You should **submit** an single electronic package that contains the **source** files for your work on the programming questions, by following the submission procedure described in class and on the class discussion forum. *Using the **discussion forum** to clarify details of both the main program and the submission format and procedures is an important part of this homework.*

You are welcome to use any inanimate **resources** (e.g., books, Web sites, publicly available code) to help you with your work. However, *all such help must be clearly noted* in your submissions. Further, no matter what you use, *you must be able to explain, in detail, how it works.* (You may be called upon to explain your homework individually.) Refer to the class policy for details, and ask for clarifications if you are unsure if something is allowed.

Your implementation must use **clean, portable Python 3** that minimizes dependencies on OS, version (beyond 3.x), etc. (If in doubt, please ask.) Packaging and documentation of code are worth a very significant portion of the grade. This homework centers on **extending the `calc.py`** example (which is included with PLY and has been discussed extensively in class) as follows.

1. It should support real numbers, in addition to integers, represented in the conventional manner (e.g., 3.1415, 602.3).
2. It should support *div* and *mod* operators with the usual semantics with operator tokens `//` and `%`, respectively.
3. Operations on integers should produce integers as outputs unless the result cannot be represented as an integer. Thus, `2 + 3` should produce `5` and not `5.0`, while `3/2` should produce `1.5`.

Input-output: The program should read its input from the *standard input* stream and write its output to the *standard output* stream. Optional diagnostics may be written to the *standard error* stream. It is very important that the program read its input only from the standard in put stream and that it write nothing except the specified output to the standard output stream. In particular, there should be no prompts or informational messages printed to standard output.

The **input** consists of the calculator language of calc.py as discussed in class, extended to support the above features. The **output** consists of (only) the values of stand-alone expressions (excluding expressions that are part of an assignment statement) in the input. The output for a statement must be produced as soon as the statement appears in the input stream (before waiting for or reading any further input that may appear). There is exactly one (i.e. a unique) output for any given input (but there may be multiple inputs resulting in the same output). If two outputs for the same input differ by even a single character/byte then at least one of them is incorrect. Sample inputs and outputs should be discussed on the class discussion forum.

Submission: Refer to the class discussion forum for further submission instructions. Use the *gzipped tar* (strongly preferred) or *zip* formats to package your submission. Name the electronic submission using the template

```
cos301-hw01-lastname-firstname-pqrs.tgz
```

where *lastname* and *firstname* are replaced by the obvious and *pqrs* is replaced by a 4-digit string of your choosing. (Replace `.tgz` with `.zip` if you use zip instead of tar for packaging.) The submission should be designed so that the command

```
tar zxf cos301-hw01-lastname-firstname-pqrs.tgz
```

results in the creation of a directory `cos301-hw01-`*`lastname-firstname-pqrs`*. In that directory should be all the source code (organized in further sub-directories as needed) as well as a README file with the usual semantics. *Do not submit any kind of non-source files* (results of compilation, etc.).