Computer Modeling for Science Majors
CSCI 250
Fall 2017 Syllabus

> The sciences do not try to explain, they hardly even try to interpret, they mainly make models. By a model is meant a mathematical construct which, with the addition of certain verbal interpretations, describes observed phenomena. The justification of such a mathematical construct is solely and precisely that it is expected to work – that is, correctly to describe phenomena from a reasonably wide area.
>
> –JOHN VON NEUMANN

Instructor Details

Name:	Jesse Johnson
Office:	406A Interdisciplinary Science Building
Telephone:	(406) 243-2356
Email:	`jesse.johnson@umontana.edu`
Web:	http://hs.umt.edu/hs/faculty-list/faculty-details.php?id=540
Office Hours:	MWF 15:00–16:00 , Interdisciplinary Science Building 406A
	Or, by appointment.

Prerequisites

Students taking this course are expected to have:

- basic computer literacy to include file systems, use of the terminal, and software installation,
- mathematical maturity to include some calculus, and
- familiarity with the scientific method,
- *or, consent of instructor.*

Course Objectives

This course will provide a hands on introduction to computer programming by examining and solving the types of computing problems that arise in the physical and biological sciences. In spite of the applied nature, the course will remain true to best practices in computer science by using a modern, object oriented language for instruction. Throughout, emphasis will be placed on scientific problem solving, critical thinking, communication of scientific data, and the scientific method.

After taking this course, students will be able to:

- write short, modular programs of moderate complexity in Python,
- apply numerical methods to solve scientific problems,
- use advanced data structures including lists, dictionaries, and arrays to aid scientific computing,
- use specialized hardware and sensors to acquire data from the environment and make it internet accessible,
- transform mathematical concepts to working computer programs.
- quickly adapt to using other common scientific computing environments such as R, Fortran, or Matlab,

Course Format

This course has two parts.

The first focuses on learning the Python programming language, and will take about 10 weeks. This will be done using a hybrid of a website called "Codingbat" and problem solving sessions during class.

You will work through the Coding bat problems outside of classroom time, and I will assess your progress by monitoring your progress. There will also be quizzes, exams, and daily problem sets. When you enter class, you will be expected to have completed the assigned material from Codingbat, and be prepared for discussion of the material and problem solving sessions that involve the new material. Think of the Codingbat as a textbook with advanced features that compel you to do what the instructor asks.

In class, we will:

- Discuss the relevant Codingbat material.
- Answer any questions about the Codingbat material.
- Work together on short scientific problems that can be completed using concepts learned from Codingbat and short, in-class lectures.

The second part of the course corresponds to the last 5 weeks of class. During this time, students will use what learned from the Codingbat to do 3 larger projects in scientific modeling and computation. The approach to this will be:

- I will lecture over the topic providing context, motivation, background, mathematical basis, and algorithmic approach to solution.
- Students will begin implementation of the projects early in the period, allowing lectures to be given while students are actively seeking solutions.
- In addition to producing functional code, students will turn in short, written reports on the projects.
- One of the assignments will involve using sensors to collect data. The sensors will be controlled by a small, inexpensive computer.

Online Resources

- All course material will be made available online, through the University of Montana's Moodle system.
- Codingbat
- Anaconda Python Let's use the Python 2.7 version.

Textbook

I will be drawing problems for the in class sessions from the following text.
A Primer on Scientific Programming with Python (Fourth Edition)
Hans Peter Langtangen
2014, Springer
ISBN-13: 978-3642549588

Supplemental Text

Many students enjoy a concise reference for Python programming. There are many good online tutorials, but this is a good, inexpensive ($9.10) reference.
Python Pocket Reference
Mark Lutz
2014 O'Reilly Media
ISBN-13: 978-1449357016

Software

For Codingbat, all that you will require is a web browser and registration on the site. For the in class sessions, and the last five weeks, you'll need a working version of Python on your computer. I recommend Anaconda Python as an easy to install solution.

