Syllabus
Physics 248 (formerly 298OWL): An introduction to computing in physics
Fall 2019
Professor: Bryan K. Clark
Course Website: https://courses.physics.illinois.edu/phys298owl/fa2019/

Introductory Information

The basics

Physics 246 (298owl) will teach you to be a fearless code warrior, exploring the behaviors of systems that are too complicated for analytic characterization.

Prerequisites: Physics 211, Math 231. Corequisites: Physics 225, Physics 212. No prior programming experience is required.

Two credit hours.

Required stuff

There are no required texts. You must come to each class (including the first) with a laptop. Please be sure to bring your power adapter too.

But really, now

You will confirm that General Relativity is responsible for the non-Newtonian behavior of Mercury's orbit, and calculate π using simulated grains of sand. You will simulate fluid dynamics, model predator-prey reactions and learn about quantum computing.

Dates

Homework assignments are due **in class** at the start of each class. Be sure you have answered the questions about time spent, online resources, and so forth.
There will be an in-class 30 minute quiz on Thursday, September 19 and a final exam on the day of the final.

OWL?

For obscure reasons courses with numbers like 298 are classified as "independent study" even when they're not. The university requires us to append a three-letter code to the course number. Since the owl is the symbol of Athena—goddess of wisdom, inspiration, mathematics, strength, and other good things—"OWL" seemed like a sensible choice. We have filed for permanent course status; once the process is finished the course will become Physics 246. And why 246? Because that's the vacuum expectation value of the Higgs field.

About this course

Why this course?

As the needs of our students evolve—there is, for example, increasing focus on early readiness for research—the Physics faculty are obliged to adjust both what we teach, and how we teach.

There is a rich tradition of innovation in engineering pedagogy at Illinois. Fifty years ago UIUC became the first school to teach its undergraduates to design computers. More recently, our colleagues have become national leaders in successful efforts to improve instructional outcomes in elementary physics. We intend to continue this Illinois tradition by incorporating computational literacy into the set of core competencies to be mastered by our students.

Just as we require physics majors to enroll in courses taught by Mathematics, but teach the applications of mathematics to physics in our own courses, we hope to do the same with programming. We will continue to require that our students take an introductory course in Computer Science, while incorporating into our own courses machine-based approaches to problems that cannot be solved analytically. Examples include chaos and nonlinear phenomena; fluid dynamics; real-world electrodynamics; quantum mechanics of multi-electron atoms.

This course is a first step. From it, we expect that students will come away with a better grasp of complex phenomena and will be prepared to engage with research experiences that would otherwise have been inaccessible. This will bring to the department's scientific efforts the collateral benefit of an enlarged pool of competent research assistants. If we are successful, our methods should generalize to other disciplines in science and engineering.

Background

The technical foundation for physics majors includes material in physics, mathematics, computer science, and chemistry. But though the courses taught outside the Physics Department provide an excellent introduction to important subjects, they are insufficiently dense in application to specific physics topics to stand on their own. We find this to be especially true in mathematics and computer science. Consequently, the Physics Department offers undergraduate and graduate courses on mathematical methods for physics, as well as a graduate course in computation.

Recently we have now added two new undergraduate courses in computational physics: this course and 498CMP. By simulating physical systems and observing their (simulated) behaviors, students can more efficiently grasp concepts that might be otherwise obscured by mathematical equations. By developing their computational skills, students are better prepared to assist in data acquisition and analysis tasks in a research setting. In addition, about half of our graduating majors choose employment over graduate study; they often report that prospective employers are seeking to hire employees with computational skills.

This course

This is a two-hour standalone course that does not assume any pre(co)requisites other than Physics 211, Physics 212, Math 231, and Physics 225. We will not assume prior enrollment in a Computer Science course and will emphasize computational physics. To allow for portability, we will minimize the use of locally built development tools. This course is a first step in a planned longer-term effort to integrate increasingly sophisticated computational material into the advanced physics courses.

