# IST256 Syllabus Fall 2023

## Course Information

IST256: Programming for the Information Sciences

### Description

Computational literacy and learning to code are critical skills of the 21st century. Students are introduced to the Python programing language with emphasis on practical applications that are relevant to our everyday lives and common within the information field.

### Audience: IST256 or IST356?

This course is for students who are new to computer programming yet desire to learn how it applies to our everyday lives.

* IST256 is for students with little to no programming experience. The course content is 75% python fundamentals and 25% python for data analytics.
* IST356 is for students with prior programming experience. The course content is 25% python fundamentals and 75% python for data analytics.

### Learning Outcomes

At the end of the course, students will be able to:

1. Analyze complex problems by thinking computationally and systematically.
2. Solve practical, real-world problems using a modern computer programming language..
3. Demonstrate the ability to read, write, discuss and code confidently.
4. Understand how to code in teams, collaborate with others and manage source code.
5. Acquire new programming knowledge independently.

### Large Group and Recitation Sections

Every student in IST256 is assigned to the main section M001, then one of the recitation sections. You are required to attend both sections every week. Your recitation instructor is responsible for your grades.

| SU Section | Class # | Type | Professor | Professor Email | Meeting Day/Time | Location / Instruction Mode |
| --- | --- | --- | --- | --- | --- | --- |
| M001 | 16592 | Main Section | Michael A Fudge Jr | mafudge@syr.edu | Mondays 3:45pm - 5:05pm | Watson Theater |
| M002 | 16600 | Recitation | Yuheun Kim | ykim72@syr.edu | Wednesdays 2:15pm - 3:35pm | 115B Lyman Hall |
| M003 | 20487 | Recitation | Angela Usha Ramnarine-Rieks | auramnar@syr.edu | Wednesdays 3:45pm - 5:05pm | 101 Huntington Hall |
| M004 | 16601 | Recitation | Daniel V Smith | dvsmith@syr.edu | Wednesdays 5:15pm - 6:35pm | 350 Newhouse II |
| M005 | 16602 | Recitation | Daniel V Smith | dvsmith@syr.edu | Wednesdays 3:45pm - 5:05pm | 340 Newhouse II |
| M006 | 20488 | Recitation | Deborah L Nosky | dlnosky@syr.edu | Wednesdays 3:45pm - 5:05pm | 243A Hinds Hall |

### Office Hours

Office hours are for asking questions, clearing up doubts and misunderstandings in the the coursework and getting advice / guidance on labs and homework. Please to not expect to be tutored during office hours, and please do not work on your homework during our office hours. Each of your instructor’s Office Hours will be posted in Blackboard. If you require tutoring, please see the **getting help** section below.

## Understanding Approach Used in this Course

Learning to program a computer does not come easy for most people. Decades of teaching programming to students like yourself has taught me it requires time, patience, practice and a well-established routine. This is not unlike the same routine required to learn a foreign language or musical instrument. There are times to practice and then times to demonstrate what you have learned.

### Spaced and Repetitive Practice

For better or worse, this course grading is designed to force you to practice. There are various activities due each week: readings, labs, and in-class / out-of-class homework activities. These are designed to expose you to programming little each day rather than binging the content once a week. Consuming the material this way gives you multiple points of exposure and most importantly time to process. Practice activities are formative assessments. This means being correct carries the same weight as explaining your struggles when you know you are not correct and seeking help when you need it.

### Building Habit Through Routine

Another thing we do to help you to be successful is to impose a routine upon you. The course material is conceptually difficult so we try to remove some of the logistics from learning by providing the same structure week to week:

1. Before the main section lecture you are exposed to the topic through **reading and video assignments**. The focus is on understanding the concepts.
2. During the **main section lecture** you observe and ask questions. You do not need write code at this point. The focus is on reading code, understanding the concepts as applied through code, learning how the code executes, and asking questions.
3. The **lab assignments** help you to practice writing code for the first time. This is done in a guided fashion. You are given short, specific problems to solve with code. For each lab, a guidance video is provided to those who require it. The completed lab is submitted as evidence that you practiced and include what you have learned.
4. During your **recitation**, we focus on problem solving and writing solutions with code. The problem at this phase are similar complexity as the homework. At this point every student should be writing code, learning to get comfortable writing code and troubleshooting problems. Code written in Recitation must be turned in for a participation grade.
5. Finally, the **homework** assignments are your opportunity to demonstrate you can code a solution to a problem on your own. Guidance is provided as to how to approach the problem. Homework assignments are a form of practice so it is expected students will explain what they learned or are still struggling to conceptualize. Expressing your thoughts on the learning experience is an import part of learning itself.

