

# Course Syllabus

## Course Title

Foundations of Information Processing

## Semester

Summer 2022

## Course Number

IS430 – OBG/Obu

## Instructor

Kevin Trainor

## Teaching Assistant

Chenyue Jiao

## Distinction Between 4-Credit and 2-Credit Enrollment

I recommend that you enroll in this course for 4 credits. If you enroll for 4 credits, you will be expected to submit all coding assignments (see below) as well as the Final Project (see below). I believe that this will lead to the best learning experience.

Nevertheless, you may choose to enroll for 2 credits. If you enroll in this course for 2 credits, you will be expected to submit only the coding assignments. Even with this adjustment in workload, the work expected of students who enroll in this course for 2 credits is far greater than 50% of the work expected of students who enroll in this course for 4 credits. Consequently, I only recommend the 2-credit option to students who need to conserve their credits or their funds.

## Regular Class Sessions

Regular class sessions will be held on Wednesdays from 9:30 AM till 11:30 AM using Zoom. I look forward to joining you there. Please use a headset.

## Optional Lab Sessions

I hold optional lab sessions two times per week. Please join me to ask a question, to discuss solutions to previous assignments, to get help with the current assignment, to discuss the final project, or just to say hello. I will work with students on a first come, first served basis.

The first optional lab session of the week takes place immediately following our regular class session. Since class sessions typically end earlier than their scheduled two-hours, I devote the remainder of the time available to helping students. Often, I can stay beyond the two-hour period if many students need help.

On Monday afternoons from 4:00 PM till 5:00 PM, I will be holding an optional online lab session using Zoom. This represents a good opportunity to get help with your coding assignment before the submission deadline on Tuesday night. Please use a headset when joining the online lab session. If you have a schedule conflict that prevents you from attending this session, please contact me regarding attending the lab session for my other class. That lab session meets on Monday evenings.

## Office Hours

I do not hold conventional open office hours. Students needing help with assignments are encouraged to join in the optional lab sessions (see above). Students who need to discuss confidential matters can arrange for an individual meeting with me on Zoom. If you need to arrange an individual meeting with me, please send me an *Individual Meeting Request* via the Service Desk (see *Contacting Instructor or TA* below).

## Contacting Instructor or TA

The preferred method for contacting me or Chenyue is by entering a request using the [Service Desk for this course](#). PLEASE, DO NOT send requests to our regular email addresses.

The Service Desk for this course has been implemented using the JIRA Service Management product. If you are new to using the service desk in one of my courses, please visit the [Service Desk Introduction](#) for instructions and tips.

On an emergency basis, you may contact me using my mobile phone number: 847-650-9706.

## Catalog Description

Covers common data, document processing, and programming constructs and concepts. Focuses on problem solving and abstraction with a programming language. By the end of the course students will be able to design, develop and test a moderately complex computer program to manage full text, bibliographic records or multimedia. The course prepares students for working with applications in data analytics, data science, digital libraries, text mining and knowledge management. No prior programming background is assumed.

## Course Description

This is a first course in computer programming using the Python programming language. The overall goal of this course is to develop an information processing perspective of the world. More specifically, the objectives are:

1. To gain a broad introduction to computer programming concepts, principles, styles and techniques through lectures and discussion.
2. To develop a working proficiency in a programming language through lab sessions and hands-on assignments.
3. To nurture the ability to detect opportunities to apply programming in new scenarios.

## Course Topics

- Computers and Programs
- Writing Simple Programs
- Computing with Numbers
- Sequences: Strings, Lists, and Files
- Defining Functions
- Decision Structures
- Loop Structures and Booleans
- Simulation and Design
- Defining Classes
- Data Collections
- Algorithm Design
- Creating Custom Classes
- Using Jupyter Notebooks

## Learning Outcomes

After completing this course, you should be able to:

- Understand the role that programming plays in information processing and management.
- Identify opportunities for the information worker to solve every day work problems using programming.
- Design, code, and test procedural programs in Python that make use of object-oriented classes provided by others.
- Design, code, and test simple object-oriented classes in Python.
- Use the PyCharm integrated development environment competently.

## Required Texts

Zelle, J. (2016). Python Programming: An Introduction to Computer Science, 3rd Ed  
ISBN: 9781590282755

Severance, C. (n.d.). Python for Everybody ( <https://www.py4e.com/book> ).

