PHYS 2300 Scientific Computing for Physical Systems

Spring Semester 2019
TY-127
Monday & Wednesday: 2:30 pm to 3:45 pm

Instructor: Dr. Hugo Valle
Office Hours: Tuesday and Thursday from 10:30 am to 12:00 pm. By appointment.
Office: TE 110D
Phone: 801-626-7684
Email: hugovalle1@weber.edu **(preferred method of contact)**

Required book: Newman's Computational Physics (ISBN 9781480145511)
Website: Canvas

COURSE DESCRIPTION AND OBJECTIVES:

An introduction to computer programming and fundamental numerical algorithms as used for problem solving and visualization in the natural sciences. Applications may include nonlinear dynamics, chaos, many-particle systems, and Monte Carlo Techniques.

Prerequisites:

- MATH 1200 (co-requisite)
- PHYS 2210 (pre-requisite)

Course Outcomes

Upon successful completion of this course, the student shall be able to master the following concepts:

- **Data manipulation and analysis**: This is the most prevalent form of scientific computing. It can be as simple as using a spreadsheet program to analyze data recorded in a laboratory class or as complex as manipulating terabytes of information from a large astronomical survey telescope. The main features are frequent reading and writing of data files, statistical analysis of large data sets, reformatting and comparing output from multiple experiments, interpolating data from different instruments and data sets, etc. Develop a method for studying the world that is based on experimental verification and problem solving.
- **Modeling**: In modeling, the researcher generally takes a well described and self-consistent description of a problem – the model – and compares it to data. For example, one might develop a model for thermal and radiative diffusion is a star from basic physics knowledge, and then compare that model to measurements of the surface temperature and spectrum of the Sun. The point of modeling is that the self-consistent description based on physical principles can often give the research access to information that is not directly observable. However, the results of the model are nearly always compared to direct observables – or used to predict possible direct observables. Thus, some of the

skills learned in the previous section are used to analyzed the model results and compare it to data. In fact, many modelers refer to the output of their code as “data” or “models” and often a non-observable physics “simulation” quickly becomes a “model” for a direct laboratory experiment as technology improves.

- **Simulation**: Simulations are also self-consistent models of a physical system, but often have little or no connection to observable data. These can be simulations that are theoretical or of experiments that are too complicated or expensive to perform in the laboratory. The benefit of simulation is that it allows researchers to explore the interactions of physical systems even if those systems cannot be directly observed. Also, simulations are generally higher in visualization and user interaction than models. There is certainly a gray area between “simulations” and “models” and often a non-observable physics “simulation” quickly becomes a “model” for a direct laboratory experiment as technology improves.
- **Software engineering**: As you learn and research more topics, your software will grow and the scope of your project with it. In this class you will also learn the fundamentals of software engineering and software development process. You will create programs in Python using a clean, conceptual approach, and will also learn source control for your code such as git and github.

Class Format

Class will consist of lectures and programming assignments. We will use the class time to introduce new programming concepts and to discuss and work on the programming assignments and projects. We will have access to the Computational Physics Lab and all the software you will need for the course. However, it is recommended to setup your own system for this class.

Students with Disabilities

Students who have special needs or disabilities that my affect their ability to access information and/or material presented in this course are encouraged to contact the Service for Students with Disabilities (SSD), on campus at 801-626-6413 for additional disability related educational accommodations. You are not required to disclose these abilities to your instructor, but the instructor can only accommodate accommodation requests that officially come through the DSS.

Allocated Time

You should anticipate spending two to three hours of study per week for each credit hour of a university course. Computer and programming classes typically require time in the upper range (10-12 hours).

Grading

Your final grade will be calculated according to the following chart:

Programing Assignment	**70%**
Final Project	25%
Attendance/Participation	5%

Programming Assignments (70% of grade)

There will be **several** programming assignments. You will turn in your assignments by uploading your programming files to Canvas. Following the required naming convention is critical to speed up the grading so please name your folders and files exactly as specified in the lab description and ensure that your program is executable.