### Summative Assessments

Summative assessments are no longer about practice. They are about measuring what you have learned. There are two types of summative assessments in this course **exams** and a **project**. There are exams at milestones throughout the course. The project demonstrates you can learn to code independently, then explain your solution to another.

### Main Session Expectations (Mondays)

* The Monday’s Large group session, section M001 in a large lecture hall. Because it can be intimidating to ask questions in this forum, you are encouraged to use the class chat offered for this course. The first slide at the beginning of large group will explain how to access the class chat. You are welcome, and encouraged to ask questions or for clarification of concepts during the lecture.
* You will be expected to engage in class by participating in class-wide polls, quizzes and surveys. Your responses are not graded but they are recorded as a means to measure your participation and engagement in class. Students who do not participate are noted and will be flagged for poor participation in orange success.
* You are encouraged to not try to code as I code, especially if you are a beginner. Instead, watch and ask questions about what you see and experience. It’s far too early for most learners to try and code at this point, and there will be opportunities to do that in small group. Instead you should take nodes and ask questions. Except for the homework solution, the code I write will be made available to you after class.

### Recitation Expectations (Wednesdays)

* Recitation sessions meet with a fraction of the class. Here you will practice programming and algorithmic problem solving.
* You are expected to bring your fully-charged laptop to class! You will need to use it to complete in-class exercises. If you do not have your laptop, then you are not prepared for class.
* You will be expected to participate in class. This includes sharing your thoughts, ideas, and computer code when you are asked. Some of you might be hesitant to do this, but you need to get over it for your own benefit. Nobody starts out programming as an expert. It takes practice, experimentation, and repeated failure to get it.
* Please be respectful of your instructor and classmates. You are not competing against each other, you are a community. Not everyone learns at the same pace and we should be kind and respectful to our classmates who struggle.

## Course Resources

### Course Website

Our course website it located at <https://ist256.com> or [https://ist256.github.io](https://ist256.com). The course website contains the syllabus, list of due dates, and links to readings, content, videos and tools used in the course.

### Jupyterhub

Our programming environment is a private-cloud web application called Jupyter Hub. <https://v2hub.ischool.syr.edu>. This is the de-facto programming environment of the scientific community. All students have an account; use your SU Microsoft Account (NetID and password) to login. After you login you will see a **library** folder inside that folder is an **ist256** folder. All of the course content (lecture slides, code samples, labs, homework) is available in this folder.

### Textbooks

The following text is **required**:

* *Python for Everybody: Exploring Data In Python 3* by Charles Severance. <https://www.py4e.com/book>. The book is free and available in several formats from the URL provided.
* There are assigned readings which must be completed prior to each large group lecture. The reading can be found in the content section of the website and the course syllabus.
* In addition to the required reading the last 4 units have custom readings authored by your instructor.

Here are some **additional textbook recommendations**. Consider these supplemental resources:

* *Automate the Boring Stuff with Python: Practical Programming for Total Beginners* by Al Sweigert <https://automatetheboringstuff.com/>. Free online resource.
* *A Byte of Python*, <https://www.gitbook.com/book/swaroopch/byte-of-python/details>. Free online resource.
* *Dive into Python* , Mark Pilgrim <http://getpython3.com/diveintopython3/>. Free online resource.
* *Learn python the hard way*, Zed Shaw <http://learnpythonthehardway.org/book/>. Free online resource.
* *Python Practice Book*, Anad Chitpothu <http://anandology.com/python-practice-book/index.html>. Free online Resource.
* *Programming In Python 3*, Zybooks <https://www.zybooks.com/catalog/programming-in-python-3/>. Paid resource, approximately $60 for the semester.

### NetID, Google and Microsoft Accounts

This course will require you to use your Syracuse University provided Google and Microsoft Accounts. Both accounts are based on your NetId. Your Google account is netid@g.syr.edu and your Microsoft Account is netid@syr.edu. Learn more:

* NetID: <https://netid.syr.edu/>
* Microsoft Account: Login in with your SU email address and NetID password.
* SU Google Account: <https://ols.syr.edu/google-account-for-syracuse-university/>

### Bring Your Own Device

This course uses the BYOD (Bring Your Own Device) model.