*This text is available as an electronic book in several formats. Although it is also available as a printed book, I DO NOT RECOMMEND PURCHASING a printed copy. I invite you to take advantage of this author's generosity by using one of these electronic formats.*

## Technology Requirements

You will be completing coding assignments and conducting your final project using your own computer. I recommend that you use a computer that runs Windows 10 or a recent release of MacOS. While the software that we will be using for this course does run on Linux, there will be substantially less technical support available for Linux. If you want to use a Linux computer for your coursework, please contact me first.

You will be installing the following software on your computers for this class:

- The Anaconda open data science platform for Python
- The PyCharm integrated development environment (IDE) for Python

All of the software and services that you will be using are free for your use during this class. I will publish brief instructions and detailed tutorial videos to help you download, install, and begin using the required software on computers running Windows 10 and recent releases of MacOS. Links to these resources will be provided in our Weekly Schedule.

The tools described above are those that I recommend for your use when completing work for this course. If you are not able to use one of these tools because of accessibility reasons, please contact me to get approval for a suitable alternative that meets your needs as a student and our needs as graders.

### **Course Schedule**

The schedule for this course will be available via our Weekly Schedule at:

[https://courseinfo.ligent.net/2022su/illinois/is430\\_obg\\_obu/index.html](https://courseinfo.ligent.net/2022su/illinois/is430_obg_obu/index.html)

The course schedule is always subject to reasonable change to account for changes in circumstances and to correct errors. When I make changes to the schedule, I will announce them via our Canvas Announcements Forum. Postings to this forum should result in you being sent an email copy of the announcement as well.

### **Course Elements**

#### *1. Readings*

Required readings will be assigned from the textbooks listed above and from other resources that will be identified in the Weekly Schedule. Generally, readings are chosen to accompany any lecture videos or tutorial videos provided for the week. So, you should expect to complete the readings before playing the videos.

This course has two required textbooks (Zelle and Severance). Most of the required reading assignments are from the Zelle text. A few required reading assignments are from the Severance text.

In most weeks, optional readings are identified in the Weekly Schedule. Often, these are presentations of the same material from the other text book. I have found that reading a second approach to the same subject is particularly helpful for first time programmers and for anyone having difficulty with a particular topic. I expect you to use your best judgement in deciding whether the optional readings are adding appropriate value for you.

#### *2. Lecture Videos*

I typically will not be using our online class time for lectures. Instead, I have provided links to pre-recorded video lectures for each unit that we cover. While most of the material covered in the recorded lectures is from the textbook, I occasionally cover supplemental material in the lectures as well. I always include commentary that I believe adds value to the text.

While I have marked the lecture recordings for this course as *required*, I am leaving the decision to you regarding which combination of reading and playing lecture

recordings will be the best investment for you. While many graduate-level courses do not include lectures on the material from the text, some graduate-level learners prefer learning by listening. I invite you to sample all of the content and then settle into a usage pattern that works best for you.

Before playing my lecture videos, make sure that you have previously played [Tips on Playing My YouTube Videos](#) to assure that you get the most from your viewing experience.

### 3. *Tutorial Videos*

I have created tutorial videos for the following use cases:

- There are a number of tutorials that explain how to do activities necessary for the course. These include installing software, reading grading rubrics, submitting assignments, and related activities.
- Many tutorials are coordinated with the exercises in your coding assignments. They represent a demonstration of the skill you will be expected to use when doing the exercise.

Because tutorial videos do not duplicate material covered by other resources, I recommend that you always play these recordings. Before playing my tutorial videos, make sure that you have previously played [Tips on Playing My YouTube Videos](#) to assure that you get the most from your viewing experience.

### 4. *Coding Assignments*

There will be weekly Coding Assignments. As mentioned above, coding assignments will often be paired with tutorial videos. These should allow you to complete your Coding Assignment using the same general approach that has been demonstrated in the tutorial video.

A Canvas submission activity will be provided for submitting each assignment. Instructions for each Coding Assignment and a grading rubric will be published in the Weekly Schedule.

Solutions to the Regular Exercises from the Coding Assignments will be posted to our Canvas site immediately before our next online class session. You can expect us to review your solutions and mine at the beginning of our next class.

A major goal for this course is to build your proficiency in self-evaluation of your work. To build this skill, I will expect you to be able to estimate your grade on each Coding Assignment. The solutions to coding assignments posted to our Canvas site and our review of those solutions during the next class will serve as your primary feedback for the Coding Assignment.

As secondary feedback, your Coding Assignment submissions will be graded and commented upon. This feedback will be published to the Canvas assignment submission activity within 2 weeks.

