

## CS111: Program Design I (CS+Bio) Syllabus, Schedule, and Python Installation

CS Instructor: Prof. Patrick Troy	Wed, 12-3pm, 1216 SEO, troy @ uic.edu
Bio Instructor: Prof. Boris Igic	Tue, 3:15-5pm, 4085 SEL, boris @ uic.edu
TA: Zixhuan Ke	Mon 9-10am, SEL 4030, zke4 @ uic.edu
TA: Karolis Ramanaukas	Wed 4-5pm, SEL 4030, kraman2 @ uic.edu
Undergrad. TA: Johnnalee Kutzke	Mon 3-4pm, Tue 3:15-4:15pm Fri 11:45am-12:45pm CS Lounge in SEL 2250, jkutzk2 @ uic.edu

Lecture (CRN: 40575) Tue-Thu, 2:00-3:15, LC C6	Lab (CRN: 40644) Wed 10-10:50am 2249E SEL Lab (CRN: 40645) Wed 11-11:50am 2249E SEL Lab (CRN: 40646) Wed 12-12:50pm 2249E SEL

The class is for students who have no (or very little) previous programming experience who are:

1. interested in seeking a major or minor in Computer Science, and/or
2. College of Engineering majors who wish to pursue the CS111-141 path to meeting their computing requirement, and/or
3. interested in biology in a computer science context, and/or
4. interested in computer science in a biological context.
5. the informal name for this biology-themed section is "CS+Bio" or "green"

Other students are more than welcome to join CS111 or may wish to check out the similar (but less intense) course, CS100.

The course will take its examples, the things that we write programs about, from biology, ranging from DNA and genes to population dynamics and evolution. The course is team taught by a Computer Science professor and a Biological Sciences professor, and will include substantial introductory biology content. The programming language will be Python.

**Course topics:** Introduction to programming: control structures; variables and data types; problem decomposition and procedural programming; input and output; aggregate data structures including arrays; programming exercises

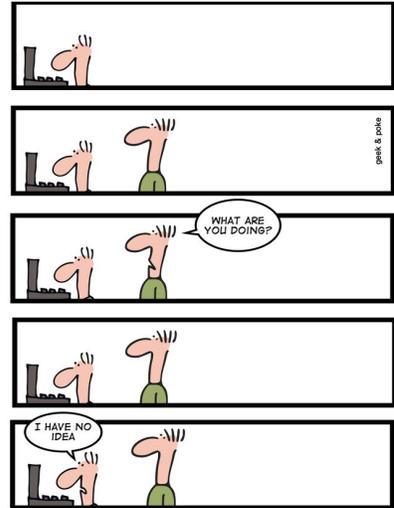
### OUTCOMES

By the end of this course you will:

1. Be able to read, understand, make functional alterations to, and create, through assembling very small code fragments, small Python programs (less than 100 lines) that achieve useful tasks.
2. Be able to design, implement, test, and debug (from scratch) a very small Python program that uses multiple standard control structures (e.g., conditional/if and iteration, or two iterations), which may be nested.
3. Gain a basic understanding of biology, which includes being able to:
  - a. Describe the fundamental organization of life.
  - b. Explain general functions at the molecular, cellular, and population levels.

## CS111: Program Design I (CS+Bio) Syllabus, Schedule, and Python Installation

- c. Apply programming skills to biological data
4. Appreciate what computer scientists do and the key concerns of that field that relate to students' professional lives:
  - a. Students will recognize that all digital data is an encoding or representation, and that the encoding is itself a choice.
  - b. Students will understand that all algorithms consist of manipulating data, iteration (looping), and making choices at the lowest level, about numbers, but we can encode more meaningful data in terms of those numbers.
  - c. Students will appreciate the value of a programming vs. direct-manipulation interface approach to computer use and will be able to describe situations where the former is preferable to the latter.



### ELECTRONIC COMMUNICATION: [Blackboard](#)

Most materials, such as lecture slides, and updated syllabus and schedule, will be posted on [Blackboard](#). You are responsible for checking Blackboard regularly and making sure you are doing all the work. **You are responsible for knowing about all of the material distributed for this class, whether it is mentioned in the syllabus, posted on Blackboard, or announced during lecture.**

### COURSE MATERIALS

#### Required texts

We will use [zyBooks](#) for reading- and graded hands-on exercises:

1. Sign in or create an account at: <https://learn.zybooks.com>
2. Enter zyBook code: **UICCS111Spring2020**
3. Subscribe

We will use [OpenStax Concepts of Biology](#) for biology-themed content (free), and/or hand-out documents. If you prefer, you can download an OpenStax PDF version for free, or buy a copy on ibooks, Amazon, and other sources.