- Students are encouraged try the Anaconda Python distribution, available here. It provides Python, all of the scientific packages we use in class, and a nice Integrated Development Environment (IDE) called Spyder which we will also use in class.

- Code visualization This is a great resource for beginners that would like to have a sketch of how the Python code executes. Try things out here first.

Meeting Times/Place

Times: Monday, Wednesday, Friday 11:10–12:00
Place: Social Science 362

Final Exam Time and Place

8:00–10:00 Monday, December 18, 2017
Social Science 362

Grading Policy

Grading scale

A	94-100
A-	90-93
B+	87-89
B	83-86
B-	80-82
C+	77-79
C	73-76
C-	70-82
D+	67-69
D	63-76
D-	60-62
F	0-59

Students taking the course pass/no pass are required to earn a grade of D or better in order to pass.

Assessments and weights

The following assessments will be used and weighted according to the values in the table to determine final grades.

Component	Description	Number	Weight
In class assessment	A combination of attendance and successful completion of in-class activities which will include think-pair-share teams, reading assessments, group problem solving, brainstorming, one-minute questions, strip sequences, decision making, concept mapping, quizes, and problem based learning .	-	20%
Codingbat materials	Assessment of student performance on Codingbat.	-	30 %
Projects	In depth application of programming techniques to questions arising in the sciences.	2-3	30%
Exams	Tests of your knowledge of basic programming and applications to science.	2	20%

Tentative Schedule:

MONDAY	WEDNESDAY	FRIDAY
Aug 28th **1**	30th **2**	Sep 1st **3** Python data types, writing scripts
4th *Labor Day*	6th **4** Functions, logical tests and return	8th **5** Conditionals I
11th **6** Conditionals II, Strings	13th **7** Writing scripts, using lists	15th **8** `math`
18th **9** Computing formula I	20th **10** Computing formula II: Functions	22nd **11** Loops I
25th **12** Loops II	27th **13** Lists I	29th **14** Lists II
Oct 2nd **15** Strings	4th **16** Dictionaries	6th **17** Libraries and scripting
9th **18** `numpy`	11th **19** Command line arguements	13th **20** Reading files
16th **21** Plotting data	18th **22** Fitting curves	20th **23** Simulation I
23rd **24** Simulation II	25th **25** `csv`	27th **26** `urllib`

Monday		Wednesday		Friday	
30th system	**27**	Nov 1st Histograms	**28**	3rd Contour plots	**29**
6th Integration	**30**	8th Differentiation	**31**	10th *Veteran's Day*	
13th Symbolic mathematics	**32**	15th **Exam I**	**33**	17th Recursion I	**34**
20th Recursion II	**35**	22nd *Travel Day*		24th *Thanksgiving Break*	
27th Objects I	**36**	29th Objects II	**37**	Dec 1st Project I: Infectious disease	**38**
4th Solving ordinary differential equations I	**39**	6th Solving ordinary differential equations II	**40**	8th Time to work	**41**

Attendance Policy

Attendance is required and enters your grade as part of the in class assessment (30% of grade). The policy for excusing absences is identical to that of late assignments.

Late Assignments

Other than in in exceptional circumstances, such as family or medical emergencies *late homework will not be accepted* unless an extension was agreed upon *well in advance* of the due date.

Academic Integrity

All students must practice academic honesty. Academic misconduct is subject to an academic penalty by the course instructor and/or a disciplinary sanction by the University. All students need to be familiar with the Student Conduct Code. I will follow the guidelines given there. In cases of academic dishonesty, I will seek out the maximum allowable penalty. If you have questions about which behaviors are acceptable, especially regarding use of code found on the internet or shared by your peers, please ask me.

Disabilities

Students with disabilities may request reasonable modifications by contacting me. The University of Montana assures equal access to instruction through collaboration between students with disabilities, instructors, and Disability Services for Students. Reasonable means the University permits no fundamental alterations of academic standards or retroactive modifications.