* For Large Group you need a device to ask and answer questions, basically to participate in group chat and polls. This can be a smartphone, tablet or laptop (Mac, Windows, or Chromebook). Do not try to code along in large group unless you are experienced. It is best to take notes and ask questions.
* For Recitation you need a device for which you can write code in the browser using Jupyter Hub. This should be a laptop computer (Mac, Windows or Chromebook).
* Since you will code in a web browser using the provided JupyterHub platform, the hardware requirements are minimal.

### Tutoring

The University offers free group tutoring for this course through the Center for Learning and Student Success (CLASS).

Sign up for tutoring at: <http://myt.syr.edu>.

For more information on individual and group tutoring sessions, please visit: <https://class.syr.edu/academic-support/>

## Methods of Evaluation

### Grading

This course uses a well thought out mix of formative, summative, in-class and out-of-class instruments to assess your knowledge acquisition. A variety of techniques are used to cater to students of different learning styles and assess the course learning outcomes.

| AssessmentName | BlackboardGradebook | Type | Learning Outcomes | Quantity | PointsEach | PointsTotal | Pct OfTotal Grade | How Do I Turn it in? |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Exams | E1 - E2 | Assessment | 1,2,3 | 2 | 45 | 90 | 35% | Taken in Class |
| Project | P1 - P4 | Assessment | 1,2,3,5 | 1 | 49 | 49 | 20% | Project folder on Jupyterhub |
| Class Coding Labs | L01 - L13 | Practice | 1,2,3,4 | 13 | 3 | 39 | 15% | Run the submission script at end of the Lab in Jupyterhub |
| Homework | H01 - H13 | Practice | 1,2,3,4 | 13 | 3 | 39 | 15% | Run the submission script at end of the Homework in Jupyterhub |
| RecitationCode | S01 - S13 | Practice | 1,2,3 | 13 | 3 | 39 | 15% | Run the submission script at the end of recitation |
| TOTAL |  |  |  |  |  | 256 |  |  |

* For a comprehensive list due dates and times for all assignments, consult the [What’s Due?](#whats-due) section of the [Course Schedule](#course-schedule).

### Exams (E1 - E2)

* Exams are high-stakes, summative assessments. They measure the individual’s ability to recall, understand, and apply the course material. They are one of two instruments in this course which measures your mastery of the learning outcomes.
* There will be 2 exams in the course.
* Each exam focuses on specific lessons, but due to the nature of the course material, all exams are cumulative.
* Exams are issued in class.
* Exams are closed book.
* You will have 60 minutes to complete 45 exam questions.
* Students can bring a one page cheat sheet with typed or hand written notes and/or code samples.
* Questions consist of multiple choice, fill-in-the blank, and short-answer.
* If you require a special testing accommodation, you should schedule to take the exam in the CDS testing center.

### Project (P1 - P4)

* The project is the other high-stakes summative assessment. The goal of the project is to demonstrate your ability as an individual to program something novel, useful and innovative in Python. It should represent an accurate culmination of what you have learned in the course.
* You will work on the project individually, be expected to produce working code, and be able to explain it at both a high and detailed level.
* The project is divided into 4 phases; due dates are posted on the course schedule.
* You will receive feedback and advice after the first two deliverables; a project grade after the final deliverable.
* Each project phase must be submitted on Jupyterhub using the provided submission notebook.
* Late submissions are not accepted. We need time to grade.
* The [What’s Due](/#whats-due) section of the syllabus outlines the two exam dates.

#### Project Phases

| Phase | Name | Deliverables |
| --- | --- | --- |
| P1 | Ideation | Outline the specific goals and objectives of your project; include evidence of its feasibility by including citations of resources you will use to complete the code. |
| P2 | Beta Version | Create a first working draft of the completed program i.e. the “Beta” version; Provide live demo of running program to your Recitation instructor for feedback. |
| P3 | Final Version | Final version of working code; Instructor feedback taken into consideration; Improvements to achieve the desired grade. |
| P4 | Demo and Reflection | Pitch / Demo video of running project program. Video reflection; |

#### Criteria for Project Grade

* Complete all project deliverables on time, and to satisfaction as per the requirements.
* Clearly demonstrate through code 3 things you learned beyond what was taught in class. This can be a new API, module, aspect of or aspect of the Python language itself. This goes beyond just use, you must identify and explain.
* Take instructor feedback into consideration.
* Journal as your work on your project, recording time and tasks.
* In addition, there is a grade limit based on the number of lines of student-written code that is used in the project. Note: copies of code from class or elsewhere do not count. This must be code you wrote yourself that directly impacts the project’s behavior.

