
# PHY/MAT 131 Syllabus: Foundations of Python Programming

*Course Philosophy*
In this course, you will be using Python as a way to begin to understand computer science. Computer science, like many liberal arts disciplines, is a way to approach problem solving. During the semester, you will be learning the specifics of the Python programming language, but more broadly you will be learning to think about and solve problems that are amenable to computer science solutions.

*Text Book*
We will be using an online textbook called Fundamentals of Python Programming. It is available at https://runestone.academy/ You are pre-registered for use of the textbook for this course and will be provided with your account name and password on the first day of class. Please use only that account for the duration of this course.

*Goals for the Course*
1. To address the question: "What is Computer Science?"
2. To get you excited about programming. You can do this!
3. To develop your problem solving skills.
4. To widen your view of the range of disciplines to which computer science is important.
5. To help you to learn the Python programming language.
6. To help you to program as a way of expressing solutions to problems.

*Objectives and Specific Skills*
- To successfully use fundamental programming constructs and data structures
- To understand that debugging is an essential part of programming, maybe even the main part of programming
- To master fundamental programming patterns for problem solving
- To learn to cope with complexity

*Homework and Class Preparation*
In this class you will have daily assignments, these will include reading assignments from the online textbook. The online textbook will provide a flipped classroom style of learning, meaning that I will not be spending a lot of time lecturing, your primary source for **new** information will come from the book. **The reading assignments are active assignments, you cannot just passively read the material you must interact with it. I will be able to gauge your level of interaction with the material from the online system.**

*Tips for success*

- Programming can be notoriously time consuming. Budget your time and start early to avoid frustration. Your program is almost never going to work right the first time, so you need to accept that and be comfortable with it taking several tries to get right.
- Start Small. A small program that is partially working is worth more than a large messy program that crashes.
- Always always feel free to ask questions!
- The materials in this class all build on each other. Therefore it is really important to keep up. You can't ignore one topic and hope it will go away as once we have introduced a new idea we will continue to use it throughout the course. If you do get behind let's talk so we can devise a strategy to get you caught up.
- It is almost inevitable that technical problems will arise. When they do, remember the words of Douglas Adams: "Do Not Panic." I will not punish someone with a bad grade due to a technical glitch.
- In Class – We will be using computers in class every day, so be wary of distractions. It is very tempting to check Facebook, email, twitter, or the score of the Arsenal match when the computer is in front of you. Don't do it!! It not only distracts you but everyone around you.

*Grading*

Your grade in the course will be determined using the following percentages. Participation will include attendance and activity/participation in class.

| | |
|---|---|
| Reading Assignments (including inline questions) | 30% |
| Participation (Attendance, In Class Activity) | 10% |
| In-Class Exercises and Projects | 30% |
| Exams (midterm and final, 15% each) | 30% |
| **Total** | **100%** |

*Autograded Work*

- Much of the homework will be automatically graded, you will get immediate feedback as you run and test your work. You are encouraged to keep working and to get as many of the automatic tests to pass as you can. You WILL NOT be penalized for making lots of tries.
- The history slider on the programming problems in the book is there to encourage you to explore and try new things. Every new version of your program is saved, but you can always go back to a previous version if something you try doesn't work.

*Honor Code*

Work performed in this course must be in accordance with the Agnes Scott College Honor Code. Students will sign a statement of the code for each exam.

*Course Evaluation*

Your feedback on the course is extremely valuable to me, the department, and the administration. In particular, I take your comments very seriously and use them to improve the

course the next time I teach it. **The completion of course evaluations is an expectation of students in this class. Near the end of the semester you will be notified by e-mail and provided with a link to follow to complete the evaluations on line outside of class.**

*Title IX Statement*
For the safety of the entire community, any incidence of or information about sexual misconduct must be reported immediately to Title IX Coordinator Karen Gilbert (kgilbert@agnesscott.edu, 404-471-6435).

*Inclusion*
This course adheres to the principles of diversity and inclusion integral to the Agnes Scott community. We respect people from all backgrounds and affirm people's decisions about gender expression and identity. Please feel free to correct me if your preferred name or gender pronoun are different from that listed on the class roster.

*Late Work & Attendance*
Work will lose a ½ grade for every day work is late. For example, on a 20 point assignment, you would lose 1 point per day late. You will receive 100% for attendance if you miss no more than 2 classes. After 2 absences, you will lose 10% of your attendance grade for every absence. For example, if you miss 4 classes during the semester, your attendance grade will be 80%.

*Schedule*
Please refer to the online Canvas course for the course schedule. The class will be organized around Projects, and the Chapters/Reading that supports those projects. The Projects will take between 1 and 3 classes (in general).

Exam I will be given the week of October 6 (in class), and Exam II will be given during the exam period.

| **Project Name (number of classes in parentheses)** | **Reading & Practice (complete before first class of the week)** | **What We Will Do In Class** |
|---|---|---|
| *A Preview of the End Goal (1)* | N/A | Setting up Runestone<br>Your First Program |
| *Driving the Turtle (1)* | Chapter 1 - General Introduction<br>Chapter 2 - Variables, Statements and Expressions | Project Work |
| *How Many Handshakes? (1)* | Chapter 3 - Debugging<br>Chapter 4 - Python Modules | Project Work |
| *Drawing a Circle with Turtle (2)* | Chapter 5 - Python Turtle | Project Work |
| *Generating a Password XKCD Style (2)* | Chapter 6 - Sequences | Project Work |

| | | |
|---|---|---|
| *Computing Statistics with Kiva Data (2)* | Chapter 7 (7.1-7.7) - Iteration<br>Chapter 12 - Functions | Project Work |
| *Graphing Kiva Data with Altair (2)* | Chapter 8 - Conditionals | Project Work |
| *MIDTERM EXAM* | | |
| *Substitution Cipher (2)* | Chapter 9 - Transforming Sequences | Project Work |
| *Image Processing (3)* | Chapter 7.8 - Nested Iteration | Project Work |
| *Exploring Common Words and SETI (2)* | Chapter 10 - Files | Project Work |
| *Monte Carlo Simulation (2)* | Chapter 10 - Functions (Exercises) | Project Work |
| *Converting Roman Numerals (2)* | Chapter 11 - Dictionaries | Project Work |
| *Stack Overflow Developer Survey (2)* | Chapter 11 - Dictionaries (Exercises) | Project Work |
| *Predicting Pizza Prices - Linear Regression (2)* | Chapter 12 - Functions (Exercises) | Project Work |