This course provides a path for students with a broad diversity in backgrounds to learn programming and computer science, including those considering a computer science major. Students will learn at their own pace with a combination of technology-based and face-to-face class structure. Students will use a problem-driven approach to build complex, interactive software systems. The course includes an introduction to a wide variety of issues relating to software, including design, problem-solving, development processes, and broader issues such as security, performance, and ethics.

**Restriction:** Must not have completed any courses from the CMSC131–499 course range.

This web page is the **syllabus** for this course sequence.

**Prof. Ben Bederson**  
3171 A.V. Williams Building  
<my last name>@cs.umd.edu [preferred]  
301-405-2764  

**Office Hours:** TBD  

**Class Location/Time:** TBD  

**Technology Requirements:** You must bring a laptop to every class. If you would like to
take the course, but do not have access to a laptop, contact the instructor as a small number of "loaner" laptops for class time will be available for this purpose.

**Textbook / Resources:**

- Course video: Lectures that provide the core "content" of the course will be provided online
- No required textbook
- Optional resources (all freely available online):
  - Online class: Udacity CS101. This teaches Python to people that have no programming experience. It includes video lectures and integrated short quizzes with an online programming environment.
  - Online exercises: Codecademy offers freely available exercises to learn the basics of Python operations.
  - Online tutorial: The "standard" tutorial for learning Python available from the distributors of Python.
  - Book: "How to Think Like a Computer Scientist" introduces programming and computer science using Python.
  - Book: "Non-Programmer's Tutorial for Python 2.6" also introduces programming and computer science using Python.

This is an introduction to computer science for non-majors. There are no prerequisites except for a willingness to try something new.

This will be a problem-driven course. In this class, we will build a simulation with virtual creatures that evolve, interact with each other (and your classmates' creatures), and respond to the changing environment. By building these creatures, you will learn about basic programming concepts, and about how programs interact with the real world (your creatures will be able to text you, and more!) You will learn about how computers can process textual data, simulate complex environments, and how even a simple computer program can result in an exciting level of complexity when networked with other programs.

You will program in Python, a widely used computer language of growing popularity. Your programs will run on a server, enabling you to access your creatures from the web, and even your phone - so you can show off your work to your friends, and encourage them to help take care of your "pets".

This class is an experiment with a new kind of course structure. It is "Mastery Based" - which means that you will need to excel at each level (by demonstrating mastery) before moving on to the next unit. Thus, you will learn at your own pace, earning 1 credit at a time (one credit per course number). By the end of the semester you will have the opportunity to earn between 1 and 3 credits, depending on how far you get.

- The course will be broken up into 12 units where 4 units correspond to each credit. Students will work at their own pace, only being allowed to move on to subsequent units when you have mastered the current one (i.e., earned an "A"). After 4 units, you will have to demonstrate mastery in order to earn the A, and will then be allowed to register for the next credit.

- Students can work at their own pace. To support this, the course will be "flipped". The lectures will be recorded and must be watched online before class. The class time will be for discussion, working on homework, classroom activities with other students, exams, etc.

- At the end of the semester, students will
have the option of getting the grade that they have earned for that credit, or taking an incomplete, and having an opportunity to continue the coursework the following semester to earn an A.

- If you complete all 3 credits, the course will count for GenEd. [DEPENDING ON APPROVAL]

These are the 12 units which you will follow in sequence:

<table>
<thead>
<tr>
<th>CMSC 198A</th>
<th>CMSC 198B</th>
<th>CMSC 198C</th>
</tr>
</thead>
<tbody>
<tr>
<td>4. Ecosystems</td>
<td>8. Talking to others</td>
<td>12. Emergence</td>
</tr>
</tbody>
</table>

Earn 1st Credit  Earn 2nd Credit  Earn 3rd Credit  Counts for GenEd (if approved)

The following technical topics will be covered throughout the three courses, and will be folded in to the above units as needed to support the simulation problem that we will build up throughout the course:

- Computer architecture
  - Memory
    - Types (persistent, volatile)
    - Memory diagrams
    - Heap v. stack
  - Flow of control
- Programming in Python
  - Syntax
  - Basics: variables, operators, expressions, statements, functions
  - Types: Boolean, int, float, string, list, dictionary
  - Conditionals
  - Iteration
  - Lists
  - Functions, return values (multiple return values)
  - Libraries / APIs
    - Text processing, math, data structures
- Program Design
  - Avoiding duplicated code
  - Coding style
  - Commenting
  - Variable, function names
  - Right-sizing functions
- Problem solving
  - Divide and conquer
  - Separation of concerns
- Software development
  - Debugging
  - Testing
  - Source code control
  - Collaborative development
    - Pair programming
    - Code review
- Programming Models
  - Event-driven programming
Homework, Projects and Exams: Each unit will include written homework that must be completed and mastered (i.e., earn an A) before a student can have their work for the next unit graded (see "Grading" below). Each course (i.e., each credit consisting of four units) will also include a more substantial project and an in-class individual exam that must be similarly mastered.