#### Optional texts

In addition to the required course materials, you may find the following sources of information may be beneficial and satisfy additional individual learning needs:

[Computing for Biologists: Python Programming and Principles](#) by Ran Libeskind-Hadas & Eliot Bush, Cambridge University Press. [ISBN-13: 978-1107642188].

## CS111: Program Design I (CS+Bio) Syllabus, Schedule, and Python Installation

*How to Think Like a Computer Scientist: Interactive Edition*, a free online interactive book at <http://interactivepython.org/runestone/static/thinkcspy/index.html>, based on an open source book by Jeffrey Elkner, Allen B. Downey, and Chris Meyers.

*Introducing Python: Modern Computing in Simple Packages* by Bill Lubanovic (UIC has 4 electronic copies available through Safari). [ISBN-13: 978-1449359362]

*Python Programming: An Introduction to Computer Science* (3<sup>rd</sup> edition) by John M. Zelle.

### COURSE REQUIREMENTS (GRADING)<sup>†</sup>

Lab assignments	20%
Lecture participation, attendance (from iClicker), and Zybooks activities	15%
Projects	25%
Two midterms	(10% * 2) 20%
Final exam	20%
<b>Total:</b>	<b>100%</b>

<sup>†</sup>Subject to change

**Note:** you must pass both the programming part of the course (labs plus programming projects) and the exam part of the course (midterms plus final) in order to receive a passing grade.

### COURSE POLICIES

#### Attendance and Participation Policy

You will be expected to attend all lectures, having completed the assigned reading and ready to discuss them with your classmates. Your lecture participation and attendance grade will come from answering clicker questions. You will receive 4 points for answering questions in each class plus 1 point for each correct answer. You will be expected to earn 4 points per class (if you miss a class, you can use correct answers to make up for the missed score). However, the total credit for attendance and participation is capped at 100% (no extra credit). We will excuse absences from the first three classes (so you can shop the other sections).

#### Policy for Missed or Late Work

All programming assignments (labs and projects) are to be turned in electronically via Blackboard. Lab assignments will be due at 11:59 pm Friday; projects will come with a stated deadline.

You have a total of **three day-late passes** that you can use on the programming assignments (labs and projects). So, for example, you can turn in three lab assignments one day late (3x1) without penalty, or one project three days late (1x3), or one project and two assignments one day late each (etc.), for a grand total of three days. Any late assignment is at least one day late (one hour late to 23.99 hours late). **Once you have used your late days, late assignments will not be accepted.**

#### Academic Integrity Policy – Honor Code

## CS111: Program Design I (CS+Bio) Syllabus, Schedule, and Python Installation

As an academic community, UIC is committed to providing an environment in which research, learning, and scholarship can flourish and in which all endeavors are guided by academic and professional integrity. All members of the campus community—students, staff, faculty, and administrators—share the responsibility of insuring that these standards are upheld so that such an environment exists. Instances of academic misconduct by students will be handled pursuant to the [Student Disciplinary Policy](#).

Consulting with your classmates on assignments is encouraged, except where noted. However, submissions are individual, and copying code from your classmates is considered plagiarism. For example, give the question “how did you do X?”, a great response would be “I used function Y, with W as the second argument. I tried Z first, but it doesn't work”. An inappropriate response would be “here is my code, look for yourself”.

To avoid suspicion of plagiarism, you must specify your sources together with all submitted materials. List classmates you discussed your assignment with and webpages from which you got inspiration or copied (short) code snippets. All students are expected to understand and be able to explain their submitted materials. Plagiarism and cheating, as in copying the work of others, paying others to do your work, etc, is obviously prohibited, is grounds for possibly failing the course, and will be reported. We will be running an automated plagiarism detection tool on your hand-ins.

***Students are advised that it is a violation to copy, or allow another to copy, all or part of an exam or program. We will enforce the outlined policies.***

### **Student Disabilities**

If you have a disability that might impact your performance in this course, or requires special accommodation, please contact instructors as soon as possible so that appropriate arrangements can be made. Support is available through the [Disability Resource Center](#). You will need to contact them to get your disability documented before accommodations can be made.