For information regarding the grading of coding assignments, please see *Coding Assignment Submissions That Meet Certain Criteria Are Subject to a Minimum Score Guarantee* under *Grading Policies* below.

5. *Final Project*

The Final Project will be a significant multi-part assignment to be completed in the final weeks of the semester. This assignment will require you to design, code, and test an application solution consisting of multiple programs and using techniques covered throughout the course.

The Final Project is an individual assignment, NOT a group assignment. More detailed instructions and a grading rubric for this assignment will be published separately.

6. *Attendance*

The iSchool expects students to attend all classes except in cases of emergency. See *Student Code on Attendance*: <http://studentcode.illinois.edu/article1/part5/1-501/>. Students who miss class are expected to play the recording of the class. If you need help locating the recording of a class, please contact the iSchool Help Desk.

7. *Participation*

Your participation in the course is an important element of the course. Accordingly, a significant portion of your grade for this course will be determined by your participation. Students will earn participation credit for:

- Making a *Greetings* post to the Service Desk during the first 2 weeks of the course.
- Making speaking contributions during class.
- Making chat contributions during class.
- Presenting your solution to coding assignments during class.
- Presenting as the spokesperson for breakout groups during class.

For information regarding the grading of participation, please see *Your Participation Grade Will Be Based Upon Participation Points Earned Throughout the Semester* under *Grading Policies* below.

## Course Grading Policies

1. *iSchool and University Grading Policies Apply*  
Many iSchool and University Academic Policies have grade implications. Please see *iSchool and University Academic Policies* below.

2. *Careful Attention to Detail is Required*  
One important goal of this course is to train you to become a responsible information professional. The work of information professionals is highly detail oriented. Clients rely on information professionals to deliver a correct work product that conforms to stated requirements and best practices.

When your work is graded, deductions will be made for all deviations from the assignment instructions. Some of these deductions will be made for small deviations that may seem insignificant to you. So, it is a good practice to carefully check your work against all instructions before submitting.

3. *Assignment Resubmissions are Not Permitted After the Assignment Deadline*  
While assignments may be resubmitted before the assignment deadline, they may not be resubmitted after the assignment deadline. If you should accidentally make a submission that is somehow defective, you must discover and correct this error before the assignment deadline. Resubmission of assignments after the deadline will not be permitted for any reason.

4. *Deadline Extensions Must be Requested Before the Deadline*  
If you believe that you have a valid reason for a deadline extension, please submit a *Deadline Extension Request* using the Service Desk before the deadline. I have a practice of granting reasonable extension requests. I will only grant extensions beyond the beginning of our next class session in very limited circumstances.

5. *Deductions Will be Made for Late Submissions*  
The grading rubrics for coding assignments and for the Final Project include substantial deductions for late submissions. Please see the assignment grading rubrics for more details.

6. *Assignments Submitted Too Late Will Not be Graded*  
Coding assignments submitted more than 14 days late will be considered *too late*. Final Projects submitted more than 7 days late will be considered *too late*. Assignments that are submitted *too late* will not be graded. These submissions will earn a grade of zero. If you are in danger of missing the *too late* deadline, and you believe that you have a valid reason for an extension, please submit a *Deadline Extension Request* using the Service Desk before the deadline.

7. *Grade Adjustments Will Be Limited to Automatic Rounding*  
All grades will be awarded on 0 to 100-point scale. Fractional values will be rounded automatically. Fractional portions of grades ending in .0 through .4 will be rounded down. Fractional portions of grades ending in .5 through .9 will be rounded up.

No further adjustments will be made to grades. This policy applies even in situations where increasing a grade by just 1 point would cause a student's final letter grade for



the course to cross a threshold (i.e. from B+ to A-). Regardless of the potential consequences, grade adjustments will be limited to automatic rounding.

8. *Re-Grading Requests Made Using the Service Desk Will be Given Fair Consideration*

It is possible for one of your assignment submissions to be missed during the grading process. This is especially true for assignments that are submitted late. If this happens to you, please submit a Re-Grading Request to the Service Desk to remind us that your submission still needs grading. Make sure to fully identify the assignment that needs attention.

Each assignment that we grade is accompanied by a grading feedback form. Please read this feedback to understand our grading decisions. If, after reading the grading feedback form, you believe that our grading decisions are somehow unfair, please submit a Re-Grading Request to the Service Desk. Include details in your request that identify the assignment and your rationale for the re-grading request. We will give these requests fair consideration and inform you of our determination by posting back to the Service Desk ticket.

