METHODS OF APPLIED PHYSICS II--COMPUTERS IN PHYSICS

PHYS 6302 - 001
Unique Number: 53599
Fall 2019

Instructor:	Amir Shahmoradi
office:	SEIR 365
e-mail:	a.shahmoradi@uta.edu
office hours:	Thursdays 3:30-5:20 pm

Class start/end:	June 3, 2019 - August 8, 2019
Lecture meeting times:	Tuesdays – Thursdays 10:30 am – 12:30 pm
Lecture meeting place:	Life Science (LS) 428

Teaching Assistants:	NONE
office:	NONE
e-mail:	NONE
office hours:	NONE

COURSE OBJECTIVES / ACADEMIC LEARNING GOALS

This is the second of a two course sequence in Method of Applied Physics, providing the necessary foundations in scientific computing for Physics majors at the graduate level. It introduces operating systems, programming languages, and tools using examples and contexts from physical sciences. **Prerequisite**: None (some level familiarity with programming is desired).

The primary objective of this course is to learn basic computer programming concepts and apply them to Physical Science related problems. By the end of the course, you should have a good understanding of general programming foundations and practices, and be able to analyze physics problems and develop computational solutions for them, collaboratively within a team. We will achieve this by learning how to program in popular operating system environments such as Linux using popular high-level programming languages such as MATLAB/Python or high-performance programming languages such as modern Fortran/C/C++. Although not essential, some prior level of familiarity with programming concepts is desirable for this course.

COURSE SCHEDULE

The following is a tentative outline of topics to be covered:

- Version Control Systems (VCS): Principles of professional project management and collaborative programming with the use of Git.
- Programming History – Operating Systems – Round-off and Truncation Errors – Number Systems – Introduction to Cloud Computing Resources.
- Brief Introduction to Principles of Programming using Python/Matlab as the main language: Python/Matlab development environment – general installation guidance –

available editors – syntax rules – variables and data types – conditionals – looping – input/output – functions – Exception Handling – Object Oriented Programming – Array Computing and Performance Optimization – Wrappers, and Cross-language Interoperability.

- Applications of Programming in Physics: Matrix Operations – Data Manipulations – plotting – symbolic calculations – libraries (e.g., Numpy, Scipy, Pandas, scikit-learn, mlpy, seaborn, …).

COURSE TEXTBOOKS

No textbook is required for this course. Online class lecture notes will be used as reference. However, a list of textbooks for those who are interested to self-educate themselves or go beyond class syllabus will be provided on the first day of the class.

COURSE LOGISTICS

Grading:
Weekly Homework: 32% (Assignments might not be weighted equally)
Weekly Quizzes: 32%
Final Project: 36%

Homework Policy:
There will be approximately one homework per week. Assignments will be due before lecture begins, and should be added to an online repository determined by the instructor. No late assignments will be accepted. No exceptions to the homework policy will be made without prior instructor approval.

Examinations:
There will be no midterm exam, and no final exam. Students will have to complete a project in place of the final exam, either individually or in collaboration with teammates who are determined randomly after the midterm.

Attendance:
Regular attendance is expected. Any absence requires prior approval from the instructor, or compelling evidence of illness or an official letter from the university administration. Student attendance will be randomly checked.

Scholastic dishonesty: All students are responsible for upholding the University rules on scholastic dishonesty. Students who violate University rules on scholastic dishonesty are subject to disciplinary penalties, including the possibility of failure in the course and/or dismissal from the University. Since such dishonesty harms the individual, all students, and the integrity of the University, policies on scholastic dishonesty will be strictly enforced.

Other matters: The University of Texas at Arlington provides, upon request, appropriate academic adjustments for qualified students with disabilities. Any student with a documented disability (physical or cognitive) who requires academic accommodations should contact the UTA's Office

for Students with Disabilities as soon as possible to request an official letter outlining authorized accommodations. For visit https://www.uta.edu/disability/.

Your Expectations:

For the fall 2019 offering of this course, we will cover the principles of computer programming using Python/Matlab programming language. Specifically, upon completion of this course you will be familiar with,

- programming paradigms,
- principles of software maintenance and collaborative project development,
- differences between compiled and interpreted programming languages,
- how to use Python/Matlab as a simple calculator,
- how to use Python/Matlab as an advanced scientific computation and graphics toolbox,
- how to use Python/Matlab to collect, clean, and analyze data and make a scientific inference,
- how to formulate and cast a scientific problem in the form of a sequence of computational algorithms.

Course Schedule:

The following is the **tentative** schedule of topics to be covered, and will be continuously updated. The exam dates are final.

Tue Jun 04	student-professor connection day; course outline
Thu Jun 06	VCS - Version Control System (Git, Mercurial)
Tue Jun 11	Python/Matlab - environment setup, history, variables and types **(HW 1 due, quiz 1)**
Thu Jun 13	Python/Matlab - operators, branching, control statements, and loops
Tue Jun 18	Python/Matlab – functions and lambdas **(HW 2 due, quiz 2)**
Thu Jun 20	Python/Matlab – strings, hash tables, and dictionaries
Tue Jun 25	Python/Matlab - input/output (IO) **(HW 3 due, quiz 3)**
Thu Jun 27	Python/Matlab - input/output (IO)
Tue Jul 02	Python/Matlab – plotting techniques **(HW 4 due, quiz 4)**
Thu Jul 04	Python/Matlab – interoperability and high performance computing
Tue Jul 09	Python/Matlab - OOP: classes **(HW 5 due, quiz 5)**
Thu Jul 11	Python/Matlab - OOP: inheritance
Tue Jul 16	Python/Matlab - OOP: encapsulation **(HW 6 due, quiz 6)**
Thu Jul 18	Python/Matlab - OOP: polymorphism
Tue Jul 23	Python/Matlab - OOP: virtual classes and dynamic binding **(HW 7 due, quiz 7)**
Thu Jul 25	Python/Matlab – unit-testing and exception handling
Tue Jul 30	Python/Matlab – numerical simulations **(HW 8 due, quiz 8)**
Thu Aug 01	Python/Matlab – probability theory and stochastic processes
Tue Aug 06	Python/Matlab – machine learning and deep learning techniques
Thu Aug 08	Semester wrap-up **(Final project due)**