When working on your assignments, if you are stuck for more than a couple of hours, submit your question to an existing or create a new discussion board on canvas, stop by my office, or seek help with our **tutoring center**: http://icarus.cs.weber.edu/Tutoring_and_Resources.html.

If you plan to do software and/or hardware development after graduation, you will almost certainly need to know how to work in groups. For some of the assignments in this class, you may need to work in groups of 2 people; the assignments will be the same no matter what size group you have. In order to ensure everyone in the group does their fair share of the work, we will ask each of you to turn in assessments of the relative contributions of your project partners. Many employers do this, by the way, in determining salaries and bonuses.

Assignments/Class Questions:

Questions on reading and programming projects should be posted to the appropriate online Classroom Discussion Groups where the instructor and all students have access. Instructor and student responses will be visible to everyone in the class.

Late Submission Policy:

We will use flexible slip dates for the programming assignments or quizzes. Each student is given an extension of 3 calendar days. You can use the extension on any programming assignment or quiz during the semester, with a maximum of 3 slip days per assignment. For instance, you can hand in one assignment up to 3 days late, or three assignments 1 day late. For group assignments, the slip date will be deducted from each team member's remaining slip time. This should let you schedule around the due dates for other courses.
After you have used up your slip dates, any assignment submitted late will be penalized according to the following chart:

1 Day Late	-15%
2 Days Late	-30%
3 Days Late	-45%
4 or more Days Late	-100%

Note: Any assignment submitted after 3 days will receive no credit.

Attendance/Participation (5% of grade)

Attendance (2.5%)

This class is a face-to-face format so attendance and participation is part of your grade. I will begin taking attendance after the first week of classes, this will count for 2.5% of your grade. In order to receive full credit (2.5%) you need to attend to 80% of the classes. Table 4: Grading will be used to grade your attendance.

Attendance Percent	Grade Percentage
80% - 100%	2.5%
75%-79%	2.0%
70%-74%	1.5%
65-%69%	1.0%
60%-64%	0.5%
Below 60%	0 %

Table 1: Attendance

Class Participation (2.5%)

The other 2.5% of your grade is based on the teacher's evaluation of yourself through the semester. This includes participation in class, and improvement through the semester. You will get three evaluations through the semester, usually after each exam

Final Independent Project (25% of grade)

In addition to the programming assignments, your grade will consist of a final project. The project could be an extension (with significant additional coding) of the earlier projects, or a simulation of some other mechanical or thermodynamic system, or a simulation of other branch of physics (electrodynamics fields, waves, quantum mechanics), or a simulation relevant to some other scientific discipline. See Canvas for more details.

Minimum Technical Requirements:

- Anaconda Python Distribution: https://www.anaconda.com/download/
- Visual Python package: http://vpython.org

LETTER GRADING SCALE

Grading:

You may view your current grade at **www.canvas.weber.edu**. Please check Canvas daily for important announcements and grade updates.

Unofficial Withdrawal (UW): The following actions will result in a student receiving a UW grade:

1. Missing the final test.
2. Not finishing a class that *was* started and *was not* officially dropped.

Note: A UW may negatively affect your Visa status and/or scholarship, if applicable.

Letter Grade	Percentage Grade
A	94-100%
A-	90-93.9%
B+	87-89.9%
B	84-86.9%
B-	80-83.9%
C+	77-79.9%
C	74-76.9%
C-	70-73.9%
D+	67-69.9%
D	64-66.9%
D-	60-63.9%
E	Below 60%

THE "I REALLY, REALLY NEED TO GET A C" POLICY

The most effective method for obtaining a C or above in this class is to submit assignments when they are due and to stay current with course topics. The curriculum is carefully designed to fit the number of course weeks. In order to uphold academic rigor and integrity, student grades must be based on the degree to which the course requirements listed in the syllabus are fulfilled. Extra credit assignments are not allowed. If you approach me anytime during the term claiming that special allowance should be made because you need a C to move forward in the program, graduate, receive financial aid, etc., I will decline your request and refer you to this clearly worded policy

Proper Use of Email

Always include your course name and section in the subject header in any email. I normally answer email messages within 48 hours (2 business days). But, remember that most answers **could be found in the syllabus** or in the course's Canvas page. If you do not get a response within 48 hours, please resend your email message.
*Note: If you do not address the email to me, I **will not** respond to your email.*

Academic dishonesty or Cheating

Students are expected to maintain academic ethics and integrity in regards to performing their own work. The WSU Student Code states and clarifies cheating.