| Lines of student-written code in the project | Maximum Possible grade |
| --- | --- |
| 200 or more | A+ |
| 100 to 200 | B+ |
| Under 100 | C+ |

#### Grading Scale For Project

Finally, you are assigned a letter grade for your project. This letter grade is translated to a number of points based on this table.

| Project Grade | Assigned Points |
| --- | --- |
| A+ | 49 |
| A | 47 |
| A- | 45 |
| B+ | 42 |
| B | 40 |
| B- | 37 |
| C+ | 35 |
| C | 32 |
| C- | 30 |
| D+ | 27 |
| D | 25 |
| D- | 20 |
| F | 0 |

Specifics on the project as well as details of each deliverable can be found under your **project** folder in Jupyterhub.

### Class Coding Labs (L01 - L13)

* Each week there will be an out-of-class hands-on lab programming activity.
* The purpose of the lab is to provide guided, hands-on programming practice. Labs are your first opportunity to get your hands on a keyboard and start programming. This is a low-stakes assessment, formative assessment.
* The [What’s Due](/#whats-due) section of the syllabus identifies the lab you should complete.
* You can find the lab activity on JupyterHub.
* This activity must be completed and turned submitted by the due date.
* You may work alone or with a partner as you complete the lab. If you work with another, you should both complete the lab individually, and you should make a note of who your lab partner was when you completed your work. As to not draw attention to a potential academic integrity violation.
* If you are having difficulty completing the lab, you are welcome to review the lab walk-through video which guide you through the more difficult parts of the lab. You are encouraged to only consult the walk-through when you are stuck.

#### Rubric for Class Coding Labs

Lab Criteria Definitions:

1. **Code Correct** means all You Code sections of the lab are correct.
2. **Code Complete** means all You Code sections have an honest attempt to code the problem at hand. Please note this does not imply the code is correct. If the code is not correct, there is an adequate reflection with student questions.
3. **Cells Executed** means all code cells in the lab display evidence they were executed in your lab submission.
4. **Metacognition Complete** means the student made an honest effort to answer the open-ended questions in the lab adequately conveying what you have learned and what still confuses you. This should be evident in the work you have done to complete the lab. We value reflection. It is important to the learning process.

| Lab Criteria | Assigned Grade |
| --- | --- |
| All 4 criteria met | 3 |
| 3 criteria met | 2 |
| 2 criteria met | 1 |
| Less than 2 criteria met | 0 |

### Homework (H01 - H13)

* Practice makes perfect. Each week you will be assigned homework to complete outside of class.
* The goal of the homework is practice problem solving with code independently. Throughout the process you should take inventory of your abilities with respect to the material. While it is admirable to get the code correct, that is not the evaluation criteria nor is it the purpose of the homework. You should use the homework as a personal gauge for how well you are grasping the material.
* You can find the the homework assignments on JupyterHub. The [What’s Due](/#whats-due) section of the syllabus identifies the homework assignment you should complete.
* Homework are **individual assignments**. You can collaborate on strategy but you must must work alone on the assignment. You must be able to explain the code you write, or it will be considered an academic integrity violation. It’s not about getting it right, but it is about making an honest self-assessment!
* For each homework there is an advice video which provides hints and tips for how you can approach the homework assignment. You are encouraged to only consult the video when you are having difficulty with the homework.
* If you get assistance from somewhere else, such as online, or someone else such as a tutor, or the an AI assistant you must divulge that in your submission or it will be considered an academic integrity violation.

#### Rubric for Homework

Homework Criteria Definitions:

1. **Code Correct** means an honest attempt was made at a solution and the solution is correct. For incorrect solutions, the code runs, might not be correct, but there is some explanation in the reflection.
2. **Code Well Written** means your code is easy to understand, modular in nature, has aptly named variables, was programmed in the style we learned in class, and demonstrates what you learned that week.
3. **Problem Analysis Complete** problem analysis was complete, identifying the problem’s inputs, outputs, and algorithm. An outline of the step-by-step process for how the program should behave.
4. **Questions Complete** means the student made an honest effort to answer the open-ended questions in the lab.
5. **Reflection Complete** means the student completed their code reflection, discussing their experiences with completing the assignment. This should provide insight as to how the work was done touching upon struggles, how you got it done, what was learned in the process.

| Homework Criteria | Assigned Grade |
| --- | --- |
| All 5 criteria met | 3 |
| 4 criteria met | 2 |
| 3 criteria met | 1 |
| Less than 3 criteria met | 0 |

### Recitation Code Assignment (S01 - S13)

Your recitation professor will measure attendance and participation each session through you turning in your recitation code activity. This is the code you work on in class with your classmates. If you are not in class, you will not recieve credit. The specific criteria is entirely at the discretion of your instructor, but most likely consists participating in the programming assignment completed in class.

