

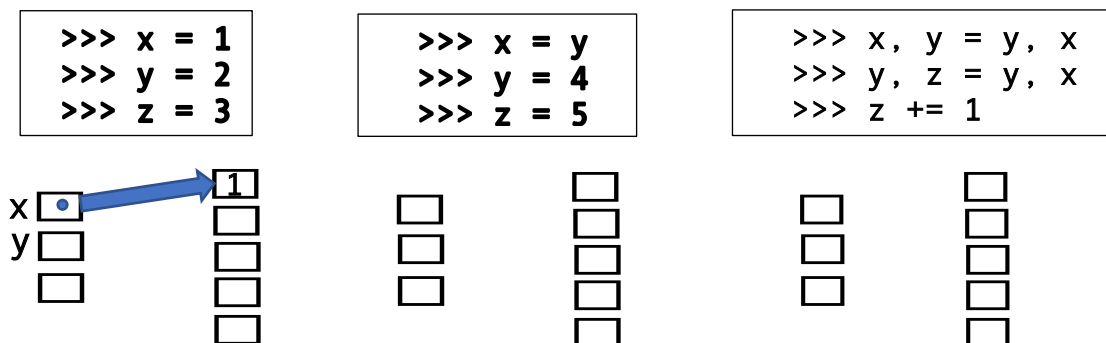
CS5001-5003, Homework 2, Spring 2022

There are 3 questions. Submit a separate file for each part (or a zipped folder with 3 files). This homework should be submitted directly on Canvas.

1. [30 points] Give 4 examples of "bugs" in programming. For each bug, indicate if it is unique to Python (such as violating Python-specific syntax rules) versus more general (and so apt to occur in other programming languages). Provide Python code illustrating each bug. If the error has an official name in Python, please provide that, as well. Here is one to get you started: `ZeroDivisionError`. Finish its description and add three more. Submit one file called **bugs.py** with code to cause the bugs and comments to describe them.

Hint: Review Halterman, Chapter 2 or Downey Chapters 1 and 2.

2. [30 points] Using the boxes below -- or your own equivalent diagrams -- use **arrows** to show the effects of the following sequences of steps. Show **variables** on the left and their **values** in corresponding "memory locations" on the right. It is conventional to use something like "●" to represent a "reference" to a memory location. Use the left-most two columns of boxes to show the results of the first three steps, the middle two the second three steps, and the right-most two the final three steps. The first step is already done.



Submit a picture in any common file format, with the appropriate boxes filled in.

Hint: Review Halterman, Section 2.2

3. [40 points] Microsoft provides this "algorithm" for determining if it is a Leap Year:

1. If the year is evenly divisible by 4, go to step 2. Otherwise, go to step 5;
2. If the year is evenly divisible by 100, go to step 3. Otherwise, go to step 4;
3. If the year is evenly divisible by 400, go to step 4. Otherwise, go to step 5;
4. The year is a leap year (it has 366 days);
5. The year is not a leap year (it has 365 days).

Using Python logical operators, conditionals, and arithmetic operators such as %, convert this to Python code. You cannot directly use "go to" steps, since that construct does not exist in Python, for good reasons.¹ Your file should define a Boolean function called **is_leap_year**, that accepts one input (a year integer). It should return True if the input is a leap year and False if it is not. Do a little manual testing first:

```
>>> is_leap_year(1999)
False
>>> is_leap_year(2000)
True
```

Then, your `main()` function should **test** a half dozen examples and print results. About half should be leap years and half not. Here is some sample input/output from the `main()` function:

```
Please enter a year greater than 0: 2000
The year 2000 is a leap year.
Please enter a year greater than 0: 2001
The year 2001 is not a leap year.
```

You do not need a loop here; just program a sequence of a half-dozen examples by hand. To help in verifying your test data, the following are the leap years in the first half of the current century: 2000, 2004, 2008, 2012, 2016, 2020, 2024, 2028, 2032, 2036, 2040, 2044, and 2048. Submit your solution, including the `main()` testing code, in a file named **leap.py**.

Standard Instructions for CS5001-5003 Assignments (Boilerplate):

1. Except for assignments or parts of assignments where Gradescope submission has been specified, please submit your solution, including any supporting files, using Canvas. Navigate to the *Assignments* page, click on *Start Assignment*, and then click on *File Upload*. (For some problems, *Text Entry* may suffice.) Please be careful to name your files exactly as specified in the assignment. You may submit multiple times prior to the deadline. However, be careful not to overwrite any needed files from previous, partial submissions. One good strategy is to do all the work for the entire assignment in a single folder, compress it to a ".zip" file, and upload that. Then you can confidently replace it with a newer version, if needed.

2. Standard *Academic Integrity* guidance: For Homework Assignments – unlike Lab Assignments -- please do not collaborate with other students, beyond posting generalized questions and answers on Piazza. When in doubt, you can post your question "Only to Instructors" – but please indicate if it is OK for us to share your question and our answer with the whole group. Also, if you rely on sources on the web or resources other than those assigned, please be sure to cite those sources. (In Labs, Pair Programming is often required or encouraged.) Except when Pair Programming is expected on a Lab, you will usually learn more if you solve each problem on your own. Python's built-in documentation features can be very helpful.

¹ In March, 1968, Edsger Dijkstra, a pioneer of the "Structured Programming" movement, published an article in *Communications of the Association for Computing Machinery* [CACM]. He argued for abolishing the "go to" construct common in early languages. The article was entitled, *Go To Statement Considered Harmful*. Reading this short but important article at some point is part of your journey. <https://homepages.cwi.nl/~storm/teaching/reader/Dijkstra68.pdf>.