

Homework Guidelines

Physics 129 Summer 2024

Copyright © 2016–2024 Everett A. Lipman. All rights reserved.

The following uses are prohibited without prior written permission:

- Duplication in any form
- Creation of derivative works
- Electronic posting and distribution

For each weekly assignment, you will create a single archive file to be submitted according to the instructions below. Here is how that should be done:

1. Create a directory for your homework. The directory name should be your last name, an underscore, your 7-digit perm number, a second underscore, and the word “homework”, all in lower case. For instance, if your name is Enrico Fermi and your perm number is 1602177,

```
cd
mkdir fermi_1602177_homework
```

2. For each problem set, you will create a subdirectory:

```
cd ~/fermi_1602177_homework
mkdir fermi_1602177_hw1
mkdir fermi_1602177_hw2
```

be sure the subdirectory names also include your last name and perm number.

3. For each numbered problem in the set, you will create a corresponding text file, which should contain answers to everything you are asked in that particular problem. The name of the file should be the problem number preceded by the letter p, followed by an underscore and the name of the set. If you are asked for a program or other type of answer, create the appropriate file (for example, p2_hw6.py for a Python program).

```
cd ~/fermi_1602177_homework/fermi_1602177_hw1
vi p1_hw1.txt (do some text editing)
```

4. When you are ready to turn in your homework, create a .tar archive file, the name of which includes your last name and perm number:

```
cd ~/fermi_1602177_homework
tar -zcf fermi_1602177_hw1.tar.gz fermi_1602177_hw1
```

5. Check to make sure the tar file has all of your work:

```
cd ~/fermi_1602177_homework
tar -ztf fermi_1602177_hw1.tar.gz
```

6. Compute a checksum (MD5 hash) for the tar file:

```
cd ~/fermi_1602177_homework
md5sum fermi_1602177_hw1.tar.gz
```

7. The output will look something like this:

```
f4468281ff4beeb20de8c58dce0b6bee  fermi_1602177_hw1.tar.gz
```

8. Log in to Gradescope:

- (a) Go to `www.gradescope.com`.
- (b) Click on **Log In** at the upper right.
- (c) Click on **School Credentials** at the lower left of the popup window.
- (d) Use <Ctrl>-F or <Command>-F (Mac) to search for “barb” on the page.
- (e) Click on **University of California Santa Barbara NetID**.
- (f) Log in and select the appropriate course. If you are registered for the class, you have been added to the Gradescope roster using your `@ucsb.edu` email address. **If you do not see our course**, you probably already have an account associated with an address containing `umail`. In that case, **you must merge your accounts according to the instructions** [here](#).

9. Select the correct assignment. Create a text file containing only your MD5 hash and upload that along with the archive file.

10. Back up your homework submissions:

```
cd
mkdir oldhw
mv fermi_1602177_homework/fermi_1602177_hw1.tar.gz oldhw
```

The formatting requirements above are important, because they enable the TAs to automate the time-consuming task of unpacking and grading the assignments. If you do not follow the instructions, the TA may deduct credit from your assignment.

You must turn in a plain-text answer file for each problem, containing at a minimum an explanation of how you got your answer and the output of any program you are asked to write. This will be graded.

Unless otherwise specified, all programs must be written in Python, version 3, and all plots and images must be turned in as Encapsulated PostScript (EPS) files.

You must be the sole author of any code you turn in for this class. The only exception is that you may use code from the example programs in the course Python directory on your Raspberry Pi. **If you turn in code that can be found on the Internet, is generated by AI, or that in the instructor's judgment is essentially the same as that turned in by another student, you will receive a score of zero for the entire homework assignment or project.**

If a program's output is not text, then typically you will be asked to turn in a separate file containing the output. In that case, your text file should include a description of how your program was designed, and how it works. The text file is also a good place to describe aspects of your code that may not be obvious, for instance how you optimized it, and how much better the performance then was. You can also describe problems you are aware of but were not able to fix, or ideas you were unable to implement.

If your output is in a separate file and your program for the problem contains detailed comments, you can note this in the text file and include only a brief overview. You must turn in both your program code and an answer file for any problem that asks you to write a program.

All code you turn in must run from the command line on a Raspberry Pi 5 set up for Physics 129. You may use only 7-bit ASCII text (unaltered English characters) in the code you turn in, in your program interfaces, and in your program output.

Better answers and code will get better grades.

It will often happen in the homework for this class that you will run across shell commands or Python statements, functions, and methods that have not been mentioned in lecture, nor in the reading assignments. You will then need to look these up and figure out how they work. Use the `man` command, your textbooks, and web searches. For commands such as `alias`, which are built into `bash`, you can use the `help` command.

In the case of Python, the official documentation is very helpful. If you encounter a new statement, try here:

<https://docs.python.org/3.11/genindex-all.html>

For more general questions, here:

<https://docs.python.org/3.11/> .