### Grading Scale For Final Grade

We use the following grading scale for translating your total points earned into a letter grade to be submitted to the University registrar.

| Student Achievement | Total Points Earned | Registrar Grade | Grade Points |
| --- | --- | --- | --- |
| Mastery | 243 - 256 | A | 4.000 |
|  | 230 - 242 | A- | 3.666 |
| Satisfactory | 217 - 229 | B+ | 3.333 |
|  | 204 - 216 | B | 3.000 |
|  | 192 - 203 | B- | 2.666 |
| Low Passing | 179 - 191 | C+ | 2.333 |
|  | 166 - 178 | C | 2.000 |
|  | 153 - 165 | C- | 1.666 |
| Unsatisfactory | 128 - 152 | D | 1.000 |
|  | 0 - 127 | F | 0.000 |

## Course Specific Policies

### Due Dates

* Late work is not accepted. Due to the formative nature of work in the course (practice with timely feedback), it does not make sense to accept late work.
* Due dates are posted on the Syllabus in the [course schedule](#course-schedule) section, specifically [What’s Due?](#whats-due). Due dates are also posted in Blackboard.
* In order to provide timely and relevant feedback, no late work is accepted. Exceptions will only be made under extreme circumstances with supporting University documentation of illness or personal reasons.

### Extra Credit

* No extra credit is offered in this course.
* It is not necessary given a good portion of the work in this course is formative assessment.
* Try and reflect upon your learning. This is what we ask, and will yield the highest marks on the formative assessments.

### Course Withdraw

* Only the main section instructor can sign a withdraw petition.
* We will only sign a withdraw from this course based on poor academic performance (F grade).
* You can always re-take the course and replace your failing grade with a passing grade.
* If you are putting in the time and effort it is very difficult to fail this course in the first place.
* This reserves course withdraws for their original intended purpose - extreme circumstances with supporting University documentation of illness or personal reasons.

### Use of AI Assistants

* It is my opinion that using AI assistants such as ChatGPT to help you understand and write code is extremely valuable to the learning process.
* To provide equitable access to AI, I’ve created a Jupyter plugin, %%pybot which talks to Chat GPT through Jupyterhub. You are welcome to use it in this course.
* When addressing the bounds of academic integrity, treat the AI assistant as you would your personal tutor. You would not copy a tutor’s code and present it as your own, as that’s a clear violation of academic integrity. It is expected you will the outputs from AI similarly. When in doubt, ASK!

### Course Honor Code - Academic Integrity

**What Does Academic Integrity Mean In This Course?**

The course honor code represents our commitment to Academic Integrity in a programming course. I drafted the class honor code to avoid academic negligence - situations where students are unaware that their actions are actually a form of cheating. Our honor code remedies this problem by clearly stating the expectations of Academic Integrity for this course. It states:

1. **All work is my own**. Answers on all student work, assignments (labs, homework, problem sets, projects, papers, etc…) and assessments (quizzes, exams, tests, etc…) are my own individual work (except where collaboration is explicitly allowed and disclosed). In the case where collaboration is permitted I will only collaborate within my team. Your own work means it manifests your own thoughts and ideas, not someone else’s. AI is included in the “someone else.”
2. **I will not share answers.** I will not make answers (either my own or the professor’s) to work, assignments (labs, problem sets, projects, papers, homework, etc…) and assessments (quizzes, exams, tests, etc…) available to anyone else in or out of class. This includes posting them on the web / chegg / course hero, or sharing them in test banks.
3. **I will not misrepresent my ability.** I will not engage in any activity which misrepresents or falsifies my knowledge of the subject matter and therefore improves my grade dishonestly. This includes unsanctioned test aids, copying homework, and assistance from unapproved sources outside of class. This includes passing off AI output as your own work.
4. **I will give credit.** I will always pay attribution to my sources, and not misrepresent the works of others as my own. If you get code from the internet, a tutor, or an AI assistant, you must cite it like you would any source in an academic paper.
5. **I accept the honor code and its consequences.** I understand and accept that that all work I submit is subject to the honor code, and if I violate this honor code I my instructor is obligated to report me to the University’s office of Academic Integrity.