Students will work directly in Jupyter notebooks (a common approach both in industry and academia these days).

Most classroom time is spent solving the computational problems.

There will be a machine-based problem set for each unit, one in-class quiz, and a final exam.

Units

There will be between 10-14 units in this course. This will be adjusted as the course goes on.

Unit 1: N Ways To Generate PI

- Setting up your owl account
- 0. PDF on Hardware (nothing to hand in)
- 1a. Python Preliminaries (nothing to hand in)
- 1b. N Ways to Compute PI

Learning objectives: Familiarity with Jupyter notebooks; Being able to program loops and list comprehensions; Plotting data; Understanding and using random numbers; Developing an intuition for stochastic errors; Limits and Series;

Unit 2: The Dynamics of Throwing Balls into the Air

- 2. Dynamics of Throwing Balls

Learning objectives: Remembering dynamics; Differential Equations as a route to understand scientific phenomena; Air Resistance; Programming Functions; Understanding abstraction of computational components; Validating code; Euler Integration and errors associated with them; Plotting on logarithmic scales;

Unit 3: Improved Dynamics and Springs

- 3a.Improved Dynamics
- 3b.Springs and Pendulum

Learning objectives: Programming and using classes; Midpoint formulas for differential equations and their associated errors; Learning how to modify small parts of a code to change the physics/more abstraction; Damped harmonic oscillators; Pendulums; Forcing functions;

Unit 4: Orbital Dynamics

- **30 minute Quiz**
- 4. Orbital Dynamics

Learning objectives: Developing two-dimensional dynamics code; gravitational forces; central potentials; Kepler's laws; analyzing data; orbital dynamics;

Unit 5: Mercury Perihelion

- 5. Mercury Perihelion

Learning objectives: General relativity; machine precision;

Unit 6: Exoplanets

- 6. Exoplanets
- Exoplanet data file

Learning objectives: Fourier transforms; analyzing real data; basic data I/O techniques & signal processing;

Unit 7: Quantum Computing

- 7. Quantum Computing

Learning objectives: quantum computing

Unit 8: Predator-Prey

- 8. Predator-Prey

Learning objectives: biophysics; differential equations for other systems; ecosystems; continuous time markov chains; stochastic methods

---

Unit 9: Fluid Dynamics

- 9. Fluid Dynamics

Learning objectives: Fluid dynamics; animation; turbulence; convention

---

Unit 10: Random Walks

- 10. Random Walks

Learning objectives: Random walks; stochasticity;

Unit 11: Markov Chains

- 11. Markov Chains

Learning objectives: Markov chains; Metropolis Algorithm; Boltzmann distribution; statistical physics; limiting cases;

Unit 12: Particle Physics

- 12. Particle Physics

Learning objectives: Particle Physics; simulation of a particle physics experiment

---

Extra Credit 1:

- N Ways to Integrate

Learning objectives: markov chains; integration; symbolic computation

---

Extra Credit 2:

- Relativistic Spaceflight

Learning objectives: General Relativity;

Extra Credit 3:

- GR Embedding Diagrams

Learning objectives: General Relativity; Embedding Diagrams

Course staff and office hours

Instructor: Bryan Clark, Assistant Professor of Physics.
• Office: ESB 2111
• Office hours: Wednesday 2:00 - 3:00.

Teaching assistant/grader: Ryan Levy.
• Office: ESB 3101
• Office hours: Tuesday 10:00-11:00 am.