Homeworks and projects must be done individually, but it is ok to give or receive help from others. This means that each student must physically sit at the computer and do all of the typing themselves. I.e., it is acceptable for other students to look at your work and give advice, but other students may not use the computer to type, run, debug, etc. your work.

Exams are closed-book, but students may bring in one 8.5x11" two-sided sheet of paper with hand written original notes. Because students will be doing work at their own pace, exams will be offered each class period. Exams are considered private, and may not be shared or discussed. Exams will also be designed to have variations so each student will see slightly different versions of the exam.

Grading: All grades will be posted at the CMSC Class Web System.

This course is structured around "mastery", which means that students can not go on from one unit to the next until they have demonstrated that they have truly learned the material, which will be measured by earning an A for that unit's homework and exam. Similarly, students will not be able to move from one course to the next unless they have earned an A. If student's work is not at the A level, feedback will be given, and the student will be expected to refine their work responding to the feedback and resubmit. Homework will be graded and returned as quickly as possible (with a goal of 48 hour turn-around) so that students are not delayed, except that re-grades (on refined work) will be returned one week after the earlier grade.

Much of the "content" of the course will be available online through recorded video and exercises. However, attendance in class is required for discussion, working on homework, classroom activities with other students, exams, etc. In addition, students must meet with the instructor in small groups for 30 minute sessions once per credit.

That said, each of the 1-credit courses will be graded as follows:

- 10%: Attendance
- 5%: Meet with Instructor
- 40%: Homework
- 5%: Code review of other student's code
- 20%: Project
- 20%: Exam

Writing: This course involves writing (both code and prose). You will be graded on your writing quality as well as your writing content. Good writing conveys ideas more clearly than poor writing, and the process of writing down your thoughts often helps to clarify them. Thus it is to your advantage to make the effort to organize your thoughts and write well.

Any writing that comes from anyone but you must be clearly and explicitly cited. That is, if you quote a web page, article, person, source code, or anything that you did not write yourself, you must indicate which text is a quote, and where that quote comes from.
**Attendance policy:** Students are required to attend the weekly 3 hour class session. If a student will miss class for a non-medical reason, then it must be discussed with the instructor in advance, and the instructor will make a decision on a case by case basis.

For medically necessitated absences, the student must make a reasonable attempt to inform the instructor of his/her illness prior to the class. Upon returning to the class, the student must present their instructor with a self-signed note attesting to the date of their illness. Each note must contain an acknowledgment by the student that the information provided is true and correct. Providing false information to University officials is prohibited under [Part 9(i) of the Code of Student Conduct](https://sites.google.com/site/introcsumdfall2013/) and may result in disciplinary action. The self-documentation may only be used for only 1 class meeting. Any student who needs to be excused for multiple class meetings must provide written documentation of the illness from the Health Center or from an outside health care provider. This documentation must verify dates of treatment and indicate the timeframe that the student was unable to meet academic responsibilities. In addition, it must contain the name and phone number of the medical service provider to be used if verification is needed. No diagnostic information will ever be requested.

**Academic Honesty:** Please visit the webpage of the [Student Honor Council](https://sites.google.com/site/introcsumdfall2013/) for a detailed explanation of what constitutes academic dishonesty. Note that it includes not only cheating, fabrication, and plagiarism, but also includes helping other students commit acts of academic dishonesty by allowing them to obtain copies of your work. You are allowed to use the Web for reference purposes, but you may not copy any significant amount of code from any website or any other source (i.e., a line or two is acceptable, but a full function is not). In short, all submitted work must be your own.

Cases of academic dishonesty will be dealt with harshly. Each such case will be referred to the University's Office of Judicial Programs. If the student is found to be responsible of academic dishonesty, the typical sanction results in a special grade "XF", indicating that the course was failed due to academic dishonesty. More serious instances can result in expulsion from the university. If you have any doubt as to whether an act of yours might constitute academic dishonesty, please contact the instructor.

**Course Evaluations:** Your participation in the evaluation of courses through CourseEvalUM is a responsibility you hold as a student member of our academic community. Your feedback is confidential and important to the improvement of teaching and learning at the University. When it is announced, please go directly to the [course evaluation website](https://sites.google.com/site/introcsumdfall2013/) to complete your evaluations. By completing all of your evaluations each semester, you will be able to see the evaluation reports for the thousands of courses for which 70% or more students submitted their evaluations.

**Disability Support Services:** Any student eligible for and requesting reasonable academic accommodations due to a disability is requested to provide, to the instructor in office hours, a letter of accommodation from the [Office of Disability Support Services (DSS)](https://sites.google.com/site/introcsumdfall2013/) within the first two weeks of the semester.

**Copyright:** All course materials that exist in a tangible medium, such as written or recorded lectures, presentation slides, handouts and tests, are copyright protected. Students may not copy and distribute such materials except for personal use and with the instructor's permission. Course materials may also be marked copyrighted (© 2013 Bederson).