9. *Extra Credit Opportunities are Not Available*

I have designed the grading policies for coding assignments such that poor performance on one coding assignment should not spoil your entire semester grade (see Item 11, below). Consequently, I do not offer any opportunities to submit work for extra credit.

10. *The Expected Grade for Assignment Submissions That Meet All Expectations is 95*  
The grading rubrics for all assignments have been designed such that submissions that meet all expectations for the *Regular Exercises* will score 95 points.

Assignments will also include a *Challenge Exercise*. Students who choose to submit the Challenge Exercise may earn from 0 to 5 additional points. Challenge Exercises are designed to be quite difficult and the amount of help available from the instructor for Challenge Exercises will be limited to hints.

11. *Coding Assignment Submissions that Meet Certain Criteria are Subject to a Minimum Score Guarantee*

The grading rubric for Coding Assignments has been designed to promote two important behaviors:

- Submitting your work in a properly named and formatted file. This helps substantially with grading workflow.
- Submitting your work by the assignment deadline. This assures that you will get the benefit of having tried to solve the problem on your own before seeing the solutions of others.

While separate grading rubric and assignment submission instructions documents will be published, the following is a summary of the coding assignment grading rubric features:



- 10 points will be awarded for submitting a single, properly named and properly formatted file to the proper Canvas assignment submission activity.
- A minimum of 75 points will be awarded for submissions that are submitted on time, and that demonstrate a good faith effort on all Regular Exercises. Late submissions will be awarded 74 points or fewer in this category.

The implication of this grading scheme is that you can expect a score of 85 or higher on all coding assignment submissions that meet both criteria.

*12. Your Participation Grade Will Be Based Upon Participation Points Earned Throughout the Semester*

The table below lists activities for which you may earn participation points and the points earned for each instance.

<b>Activity</b>	<b>Points Earned</b>
A <i>greetings</i> post made to the Service Desk for this course by the end of Week 2	10
Speaking contributions during class	2
Chat contributions during class	1
Presentations of your Coding Assignment solution during class	5
Presentations as spokesperson for your a breakout group during class	5

Your participation grade for the course will be calculated at the end of the semester based upon the total number of participation points earned. Participation grades will be calculated using a curve. The grading curve calculation is summarized in the

following table. Please note that students earning fewer than 10 participation points will earn a participation grade of zero.

<b>Participation Points Percentile Rank</b>	<b>Participation Points</b>	<b>Participation Grade</b>
99	NA	100
75	NA	95
50	NA	90
25	NA	80
10	NA	70
< 10	$\geq 10$	60
< 10	< 10	0

*13. Attendance at Class Sessions May Affect Your Grade*

While attendance is not graded directly, it may have a significant impact on your participation grade. Nearly all activities that earn participation credit occur during class.

## Academic Integrity Policies and Practices Specific to This Course

There are iSchool and University policies regarding academic integrity. Please see the *iSchool Academic Integrity Statement* under *iSchool and University Academic Policies* (below).

Designing, coding, and testing your own programs is the primary learning activity in this course. Most of the learning value of these activities is the result of doing them on your own. Coding your programs independently leads to expected differences between the program files submitted by different students. These differences are not only expected, but they are often celebrated, as we explore the boundaries of good programming style and good programming practice.

The program files that you submit as your own work in this course will be compared to those submitted by other students in your section and by students in other sections. Your submitted work will also be compared to the official class solution. Instances of high correlation between the work that you submit, and the work of others (including the official class solution) will be investigated as possible instances of plagiarism or cheating. Both plagiarism and cheating are serious academic integrity infractions. See <https://studentcode.illinois.edu/article1/part4/1-402/>.

The submission of highly correlated work represents an academic integrity violation for students who copy the work of others, for students whose work is copied by others, and for students who create and submit a collaborative work when independent work is expected. So, please do your work independently and do not put yourself in a position where your work could be submitted by another student.

Our work in checking your submissions for academic integrity violations will continue throughout the course. File comparison is a time-consuming activity and late submissions create new opportunities to discover highly correlated work. So, investigations can be initiated at any time during the course – even after an assignment has been graded.

The only exceptions to this general prohibition on submission of highly correlated work come from two situations where I give you code:

### 1. *Starter Files*

As part of coding assignments, I sometimes give you starter files. Typically, these are test data files. But, sometimes, these are programs that I expect you to include in your project before submission. You are permitted to submit any programs that I provide to you as starter files without further modification.