Calendar

Unit, class date	HW due date	50% credit due date	Comments
1, 8/29	Thursday,September 5	Thursday, September 19	
2, 9/5	Thursday, September 12	Thursday, September 26	
3, 9/12	Thursday, September 19	Thursday, October 3	
4, 9/19	Thursday, September 26	Thursday, October 10	quiz 9/19
5, 9/26	Thursday, October 3	Thursday, October 17	
6, 10/3	Thursday, October 10	Thursday, October 24	
7, 10/10	Thursday, October 17	Thursday, October 31	
8, 10/17	Thursday, October 24	Thursday, November 7	
9, 10/24	Thursday, October 31	Thursday, November 14	
10, 10/31	Thursday, November 7	Thursday, November 21	
11, 11/7	Thursday, November 14	Thursday, December 5	
12, 11/14	Thursday, November 21	Wednesday, December 11	
13, 11/21	Thursday, December 5	Wednesday, December 11	
Thanksgiving break			
14, 12/05	Wednesday, December 11	Wednesday, December 11	
final exam			

Course policies

Attendance

You are required to attend each and every one of the course meetings, arriving on time with your laptop computer and charger. Excused absences will be granted and documented in accordance with University policy as described in Article 1, Part 5 Class Attendance, of the Student Code.
You ***must*** file your documentation concerning an excused absence on the Physics Department's Excused absences portal within ***two weeks*** of your absence.
Excused absences fall into the following categories as defined by the code:

- illness
- emergency beyond the student's control (e.g. an auto accident or death in the family)
- required attendance at a University event (e.g. varsity athletics)
- serving as a volunteer emergency worker
- religious observance or practice: this requires you to file a "request for accommodation for religious observances form." The form must be uploaded to the Excused Absences portal ***no later than two weeks after the first day of class.***

Each missed class results in a 5% reduction of your final grade. If you have an unavoidable interview for an internship, discuss it with me in advance as well as filing your documentation here.

Late homework

Homework assignments are due at the ***start*** of the weekly class meeting. Assignments that are up to one week late will receive at most 50% of full credit. (Toward the end of the semester, all assignments must be in by the last day of the semester - see calendar for details!) We will not grade assignments that are more than one week late.

Grading

- Homework: 70%
- Midterm Quiz: 10%
- Final: 20%
- Missing class: 5% off final grade

Your final numerical score is computed as

$$100 \times \left(0.7\frac{\text{Homework Points} + \text{Extra Credit Points}}{\text{Total Homework Points}} + 0.2(\text{Final}) + 0.1(\text{Midterm})\right) - 5 \times \text{Unexecused Absences}$$

The final breakdown of how your grade depends on your numerical score goes as:

- 100+: A+
- 90-100: A
- 80-90: A-
- 70-80: B+
- 60-70: B
- 50-60: B-
- 40-50: C+
- 30-40: C-

- 20-30: D
- 10-20: D-
- 0-10: F

All problem sets count for the same amount. There may be some problem sets which may take two weeks (check the schedule when they are assigned). Even so, they count the same as one homework in the final grade calculation. Unless otherwise noted, every exercise in a problem set counts an equal fraction of the assignment and every part (a,b,c,...) of an exercise counts as an equal fraction of the exercise. 5 points of the problem set will be for mandatory questions (e.g. time spent on assignment, references, collaborators).

Sometimes there are typos in the assignment (although we are working hard to remove them). Please ask when confused! Don't spin your wheels a long time on something that might be a typo. These aren't trick questions - we are trying to ask reasonable things.

Calculators, smart phones, and network access to irrelevant content

You will be using your laptops in class. That's the only kind of networked device I will permit: cellphones, calculators, and smart watches are not to be in evidence in class, or during exams. Period.

During class you are not to access anything that is not directly relevant to the work at hand: no visits to social media sites, or logins to your email accounts. During exams I may insist that you turn off your laptop's wireless networking hardware.

Extra Credit

There will be occasional opportunities to get extra credit. This will include additional exercises/part of exercises as well as additional assignments. To zeroth order these exist because I think they are cool and useful for understanding computational physics but I can't justify within the 2 credit hours of the course.

Extra credit assignments will often be described poorly (maybe even something like, `get a full solar system simulation working'). If you have questions about it, please ask before you spend too much time on it. Also, we have no obligation to make extra credit typo-free. Please try to answer the question we mean to be asking.