Cheating, which includes but is not limited to:

1. Copying from another student's test paper.
2. Using materials during a test not authorized by the person giving the test.
3. Collaborating with any other person during a test without authority.
4. Knowingly obtaining, using, buying, selling, transporting, or soliciting in whole or in part the contents of any test, without authorization of the appropriate official.
5. Bribing any other person to obtain any test.
6. Soliciting or receiving unauthorized information about any test.
7. Substituting for another student or permitting any other person to substitute for oneself to take:
    a. Exam.
    b. Plagiarism, which is the unacknowledged (uncited) use of any other person or group's ideas or work. This includes purchased or borrowed papers.

c. Collusion, which is the unauthorized collaboration with another person in preparing work offered for credit.
d. Falsification, which is the intentional and unauthorized altering or inventing of any information or citation in an academic exercise, activity, or record-keeping process.
e. Giving, selling or receiving unauthorized course or test information.
f. Using any unauthorized resource or aid in the preparation or completion of any course work, exercise or activity.
g. Infringing on the copyright law of the United States which prohibits the making of reproductions of copyrighted material except under certain specified conditions.

Depending on its severity, it may result in a failure of the assignment/paper, or even the course. All cases of academic dishonesty will be reported to the Dean of Students. If you have questions or concerns, please refer to the WSU PPM (Policies and Procedures Manual).

CS Department policy dictates that any verifiable evidence of students' academic cheating, as defined by the instructor, will result in:

1) Automatic failing grade for the class
2) Report to the Dean of Students that will include the student's name and description of the student's dishonest conduct.

Other Important Information

Cell phones (face-to-face): use the vibrate mode only. If you need to answer a call, please do so outside the classroom. Absolutely no text messaging allowed. If you must take an emergency call or page, quietly leave the classroom to conduct your conversation. We will be using computers in classrooms. Please ensure that all classroom computer activity is directly related to the lecture or assignment.

Course Fee Statement

Course fees for the Computer Science major are designed to cover the costs of lab equipment maintenance and replacement including desktop and server computer systems and software; consumable materials and supplies; and support for lab aides, student tutors, and online instructional resources.

Emergency campus closure: In the event of an extended campus closure, I will continue to provide instruction by utilizing Canvas, the online course system. I will expect you to log in to the system on a regular basis to keep up with coursework. Assignments will be provided through the online system with clear due dates and expectations. Discussions will be made available to allow you to interact with other students and me about course material. I will check my Weber email on a daily basis should you need to communicate with me personally. It is imperative that I am able to contact you and that I have accurate contact info on you. You are responsible for checking your Weber e-mail or for having Weber messages forwarded to accounts you do check.

DAILY SYLLABUS

(Tentative. Please follow Canvas calendar)

Module	Topic	Assignments
1	• Configuring your system • Physics Basics • Introduction to Jupyter notebook • Input/output • Variables part I	Program M1
2	• Tod Down Design • Variables part II • Conditionals	
3	• List and Arrays • Flow control	Program M2
4	• Functions • Graphing	
5	• Accuracy and Speed • Objects and classes	Program M3
6	• Integrals • Derivatives	
7	• Data Analysis • Interpolation	Program M4
8	• Projectile Motion	
9	• Pendulum	Program M5
10	• Gravitational Dynamics	
11	• Molecular Dynamics	Program M6
12	• Random numbers • Montecarlo Simulations	
13	• Final Project	Program M7
14	• Final Project	Final Project