### 2. *Tutorial Examples*

During tutorial videos, I create programs with the expectation that you may follow along. Often, these tutorial programs can serve as a great starting point for the program that you are expected to submit for the exercise. You are permitted to copy and adapt code from my tutorial programs provided that you type the code into your computer yourself.

### Basis for Determining Overall Course Grade

The various components of student work will contribute to the overall course grade based upon the following percentages:

Category	4-Credit Students	2-Credit Students
Participation	10%	10%
Coding Assignments	45%	90%
Final Project	45%	NA

Letter grades will be determined as follows:

- A+: 97 - 100%;
- A : 93 - 96%;
- A- : 90 - 92%;
- B+: 87 - 89%;
- B : 83 - 86%;
- B- : 80 - 82%;
- C+: 77 - 79%;
- C : 73 - 76%;
- C- : 70 - 72%;
- D+: 67 - 69%;
- D : 63 - 66%;
- D- : 60 - 62%;
- F : 0 - 59%;

Please note that when converting overall course number grades to letter grades, simple rounding of number grades will be used. Please see *Grade Adjustments Will Be Limited to Automatic Rounding* under *Grading Policies* above.

## ISCHOOL AND UNIVERSITY ACADEMIC POLICIES

### Incomplete Grades

An exceptional request for an incomplete grade is most often granted to students encountering a medical emergency or other extraordinary circumstances beyond their control. Students must request an incomplete grade from the instructor. The instructor and student will agree on a due date for completion of coursework. The student must submit an Incomplete Form signed by the student, the instructor, and the student's academic advisor to the front office:

<https://uofi.app.box.com/s/sx7arobhr0gfw12teaetmp1qq32ifdrd>

Please see the Student Code for full details:

<http://studentcode.illinois.edu/article3/part1/3-104/>

### iSchool Academic Integrity Statement

The iSchool has the responsibility for maintaining academic integrity so as to protect the quality of education and research in our school and to protect those who depend on our integrity. Consequences of academic integrity infractions may be serious, ranging from a written warning to a failing grade for the course or dismissal from the University.

See the student code for academic integrity requirements:

<http://studentcode.illinois.edu/article1/part4/1-401/>

### Statement of Inclusion

<https://diversity.illinois.edu/about/senate-diversity-resolution/>

As the state's premier public university, the University of Illinois at Urbana-Champaign's core mission is to serve the interests of the diverse people of the state of Illinois and beyond. The institution thus values inclusion and a pluralistic learning and research environment, one which we respect the varied perspectives and lived experiences of a diverse community and global workforce. We support diversity of worldviews, histories, and cultural knowledge across a range of social groups including race, ethnicity, gender identity, sexual orientation, abilities, economic class, religion, and their intersections.

### Religious Observances

In keeping with our Statement of Inclusion and Illinois law, the University is required to reasonably accommodate its students' religious beliefs, observances, and practices in regard to admissions, class attendance, and the scheduling of examinations and work requirements.

If you anticipate the need for an accommodation, please communicate with your instructor in the first two weeks of class. If you are an undergraduate student and your instructor requires an absence letter, you must fill out the Religious Observance Accommodation Request form:

[https://cm.maxient.com/reportingform.php?UnivofIllinois&layout\\_id=19](https://cm.maxient.com/reportingform.php?UnivofIllinois&layout_id=19) . Other accommodations may be available.

### Accessibility Statement

To obtain accessibility-related academic adjustments and/or auxiliary aids, students with disabilities must contact the course instructor and the Disability Resources and Educational Services (DRES) as soon as possible. To contact DRES you may visit 1207 S. Oak St., Champaign, call 333-4603 (V/TTY), or e-mail a message to [disability@uiuc.edu](mailto:disability@uiuc.edu) .

**iSchool COVID-19 Statement**

In keeping with university and iSchool policy, all students are required to engage in appropriate behavior to protect the health and safety of our community. If you are on campus, this includes being fully vaccinated, wearing a facial covering properly when required, maintaining social distance, if requested, and using hand sanitizer as needed.

If you feel ill or are unable to come to class or complete class assignments due to issues related to COVID-19, including but not limited to testing positive yourself, feeling ill, caring for a family member with COVID-19, or having unexpected child-care obligations, you should contact the instructor immediately and cc your advisor.

**Contact Hours**

This course will require approximately 54 contact hours.

**Last Revised**

2022-05-09