### **Incomplete Grade Policy**

The *UIC Undergraduate catalog* states that in addition to needing excellent justification for an incomplete grade, a student must also have been “making satisfactory progress in the course”. Therefore, *no matter how good your excuse*, we will not grant you an incomplete grade if you have less than a C average at the time you ask for an incomplete.

## CS111: Program Design I (CS+Bio) Syllabus, Schedule, and Python Installation

### CALENDAR OF SELECTED MAJOR COURSE EVENTS AND DEADLINES

Date	Event
Monday, Jan. 13	UIC semester begins
Tuesday, Jan. 14	First lecture
Wednesday, Jan. 15	Register your clicker and do assigned reading <b>before</b> 2 <sup>nd</sup> meeting of class
Wednesday, Jan. 22	First Lab meetings
Friday, Jan. 24	Last day to complete late registration; last day to add a course(s) or make section changes; last day to drop individual courses via Student Self-Service without receiving W (Withdrawn) grade on academic record.
Thursday, Feb. 13 (tentative)	First hour exam
Friday, March. 20	Last day for undergraduate students to use optional late drop in college office and receive grade of W on academic record.
March 23-27	Spring vacation
Thursday, Apr. 2 (tentative)	Second hour exam
Friday, May 1	Instruction ends
Wednesday, May 6, 3:30-5:30	Final exam

**Acknowledgment:** This material is based upon work supported by the [National Science Foundation](#) under Division of Undergraduate Education Grant, NSF-DUE-1612113. Any opinions, findings, conclusions or recommendations expressed in this material are those of the professors and do not necessarily reflect the views of the National Science Foundation.

### COURSE SCHEDULE v1.0

Week	Date	Lecture Topic	Notes & Assignments	
			Reading	HW/Lab lecture notes (PDF)
1	1/14	Why computer science & biology?		
	1/15	<i>No Lab This Week</i>		
	1/16	Introduction to Computer Science <a href="#">Introduction to Biology, DNA</a>	Zybook Ch. 1.1–1.5, 1.7  <a href="#">CoB 1.1; 2.3 Nucleic acids; Ch. 9.1</a>	
2	1/21	Variables, strings, brief intro to functions	Zybook: end of Ch. 1, 2.1–2.2, 3.1, 6.1–6.2. (Do challenge activities only for 2.1–2.2)	
	1/22	<a href="#">Lab 1: Python basics, "Hello world"</a>		
	1/23	Strings, sequences & indices	Zybook 2.3-2.4, 2.6	
3	1/28	Slicing & lists, <a href="#">DNA</a>	Zybook: 7.1; 3.2; go back and do 6.1 Challenge activities (they're short and easy)	
	1/29	<a href="#">Lab 2: Strings, variables, and functions</a>		
	1/30	<a href="#">DNA-RNA-proteins ('central dogma')</a> & Control flow: branching	Zybook, 4.1-4.3, 6.3; <a href="#">CoB 2.3 (proteins), 9.2- 9.4</a>	

