VMI-Specific Chapter Instructions

Chapters 4 and 5: Lab Programming Problems

These problems will be graded by a computer program, and so must be in a certain format to earn points. A large part of your overall grade is based on whether or not your homework submission follows this format – don’t miss these easy points!

Unlike previous lab assignments, you will email these to me as a collection of functions saved in a single .zip file, and named with the names of the people in your lab group separated by commas, for example “Ron Howard, Henry Winkler, Erin Moran.zip”. This alone is 10% of your overall grade. seven files (or eight if you attempt the bonus) named “problem1.m”, “problem2.m” etc. The names must be in lowercase. This is 10% of your overall grade. If the zip files contain anything else (a zip file within a zip file, extra .m files, favorite chocolate cake recipes) then it will fail the automated grading and you will lose a whole letter grade, since if any are incorrect it will require debugging of your entire file to fix and award partial credit.

Each of the functions must have the proper function declaration statement that is provided with the lab problem, and should not generate an error using the test input supplied for each problem. This is worth 10% of the problem grade. **Even if you cannot solve the problem, if your programs do not generate a Matlab error when run you will still earn this 10% for every problem**, but if just one of your functions generates an a Matlab error you will lose the 10% on every problem. This is easy to ensure; if the function declaration given is, for instance,
result = problem10(x1,x2)
Then simply assigning the output value to something, like
result = 0;
will prevent the program from generating an error and earn 10% of the grade. It is possible to earn (or lose) about one-third of your grade by submitting a properly-formatted set of problems.

Note that displaying a number to the screen is not the same as having a function return a number. For instance, say a requirement was to write a function that took no arguments and always returned the number “2”. The below function is correct:

```
function out = goodfunction()
out = 2;
```

The function below is not correct, although it does write “2” to the screen, because it doesn’t return the number.

```
function badfunction()
fprintf('2');
```

The reason functions return numbers is so their results can be saved in a variable.
x = sin(2); saves the sin(2) in variable x. If the sin() function didn’t return the variable but just wrote it to the screen, it would not be of much use. Calling the above functions from the command line yields the following results:

```
x = goodfunction();% This returns 2 saved in variable x.
x = badfunction(); Errors because it doesn’t return a number.
```

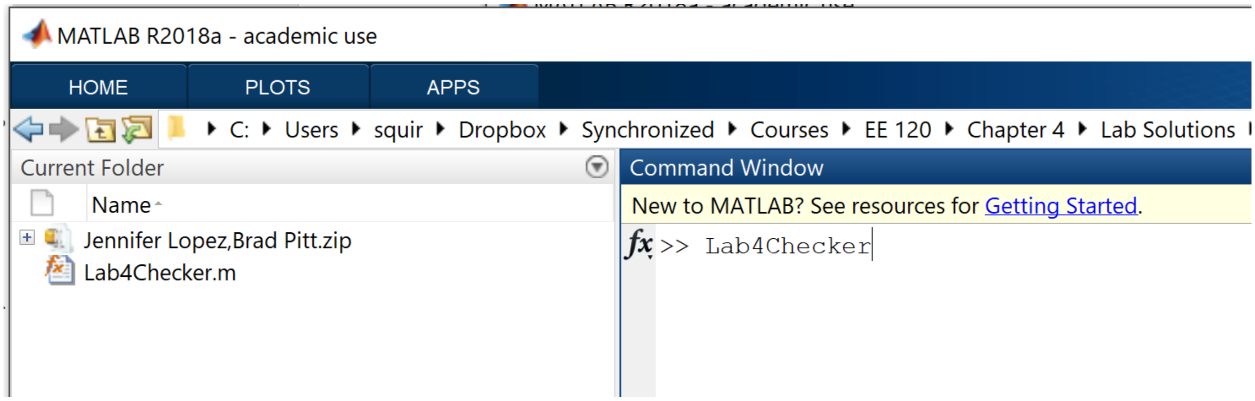
The last problem in Chapters 4 and 5 (problems 17 and 33, respectively) are bonus problems, and if solved correctly give +15 points, making it possible to earn over 100 points. No partial credit is awarded for this problem, and it is possible to get a perfect 100 score without attempting it. This is a tough bonus problem designed to stretch the abilities of students who find the earlier problems straightforward.

Note: The last two problems in Chapter 5 (problems 32 and the bonus problem 33) are exceptions; these graphic results must go into a Word file in .docx format, not .m, so your Chapter 5 .zip bundle will have six .m files and up to two .docx files (if you attempt the optional challenge problem #33).

DO NOT SEND IN YOUR .ZIP FILE WITHOUT VERIFYING
1) it is a .zip file (not a .rar, .7, etc.)
2) the .zip file just holds .m files (or .m files and two .docx files for Chapter 5)
3) the .zip file is not corrupted – can you open it?
4) the .zip file does not contain another nested .zip file
5) the .zip file is named with the names of the people in your lab group
6) the .m files take and return numbers, and do not return plots (except for Chapter 5, problems 32 and 33, whose output goes into a Word file.)

If you ignore the above steps, I will still grade your work (well, I won't be able to if the problem is #3) but it will needlessly cost you a letter grade. Get the entire grade you deserve and complete those checks. If you give me a corrupted file that you did not check and that I cannot open, then I will alert you, and your submission will be graded as a late grade.

Optional step for students wishing to check their zip files before submission
You can run your .zip file through a format-checker written in (what else?) Matlab to make sure all your files are in the correct format. It is called "Lab4Checker". It checks to make sure files are correctly zipped and that they take and return the correct number of arguments; it does not check whether the answers they provide are correct. Run it by downloading it from my website, put it in an empty directory with your .zip file, and run it from the command window as shown in the screengrab below. It will read the format of your .zip file, make sure all the problems are in there and named correctly, and alert you if any of your files do not take or return the correct number of arguments. It is designed to be crash-proof, even if you coded your files incorrectly. If you manage to crash it, send me your .zip file that managed to crash it; I'll personally check the format of your files for you and fix the file checking program as well.


MATLAB R2018a - academic use
HOME
PLOTS
APPS
C: ▸ Users ▸ squir ▸ Dropbox ▸ Synchronized ▸ Courses ▸ EE 120 ▸ Chapter 4 ▸ Lab Solutions
Current Folder
Name
Jennifer Lopez,Brad Pitt.zip
Lab4Checker.m
Command Window

New to MATLAB? See resources for Getting Started.

```
fx >> Lab4Checker
```