ***Student Workspace 1.1***

```

>
>
>

```

#### ▼ ***Answer 1.1***

```

> 37^43;
>

```

### ▼ **Exercise 1.2**

Calculate  $\sqrt{34}$  to 18 digits.

▼ **Student Workspace 1.2**

```
[>
[>
[>
```

▼ **Answer 1.2**

```
[> m := sqrt(34);
[>
[> evalf(m,18);
[>
```

▼ **Exercise 1.3**

Find a numerical approximation for the expression :  $\frac{3 + \pi}{7 - \sqrt{13}}$

▼ **Student Workspace 1.3**

```
[>
[>
[>
```

▼ **Answer 1.3**

```
[> answer := (3+Pi)/(7-sqrt(13));
[>
[> evalf(answer);
[>
```

▼ **Exercise 1.4**

The percent sign ( %) is a handy shortcut but it can occasionally lead to some unexpected results.

Here is an example.

First execute each of the next three lines. You should be able to predict the result in advance.

```
[> 4+Pi;
[>
[> evalf(%);
[>
[> %+10;
[>
```

Now go back and re-execute the last line (i.e., **%+10**). Note that the output changes from **17.14159265** to **27.14159265**

Can you explain why?

▼ **Student Workspace 1.4**

```
[>
[>
```

```
[ ]>
```

#### ▼ **Answer 1.4**

The ditto symbol (%) represents the chronologically last number calculated by Maple. So after you executed the first three lines

% = 17.14159265. The second time you executed the line %+10; Maple added 10 to 17.14159265.

To avoid this sort of confusion assign names to each output:

```
[ ]> a := 4+Pi;  
[ ]>  
[ ]> b := evalf(a);  
[ ]>  
[ ]> b+10;  
[ ]>
```

### ▼ **Clearing Variables**

#### ▼ **Assigning new values to existing variables**

Once you have defined a variable, Maple will remember its value during your entire working session. If you want to overwrite the variable with a new value, you can simply make a new assignment.

For example each assignment below redefines the value of the variable  $h$ . (Note: to check the current value for a variable just type it followed by a semicolon.)

```
[ ]> h;
```

```
[ ]>
```

Assign  $h$  the value 56.

```
[ ]> h := 56;
```

```
[ ]>
```

Notice the value of  $h$ , 56, after the assignment.

```
[ ]> h;
```

```
[ ]>
```

To overwrite  $h$  with a new value, make a new assignment using :=.

```
[ ]> h := sqrt(Pi);
```

```
[ ]>
```

You'll notice  $h$  now has the new value you assigned.

```
[ ]> h;
```

```
[ ]>
```

#### ▼ **Clearing one variable at a time: the unassign command**

Sometimes you will want to "clear" a variable in memory so that you can use it in a new situation.

Here is an example. First we assign  $x$  the value 65.

```
[ ]> x := 65;
```

```
>
```

Now assume that we start a new problem and want to enter the general algebraic expression  $x^2 - 4x + 7$  and assign it the name  $w$ . If we just enter this, Maple automatically substitutes the previous value for  $x$ .

```
> w := x^2-4*x+7;
```

```
>
```

In order to get  $x$  to be a general variable again we must first "clear" (i.e. erase from Maple's memory) our earlier value for  $x$ . This is accomplished by using **unassign** command. Note that we use single quotes here.

Execute the next two lines to see how this works.

```
> unassign('x');
```

```
> w := x^2-4*x+7;
```

```
>
```

### ▼ *Clearing all variables at once: the restart command*

The **restart** command will clear Maple's memory of all definitions that you have made. It is like starting a new Maple session. If you are starting a completely new problem you can use the **restart** command to guarantee that there are no leftover definitions from your earlier work. Before you execute the second line below, quickly predict the output.

```
> p := 4;
```

```
> p;  
x;  
h;
```

```
>
```

You probably remembered that  $p$  was 4 and  $x$  had been reassigned to be the variable  $x$ , but you may not have remembered that  $h$  was assigned the value  $\sqrt{\pi}$ . That's why it's a good idea to use **restart** to remove all definitions at once. (As you work through this tutorial you will notice that we start most new sections with a **restart** command.)

```
> restart;
```

```
>
```

Now check the values of variables.

```
> p;  
x;  
h;
```

```
>
```