New users may read this side before starting. Please don't read the other side until later. It may be confusing now.

Goals of showing `Maple` to students

Students should know about `Maple`, and I would like to help them learn about `Maple` as soon as possible. My ambitions are to show students the program, to equip them with the ability to run some commands, and to work with the extensive `Maple` "help" facilities. I want them to feel confident that the system will obey them. I also want them to commit and recover from some of the simple errors that everyone makes when first using such a complex program, and to make these errors in what I hoped would be an environment which would be both emotionally and intellectually supportive.

What follows are four sheets of instructions covering, in turn, arithmetic, algebra, calculus, and graphing. I recommend that students work in pairs at two adjoining computers. Both partners should work through each sheet, helping each other when necessary (and such help likely *will* be necessary, since the instructions are rather minimal!). The students should begin each sheet together. Thus one student may need to wait (and help) if their partner is slower to complete one of the sheets.

Students should at least think of `Maple` as an effective symbolic and graphical and numerical "calculator". It is another resource which can help them do homework and can help them communicate technical information. I do *not* discuss `Maple`'s capabilities for running user programs and only suggest that `Maple` can be a valuable **experimental tool** for investigating mathematics.

Local Rutgers note

The simplest way to begin is at a Rutgers computer lab. Please login, and click on **Start** (lower left location of the screen), then **All programs**, then **Class software**, and, finally, **Maple 12**. At **Startup** please click **Start with Blank Worksheet**. Then begin with the algebra page.

`Maple` is available on `eden` and you may be able to use it remotely. Also there is a student edition which can be bought, but see how you like the program first.

Disclaimer! Non-advertisement!! Important information!!!

Symbolic manipulation programs such as `Maple` are becoming increasingly available. Other popular programs with about the same capabilities are `derive` and `mathematica` and there are many special purpose programs in various fields of science, engineering, and mathematics which have extensive "intelligence" to analyze models. We're considering `Maple` here because Rutgers has a site license for this program, and it should be generally available on Rutgers systems. The specific instructions won't be the same from program to program, but many of the same ideas will be present. Students should expect to have a machine do tiresome or elaborate symbolic computations as well as numerical computations. Please note that many of `Maple`'s capabilities are also available through `matlab`.

PLEASE: NEW `Maple` USERS SHOULD WORK THROUGH THE OTHER PAGES FIRST. THEN READ WHAT'S BELOW.

The `Maple 12` GUI compared to the command line version and older editions of `Maple`

xmaple gives a graphical user interface (GUI) to the `Maple` program. `Maple` can also be used by typing *maple* at any command line. Then the commands and responses are all in ASCII characters. The plotting commands have limited responses, of course. Everyone at some time may need to use the command line version. Good displays may not be available, or there may be limited bandwidth between the user and the CPU. Images usually take many more bits than characters.

The major difference is that commands in the GUI need not end with a semicolon to be executed. This is very important, and can be rather frustrating to the naive user of the command line interface! Just typing *3+2* gets no response. Commands *must* end with a semicolon to be executed.

Another significant difference affects `Maple` names indirectly. A `Maple` help page states:

> A **name** in its simplest form is a letter followed by zero or more letters, digits, and underscore characters, with lowercase and uppercase letters distinct.

Example The input is *3x* **RET** (here **RET** means "Hit the enter key.").

`Maple` 12's GUI response

3x is displayed, and the system interprets this as 3 multiplied by x.

Command line version's response

Both *3x* and *3x*; get

syntax error, missing operator or ` ;` :

and the program does nothing further.

The GUI allows some implied use of $*$. The GUI response to *int(x^2→sin(x),x)* (here $\rightarrow$ means the right arrow key) is the correct antiderivative. But *int(x^2→3sin(x),x)* gets `Error, missing operation`. Various inputs requesting differentiation of x^2 multiplied by $3\sin(x)$ sometimes get incorrect or misleading responses depending upon placement of parentheses and whether $*$ is absent or not. To a casual user, this is at least annoying. I recommend that students type $*$ whenever they want to indicate multiplication.

Output differences

I've used *>2^(2^(2^2))*; to show why a displayed result may not be useful (that is, using : rather than ;). The command line interface (on a fast enough computer!) will show all 19,729 decimal digits of the result. The GUI response is interesting. It gives the initial 100 digits, then the character string ...`19529 digits`... and then the last 100 digits. Since $19,729 = 100 + 19,529 + 100$, things are o.k., but the difference was startling*.

Another output difference occurs if we ask for *int(diff(arctan(x^3),x$3),x);* which takes a bit of work. The same answer is shown but in the command line interface, `Maple` abbreviates certain common algebraic pieces of the answer, using %1 and %2. The GUI answer does not bother. Just as in the previous case, this probably can be adjusted by tweaking some preset display parameters, but the defaults are different.

---

* 10^{9999} displays entirely, but 10^{10^4} does not.