This can be easily summarized as: **If code did not originate from you, you must clearly disclose where it came from.**

When in doubt, ask. When unsure, disclose openly. Most students who get into trouble are trying to hide their academic dishonesty. Don’t do that. We will catch you easily.

### Sanctions for Violations of Academic Integrity

Oops. I violated academic integrity. What now?

* All suspected academic integrity violation will be reported to the university’s office of academic integrity.
* Proposed sanction for violations of a low-stakes assessment such as a homework assignment or lab, is a **grade of zero**.
* Proposed sanction for any violation on a summative assessment such as an exam or the final project is a **grade of F in the course**.

## Syracuse University Policies

Syracuse University has a variety of other policies designed to guarantee that students live and study in a community respectful of their needs and those of fellow students. Some of the most important of these concern:

**Diversity and Disability** (ensuring that students are aware of their rights and responsibilities in a diverse, inclusive, accessible, bias-free campus community) can be found here, at: <https://www.syracuse.edu/life/accessibilitydiversity/>.

**Religious Observances** Notification and Policy (steps to follow to request accommodations for the observance of religious holidays) can be found here, at: <http://supolicies.syr.edu/studs/religious_observance.htm>

**Orange SUccess** (tools to access a variety of SU resources, including ways to communicate with advisors and faculty members) can be found here, at: <http://orangesuccess.syr.edu/getting-started-2>.

### Disability-Related Accommodations

Syracuse University values diversity and inclusion; we are committed to a climate of mutual respect and full participation. There may be aspects of the instruction or design of this course that result in barriers to your inclusion and full participation in this course. I invite any student to meet with me to discuss strategies and/or accommodations (academic adjustments) that may be essential to your success and to collaborate with the Center for Disability Resources (CDR) in this process.

If you would like to discuss disability-accommodations or register with CDR, please visit Center for Disability Resources. Please call (315) 443-4498 or email <disabilityresources@syr.edu> for more detailed information.

CDR is responsible for coordinating disability-related academic accommodations and will work with the student to develop an access plan. Since academic accommodations may require early planning and generally are not provided retroactively, please contact CDR as soon as possible to begin this process. <https://disabilityresources.syr.edu/>

### University Attendance Policy

Attendance in classes is expected in all courses at Syracuse University. Students are expected to arrive on campus in time to attend the first meeting of all classes for which they are registered. Students who do not attend classes starting with the first scheduled meeting may be academically withdrawn as not making progress toward degree by failure to attend. Instructors set course-specific policies for absences from scheduled class meetings in their syllabi.

It is a federal requirement that students who do not attend or cease to attend a class to be reported at the time of determination by the faculty. Faculty should use “ESPR” and “MSPR” in Orange Success to alert the Office of the Registrar and the Office of Financial Aid. A grade of NA is posted to any student for whom the Never Attended flag is raised in Orange SUccess. More information regarding Orange SUccess can be found here, at: <http://orangesuccess.syr.edu/getting-started-2/> Students should also review the University’s religious observance policy and make the required arrangements at the beginning of each semester

### Academic Integrity Policy

Syracuse University’s Academic Integrity Policy reflects the high value that we, as a university community, place on honesty in academic work. The policy defines our expectations for academic honesty and holds students accountable for the integrity of all work they submit. Students should understand that it is their responsibility to learn about course-specific expectations, as well as about university-wide academic integrity expectations. The policy governs appropriate citation and use of sources, the integrity of work submitted in exams and assignments, and the veracity of signatures on attendance sheets and other verification of participation in class activities. The policy also prohibits students from submitting the same work in more than one class without receiving written authorization in advance from both instructors. Under the policy, students found in violation are subject to grade sanctions determined by the course instructor and non-grade sanctions determined by the School or College where the course is offered as described in the Violation and Sanction Classification Rubric. SU students are required to read an online summary of the University’s academic integrity expectations and provide an electronic signature agreeing to abide by them twice a year during pre-term check-in on MySlice.

### Use of Blackboard

This course involves the use of Syracuse University’s Blackboard system as an online tool. The environment is composed of a number of elements that will help you be successful in both your current coursework and your lifelong learning opportunities. To access Blackboard, <http://blackboard.syr.edu> use your Syracuse University NetID & Password. This specific course will appear in your course list.

To search for answers to your Blackboard questions, visit the Answers self-help knowledge <https://answers.syr.edu/display/blackboard01/Blackboard>. If you have problems logging in or need assistance with Blackboard, contact the ITS Service Center at: help@syr.edu or 315.443.2677. The Syracuse University Blackboard support team will assist you.