For the extra credit, per exercise, the grading is all or nothing. We aren't going to hunt for typos and give partial credit for sortof working code. The amount of extra credit per exercise/etc is listed on the assignment.

About using code you find on the web

The quickest way to deal with the arcana of programing is to ask Google for examples of what you are seeking to accomplish. But you will need to use your judgment in doing this: the Google search “how do I use color maps in python?” is fine, while “show me a script that calculates pi” is not. And you should always credit the original source of code that you paste into your own programs in a comment that includes the URL for the original code. If an author says that his/her code is not to be copied or incorporated into your programs, then DON'T.

I have two principal goals in this course. I want all of you to become fearless coders with the confidence to walk up to baffling problems and pound them into submission. And I want you to develop numerical descriptions of cool systems normally thought to be too difficult for students at your level, whose analytic descriptions might obscure the underlying physics. For this to work, you'll need to write your own code.

Academic integrity

You must never submit the work of someone else as your own. We understand that many of you will find it helpful to work with other students to master Physics 298 owl. But when you collaborate with your study group on homework assignments, you must be a full, active participant in developing the solutions that you submit for credit.

It is cheating to receive answers from another student and then use them as your own. It is cheating to submit as your own work solutions that you find by searching on the worldwide web (though see "About using code you find on the web"), or by subscribing to an online service that suborns cheating. It is cheating—and a violation of U.S. copyright law—to give (or sell) course material to someone else who intends to redistribute and/or sell it.

Cheating will be penalized harshly: I will award zero credit for any assignment in which a student is found to have cheated. I will also probably reduce your course grade by two letter grades (so that an A becomes a C), though I reserve the right to issue an F for the entire course to any student who is found to have cheated.

All activities in this course, including documentation submitted for petition for an excused absence, are subject to the Academic Integrity rules as described in Article 1, Part 4, Academic Integrity, of the Student Code.

Programming resources

- **Numpy numerical library**
  Numpy Reference
  Numpy mathematical functions

- **Matplotlib graphics library**
  http://matplotlib.org/

- **Continuum Analytics Anaconda Python downloads**
  https://www.continuum.io/downloads

- ***Learn Python the Hard Way*: a primer with attitude!**
  http://learnpythonthehardway.org/book/

- ***Learn X in Y minutes Where X=python3***
  https://learnxinyminutes.com/docs/python3/

Please let me know about other references you find to be useful. I will add them to the list.

Textbooks and so forth

There are no required texts. You must come to each class (including the first) with a laptop. Please be sure to bring your power adapter too. If you'd like a Python text (at a highly reduced price), consider picking up a copy of Hans Petter Langtangen, *A Primer on Scientific Programming with Python, 5th ed.*(2016). It is the required text for CS 101. Quoting from the CS 101 web site, "Please note that softcover copies are also available through Springer at a price much reduced relative to the hardcover edition. (You'll need to be on the campus network or log in via the library to see the offer.) An e-book is also available through the SpringerLink service, although we highly recommend acquiring a physical textbook."

Disability Access

(https://www.disability.illinois.edu/academic-support/instructor-information/examples-disability-statements-syllabus)
The Department of Physics is committed to being an open and welcoming environment for all of our students. We are committed to helping all of our students succeed in our courses.

To obtain disability-related academic adjustments and/or auxiliary aids, students with disabilities must contact the course instructor and the Disability Resources and

Educational Services (DRES) as soon as possible. To contact DRES, you may visit 1207 S. Oak St., Champaign, call 333-4603, e-mail disability@illinois.edu or go to the DRES website. If you are concerned you have a disability-related condition that is impacting your academic progress, there are academic screening appointments available on campus that can help diagnosis a previously undiagnosed disability by visiting the DRES website and selecting “Sign-Up for an Academic Screening” at the bottom of the page.