## CS111: Program Design I (CS+Bio) Syllabus, Schedule, and Python Installation

		(if, else) and return values from functions		
4	2/4	Booleans, conditional statements	Review <a href="#">Zybook</a> 4.1-4.3;	
	2/5	<a href="#">Lab 3: Branching and functions: DNA and codons</a>		
	2/6	<a href="#">Genes</a> & more on branching, functions, and first peek at loops	Zybook 4.4-4.6 (in 4.5 skip any question mentioning dictionaries and the challenge activity); 5.1, 5.5 (do complete challenge 5.5.1); 6.5 <a href="#">CoB</a> 6.1	
5	2/11	Exam review		<a href="#">Sample exam</a> <a href="#">Exam review</a>
	2/12	<a href="#">Lab 4</a>	<a href="#">Review CoB 9.2- 9.4</a>	
	2/13	<b>Exam</b>		
6	2/18	Loops, more on parameter passing	If you have time: <a href="#">Zybook</a> 5.5 (review), 5.6, 6.6	
	2/19	<a href="#">Lab 5</a>		<a href="#">Exam solutions</a>
	2/20	Loops	<a href="#">Zybook</a> 5.6, 6.6, 5.2-5.4, 12.1, 12.2.	
7	2/25	Loops, with a focus on <i>while</i> ; <a href="#">Molecular comparisons-- context for Project 1</a>	<a href="#">CoB</a> Sections 3.2, 13.1 <a href="#">Zybooks</a> : review <i>while</i> loops 5.2-5.4; and 3.6, 8.1-8.3. Skip challenge activity 8.3.1.	
	2/26	<a href="#">Lab 6</a>		<a href="#">file I/O</a>
	2/27	notion of list and string methods, dot notation: methods of classes, <a href="#">BLAST, FASTA files</a>	<a href="#">Zybooks</a> : 7.3, 8.2	
8	3/3	CSV files, Modules, random numbers, plotting. & <a href="#">Traits</a>	<a href="#">Zybooks</a> : 2.10, 11.1-11.2, 11.7, 12.6-12.7, 13.1-13.2	
	3/4	<a href="#">Lab 7- no new assignment (Project 1 due 3/6)</a>		
	3/5	Nested Loops; list methods & <a href="#">Heredity</a>	<a href="#">Zybooks</a> : 5.8, 8.2, 8.4, 8.9 <a href="#">CoB</a> 6.1 and 8.1	
9	3/10	Incremental development, scope	<a href="#">Zybooks</a> : 5.9-5.10	
	3/11	<a href="#">Lab 8 - random numbers, plotting</a>		<a href="#">csv, matplotlib</a>
	3/12	More about developing functions. & <a href="#">Inheritance, Mendelian genetics</a>	<a href="#">Zybooks</a> : 6.4, 6.7, 6.9-6.11 <a href="#">CoB</a> : 8.1, 8.2	
10	3/17	Nested loops with for and while, for vs. while loops. & <a href="#">evolution</a>	<a href="#">Zybooks</a> : 5.7 <a href="#">CoB</a> : 11.1 & 11.2	
	3/18	<a href="#">Lab 9: Lists of lists, nested, list methods</a>		

## CS111: Program Design I (CS+Bio) Syllabus, Schedule, and Python Installation

	3/19	String methods, nested lists, if time, first look at dictionaries	<a href="#">Zybooks</a> : 7.3, 8.5.	
	<b>3/23</b>	<b>Spring Break</b>		
11	3/31	Review for Exam 2		<a href="#">Exam review</a>
	4/1	Lab 10		
	4/2	<b>Exam 2</b>	<b>no new reading; review</b>	
12	4/7	First look at dictionaries	<a href="#">Zybooks</a> : 7.3, 8.5, 8.12	
	4/8	<a href="#">Lab 11: going over exam solutions</a>		
	4/9	Dictionaries	<a href="#">Zybooks</a> : 8.12–8.13 <a href="#">CoB</a> : Ch. 12	
13	4/14	Trees of life		
	4/15	<a href="#">Lab 12: Dictionaries, string methods, and the fundamental dogma</a>		
	4/16			
14	4/21	String parsing	<a href="#">Zybooks</a> : 8.12, 8.13, 8.14 (if you haven't already read them)	
	4/22	<a href="#">Lab 13</a>		
	4/23	Speed (including more on dictionaries)	<a href="#">Zybooks</a> : 7.4	
15	4/28	Review Biology		
	4/29	<a href="#">Lab 14</a>		
	4/30	Review CS		
16	5/6	<b>FINAL EXAM</b> <b>Wednesday, May 6, 3:30-5:30pm</b>	<b>C6</b>	

### Installing Python and Spyder on your machine

*(Note: These are instructions for your home computer/laptop, not for the UIC lab computers! Python is already installed on lab computers.)*

Our class will use *Python 3*, and more specifically, the Anaconda distribution of Python 3.7. It comes with a nice editor that also helps us run and debug programs (a so-called “IDE,” or integrated development environment), which we will also use for *writing* Python programs for homework, projects, and practice.