## Course Schedule

| Dates | Topic (Click Link for Content and Assigned Readings) |
| --- | --- |
| 8/28 - 9/3 | [Lesson 01: Introduction to Python Programming](/content/#lesson-01-introduction) |
| 9/4 - 9/10 | 9/4: Labor Day No Classes9/6: [Lesson 02: Input, Output, Variables and Types](/content/#lesson-02-variables) |
| 9/11 - 9/17 | [Lesson 03: Conditionals](/content/#lesson-03-conditionals) |
| 9/18 - 9/24 | [Lesson 04: Iterations](/content/#lesson-04-iterations) |
| 9/25 - 10/1 | [Lesson 05: User-defined Functions, Modules](/content/#lesson-05-functions) |
| 10/2 - 10/8 | [Lesson 06: Strings and Text Processing](/content/#lesson-06-strings) |
| 10/9 - 10/15 | 10/9: Fall Break No Classes10/11: Exam 2 (E2) In Small Group |
| 10/16 - 10/22 | [Lesson 07: File I/O and Persistence](/content/#lesson-07-files) |
| 10/23 - 10/29 | [Lesson 08: Lists](/content/#lesson-08-lists) |
| 10/30 - 11/5 | [Lesson 09: Dictionaries and JSON](/content/#lesson-09-dictionaries) |
| 11/6 - 11/12 | [Lesson 10: HTTP Protocol and Network Programming](/content/#lesson-10-http) |
| 11/13 - 11/19 | [Lesson 11: Web API’s](/content/#lesson-11-web-apis)11/17: P1 Due |
| 11/20 - 11/26 | No Classes Thanksgiving Break |
| 11/27 - 12/3 | [Lesson 12: Data Analysis with Pandas](/content/#lesson-12-pandas) |
| 12/4 - 12/10 | [Lesson 13: Data Visualization](/content/#lesson-13-visualization) |
| 12/11 - 12/18 | 12/11: Exam 2 (E2) In Large Group12/12: P2 Due (No Class Wed) |
| 12/18 - 12/19 | 12/19: P3,P4 Due |

### What’s Due?

Use this table to track the due dates of all graded work in this course. Dates and times are Eastern Time Zone.

