PHYS 162 Fall 2019 Final Exam

Rules, Guidelines, and Suggestions:

- **Do not begin until you are instructed to do so.**
- **Please indicate somewhere in your source code what operating system you're using (Windows, Mac, etc.), what text editor you use (Xcode, notepad++, Emacs, etc.), and by what means you compile your code (from within some fancy software, on the Windows command line, from the Mac Terminal, etc.). Providing this information helps me to figure out if some error is system-dependent.**
- No communicating with anyone except for me.
- The only internet resource you can use is Google Drive.
- For your `C++` source files, make sure they compile!
- Compress all files to be submitted into a single zip file before submitting on Canvas.
- NO CELL PHONES. They cannot be out or visible. You CANNOT use them to keep time and you CANNOT use them as a calculator.
- The test is out of 100 points. Point values for each problem are indicated.
- Be sure to attempt every problem. If you get stuck on a problem, skip it and move on to the next one. Work all the problems you consider easy first. An easy problem and a difficult problem of the same length have the same point value.
- Your code will be checked for good coding practices.
- Partial credit will be given assuming you SHOW YOUR WORK and USE COMMENTS to tell me what you're thinking.
- Don't hesitate to raise your hand or come to the front of the room to ask me a question if something is unclear. I can't guarantee I'll be able to give you an answer, but it never hurts to ask.
- You will have until the end of the class period.
- You may keep this copy of the exam when you are done.

Throughout this exam things for you to do will appear in **bold**. *To ensure maximal credit, do not neglect to complete, or at least attempt, any of these tasks. Also, pay careful attention to which programming language you are asked to use as many tasks can easily be completed in either language.*

1. (35 points) The position of a particle is observed to be

$$x(t) = 5 + 1.2t - \frac{2}{3}\sqrt{13}\, t^{3/2} - 4(16 + 8t + 2t^2)e^{-0.5t},$$

where $0 \le t \le 10$ and t is measured in seconds. The units of other quantities are suppressed. but you can assume base SI units throughout.

- With the power of calculus, we can use the position to find the velocity (you don't need to know how to do this):

$$v(t) = 1.2 - \sqrt{13t} + 4t^2 e^{-0.5t}.$$

Your first task if to **use** the numerical derivative that we learned in class to find $v(t)$ numerically from $x(t)$ over the given domain. **Write** a C++ program to do this and **output** the result to a file with the first column being values of t and the second column being $v(t)$. **Use** a helper function for $x(t)$.

- **Read** the data from the previous step into Python and **use** matplotlib to plot both the exact $v(t)$ given above and the numerical data you read in from the file on the same axes. **Use** different colors and **make** one curve a dotted line so that it is easy to distinguish the two plots. Be sure to **include** a title, axis labels, and a legend.

- Use Python to numerically **solve** for all times on the given domain where the particle comes to rest using the exact form of $v(t)$. You should be able to use your plot from the previous step to help you find the answers.
  - **Print** the solutions to the screen.
  - **Mark** the solutions on the plot with solid black circles.
  - **Annotate** the solutions on the plot.

2. (15 points) This problem has two parts:

- In Python, **ask** the user for a mean and standard deviation (be sure to **convert** these to floating point numbers). **Use** Python's random number generator to pick a random number of data points between 1 and 10,000 (inclusive of both endpoints; which random function should we use in this case?). **Use** Python's random number generator to pull that number of data points from the normal distribution given the mean and standard deviation read in from the user (which random function should we use in this case?). **Print** all the data points to a file, one data point per line.

- In C++, **read** in the data file you just created and **count** the number of data points it contains. **Print** this value to the screen in the form of a complete sentence.

3. (30 points) Two hundred years from now, scientists are conducting an experiment with gas particles near the surface of the sun. Their very sophisticated instruments are able to measure the kinetic energies (in eV) of the gas particles individually. They collect a large set of data, which they store in the file `EnergyData.dat`. Through the magic of time travel, this file is available for you to download on Canvas.

   (a) **Build** a histogram of the data. **Add** solid markers at the midpoints of the tops of each rectangle. Don't forget to **include** a title and axis labels.

   (b) The scientists believe that the gas particles behave like an ideal gas and it is well known in this case that the probability of finding a gas particle of a given energy E is given by the Boltzmann distribution:

$$P(E) = \frac{1}{Z}e^{-E/(kT)},$$

   where $k = 8.617 \times 10^{-5}$ eV/K is the Boltzmann constant, Z is just some normalization constant, and T is the temperature of the gas. Using `Python`'s curve fitting capabilities, **fit** the points marked by solid markers using Z and T as fit parameters; you can fix k to the known value. **Add** the fitted curve to your graph (make sure it's smooth!). I don't care about Z, but tell me (**print out**) your best estimate of what the temperature of the gas is. (*Hints: you will likely have to provide initial guesses for both Z and T. Use your graph to get a reasonable guess for Z (it's super small if you don't use* `density=True`*, but that's fine). As for T, we're near the surface of the sun so it's quite large, thousands of kelvin.*)

   (c) **Print out** the mean of the energy data.

4. (20 points) An instructor of yours has designed a curve system for assigning letter grades to his students for their Final Exam, but he's too lazy to figure out how each student's score translates to a letter grade so he has asked you to **write** a `C++` program for him to do the job. The way it will work is that the program will **prompt** the user for a numerical score, which can always be assumed to be greater than or equal to 0 and the program will then **print out** a sentence like

```
The score of 76% should be assigned a letter grade of C.
```

   Here is the weird curving scheme:

score range	letter grade
84 - 100	A
69 - 83	B
53 - 68	C
40 - 52	D
0 - 39	F

   One last task: there was some extra credit on the exam and some over-achieving students managed to get over 100%, but the instructor wants all scores to be capped at 100%. Therefore, before figuring out the letter grade, your program should **pass** the score that is read in from the user, whatever its value, to a helper function that will either **leave** the score unchanged if it is 100% or less or **modify** it so that it will be 100% if it is over 100%. *Note: this helper function modifies the score, but returns nothing. Think carefully about how to accomplish this.*