To install these items on your computer, please follow the instructions below:

- Go to <https://www.anaconda.com/distribution/#download-section>
- Click the download button for the *Python 3.7 version*. You should only see the version for the auto-detected operating system you are currently using. *(Note: if given an option for operating system, choose yours—Mac/Windows/Linux. If you are given the option to choose between the Graphical Installer and the Command-Line installer, choose the Graphical Installer.)*
- Open the downloaded file, and click through Agreements (Agree/Continue) to perform the installation. *(Note: if you see a checkbox in the installation dialog asking something*

## CS111: Program Design I (CS+Bio) Syllabus, Schedule, and Python Installation

*like, “Do you want to modify PATH so anaconda is included in your home path” be sure you do check that box. This box is automatically checked and the question is ordinarily not asked in the course of a default install.)*

Congratulations, you're done! *Spyder* is automatically installed as part of the Anaconda distribution of Python. *Spyder* (Scientific Python Development Environment) is an integrated development environment useful for writing, running, and debugging code.

### **Starting and Using *Spyder* (editing code and running Python)**

(*Note:* When you open *Spyder* the first time, it will ask you to update to the latest version, currently v4.0.1. You can simply uncheck the box to stop automatic *Spyder* version updates.)

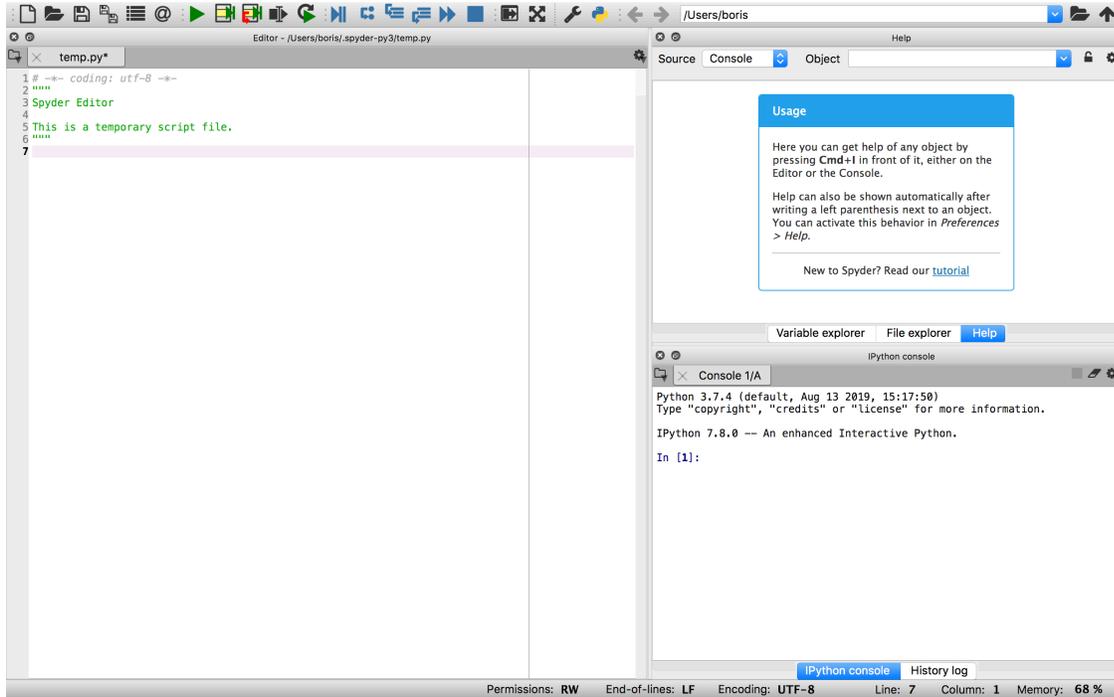
To open *Spyder*:

1. (On Windows OS:) Select Anaconda from the Start menu. (On Apple OS:) search for Anaconda Navigator with Spotlight (the magnifying glass on the top-right side of screen).
2. Having located either the Anaconda Navigator application or shortcut icon, click on either one to open the Anaconda Navigator. Finally, click on “Launch” under “*Spyder*” to launch *Spyder*.

Once you open *Spyder*, it will present with a large window, which is split into three smaller window panes, one on the left, and two on the right.

*Spyder* may take a little while to start up. (Depends on your computer.) It will look something like this:

# CS111: Program Design I (CS+Bio) Syllabus, Schedule, and Python Installation



1. Code Editor, on the left, lets you create and edit existing Python source files. It can contain several file tabs.
2. IPython console window pane, on the lower right, gives you direct access to a Python interactive console (also called the *Python interpreter* or *Python shell*). You can click on a different tab at the bottom to toggle between the console and history log.
3. Sometimes, you may find the top-right side pane useful. By default, it can display help, variables, and files. You will maybe use those later.