| Date Due | Time Due | Gradebook | Points | Where | What is Due? |
| --- | --- | --- | --- | --- | --- |
| 8/29/2023 | 11:59 PM | L01 | 3 | [Jupyterhub](https://v2hub.ischool.syr.edu) | 01-Intro/LAB-Intro.ipynb |
| 8/30/2023 | 11:59 PM | S01 | 3 | Jupyterhub | 01-Intro/SmallGroup-Intro.ipynb |
| 9/1/2023 | 11:59 PM | H01 | 3 | Jupyterhub | 01-Intro/HW-Intro.ipynb |
| 9/5/2023 | 11:59 PM | L02 | 3 | Jupyterhub | 02-Variables/LAB-Variables.ipynb |
| 9/6/2023 | 11:59 PM | S02 | 3 | Jupyterhub | 02-Variables/SmallGroup-Variables.ipynb |
| 9/8/2023 | 11:59 PM | H02 | 3 | Jupyterhub | 02-Variables/HW-Variables.ipynb |
| 9/12/2023 | 11:59 PM | L03 | 3 | Jupyterhub | 03-Conditionals/LAB-Conditionals.ipynb |
| 9/13/2023 | 11:59 PM | S03 | 3 | Jupyterhub | 03-Conditionals/SmallGroup-Conditionals.ipynb |
| 9/15/2023 | 11:59 PM | H03 | 3 | Jupyterhub | 03-Conditionals/HW-Conditionals.ipynb |
| 9/19/2023 | 11:59 PM | L04 | 3 | Jupyterhub | 04-Iterations/LAB-Iterations.ipynb |
| 9/20/2023 | 11:59 PM | S04 | 3 | Jupyterhub | 04-Iterations/SmallGroup-Iterations.ipynb |
| 9/22/2023 | 11:59 PM | H04 | 3 | Jupyterhub | 04-Iterations/HW-Iterations.ipynb |
| 9/26/2023 | 11:59 PM | L05 | 3 | Jupyterhub | 05-Functions/LAB-Functions.ipynb |
| 9/27/2023 | 11:59 PM | S05 | 3 | Jupyterhub | 05-Functions/SmallGroup-Functions.ipynb |
| 9/29/2023 | 11:59 PM | H05 | 3 | Jupyterhub | 05-Functions/HW-Functions.ipynb |
| 10/3/2023 | 11:59 PM | L06 | 3 | Jupyterhub | 06-Strings/LAB-Strings.ipynb |
| 10/4/2023 | 11:59 PM | S06 | 3 | Jupyterhub | 06-Strings/SmallGroup-Strings.ipynb |
| 10/6/2023 | 11:59 PM | H06 | 3 | Jupyterhub | 06-Strings/HW-Strings.ipynb |
| 10/11/2023 | 11:59 PM | E1 | 45 | Small Group | E1: Exam 1 (Focus on Lessons 01-06) |
| 10/17/2023 | 11:59 PM | L07 | 3 | Jupyterhub | 07-Files/LAB-Files.ipynb |
| 10/18/2023 | 11:59 PM | S07 | 3 | Jupyterhub | 07-Files/SmallGroup-Files.ipynb |
| 10/20/2023 | 11:59 PM | H07 | 3 | Jupyterhub | 07-Files/HW-Files.ipynb |
| 10/24/2023 | 11:59 PM | L08 | 3 | Jupyterhub | 08-Lists/LAB-Lists.ipynb |
| 10/25/2023 | 11:59 PM | S08 | 3 | Jupyterhub | 08-Lists/SmallGroup-Lists.ipynb |
| 10/27/2023 | 11:59 PM | H08 | 3 | Jupyterhub | 08-Lists/HW-Lists.ipynb |
| 10/31/2023 | 11:59 PM | L09 | 3 | Jupyterhub | 09-Dictionaries/LAB-Dictionaries.ipynb |
| 11/1/2023 | 11:59 PM | S09 | 3 | Jupyterhub | 09-Dictionaries/SmallGroup-Dictionaries.ipynb |
| 11/3/2023 | 11:59 PM | H09 | 3 | Jupyterhub | 09-Dictionaries/HW-Dictionaries.ipynb |
| 11/7/2023 | 11:59 PM | L10 | 3 | Jupyterhub | 10-HTTP/LAB-HTTP.ipynb |
| 11/8/2023 | 11:59 PM | S10 | 3 | Jupyterhub | 10-HTTP/SmallGroup-HTTP.ipynb |
| 11/10/2023 | 11:59 PM | H10 | 3 | Jupyterhub | 10-HTTP/HW-HTTP.ipynb |
| 11/14/2023 | 11:59 PM | L11 | 3 | Jupyterhub | 11-WebAPIs/LAB-WebAPIs.ipynb |
| 11/15/2023 | 11:59 PM | S11 | 3 | Jupyterhub | 11-WebAPIs/SmallGroup-WebAPIs.ipynb |
| 11/17/2023 | 11:59 PM | H11 | 3 | Jupyterhub | 11-WebAPIs/HW-WebAPIs.ipynb |
| 11/17/2023 | 11:59 PM | P1 | 0 | Jupyterhub | project/P1.ipynb |
| 11/28/2023 | 11:59 PM | L12 | 3 | Jupyterhub | 12-Pandas/LAB-Pandas.ipynb |
| 11/29/2023 | 11:59 PM | S12 | 3 | Jupyterhub | 12-Pandas/SmallGroup-Pandas.ipynb |
| 12/1/2023 | 11:59 PM | H12 | 3 | Jupyterhub | 12-Pandas/HW-Pandas.ipynb |
| 12/5/2023 | 11:59 PM | L13 | 3 | Jupyterhub | 13-Visualization/LAB-Visualization.ipynb |
| 12/6/2023 | 11:59 PM | S13 | 3 | Jupyterhub | 13-Visualization/SmallGroup-Visualization.ipynb |
| 12/8/2023 | 11:59 PM | H13 | 3 | Jupyterhub | 13-Visualization/HW-Visualization.ipynb |
| 12/11/2023 | 11:59 PM | E2 | 45 | Large Group | E2: Exam 2 (Focus on Lessons 07-13) |
| 12/12/2023 | 11:59 PM | P2 | 0 | Jupyterhub | project/P2.ipynb |
| 12/19/2023 | 11:59 PM | P3 | 0 | Jupyterhub | project/P3.ipynb |
| 12/19/2023 | 11:59 PM | P4 | 49 | Jupyterhub | project/P4.ipynb |

Page last revised on: {{ git\_revision\_date }}