



IS 3020

Syllabus

Application Development

Fall 2020 Hybrid & Online

Preferred communication method: Microsoft Teams tool. We will monitor it daily. Get the app for iOS or Android. Or, download the tool for Windows or MacOS / login using your KSU email at <https://teams.microsoft.com>. Then, click join team. Enter this code: **8qpr55e** This tool is used by professionals engaged in application development all over the world, and the experience of using it in this class is part of your learning (Goal 3).

Hybrid Instructor: Dominic Thomas, Ph.D.

Email: dominic.thomas@kennesaw.edu **Phone:** (470) 578-5201

LinkedIn: <http://www.linkedin.com/in/dominict> (connect; mention our class)



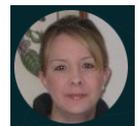
Office: Burruss 495

Office Hours: Online in Teams. In Teams, look for the green circle next to my icon indicating that I am available. Tag me in a post in our channels. Or, send me a chat in Teams and it will ping me to login if I can in a good place to login then we can have a video meeting session.

Online Instructor: Stephanie Sample, Ph.D.

Email: ssamp113@kennesaw.edu **Phone:** (404) 797-4248

LinkedIn: <https://www.linkedin.com/in/stephsample/> (connect; mention our class)



Office Hours: Online in Teams. In Teams, look for the green circle next to my icon indicating that I am available. Tag me in a post in our channels. Or, send me a chat in Teams and it will ping me to login if I can in a good place to login then we can have a video meeting session.

Course Location and Timing

Online using D2L and Teams PLUS **in class for Hybrid section on Mondays 10:10 – 11:00am BB 150** (Group A Last names starting with A through J): Aug 24, Sep 14, Sep 21, Oct 5, Oct 12, Oct 19, Oct 26, Nov 9, Nov 16. (Group B Last names starting with K through Z): Aug 26, Sep 16, Sep 23, Oct 7, Oct 14, Oct 21, Oct 28, Nov 11, Nov 18. If the room is not full, other students are welcome to join – class will simulcast in Teams for online participation.)

Catalog Description

This course will provide students with the knowledge and skills needed to develop applications in a development environment. Program logic and decision structure will be covered using a modern programming language and framework.¹

Required Materials/Resources

- Bring a **laptop computer** every day. You will be setting it up as a development machine in order to learn more deeply about application development. You will install the latest version of Python 3 (available for free at python.org) and other software for this course here. Having it will you enables collaborative development and troubleshooting.
- **Microsoft Teams account and software** (provided by KSU as part of O365, must sign in using your KSU email address) Teams is part of Office 365. Login using your KSU email at <https://teams.microsoft.com>. Then, click join team in the lower left. Enter the code listed on the previous page. Next, click the “Get App” icon in the left sidebar and install the application. Download the app on your mobile and the main software on your laptop. The full application is much faster and more powerful than the browser-based one.
- **Microsoft Github account** (provided by KSU as part of O365, must sign up using your KSU email address, sign up at: <https://education.github.com/pack/join> You will receive a bunch of software licenses and additional resources with your account including free web hosting). Mid-semester, you will need to accept the invite to be part of the KSU-IS organization within Github. This invite expires in 7 days. So, catch it when it arrives.
- **[Optional] Microsoft Azure account** (provided by KSU as part of O365, must sign up using your KSU email address) You can get additional resources by logging in with your KSU email and claiming your Azure credit at: <https://azureforeducation.microsoft.com/devtools> (Note that various Microsoft apps and tutorials are available for free here too!)
- **Microsoft Visual Studio Code development software** with Python, Python Extensions [Optional], and Code Runner extensions enabled. Available free for Mac/Windows/Linux at: <https://code.visualstudio.com>
- Our D2L course site where various materials are/will be posted. You must turn in all exercises in D2L.
- [Optional] Get a **Raspberry Pi** in order to complete our RaspberryPi tutorial at home. Then, take it further with additional projects! Search for one with a RaspberryPi along with the memory card, breadboard and some sensors and LEDs, such as CanaKit Raspberry Pi 3 B+ (B Plus) Ultimate Starter Kit (32 GB Edition, Clear Case) <https://www.canakit.com/raspberry-pi-3-model-b-plus-ultimate-kit.html>
- [Optional] There are many free resources for learning Python online. Some are decent. Many are not. We recommend this resource, which guides you to several free Python books depending on your interest in case you want a book:

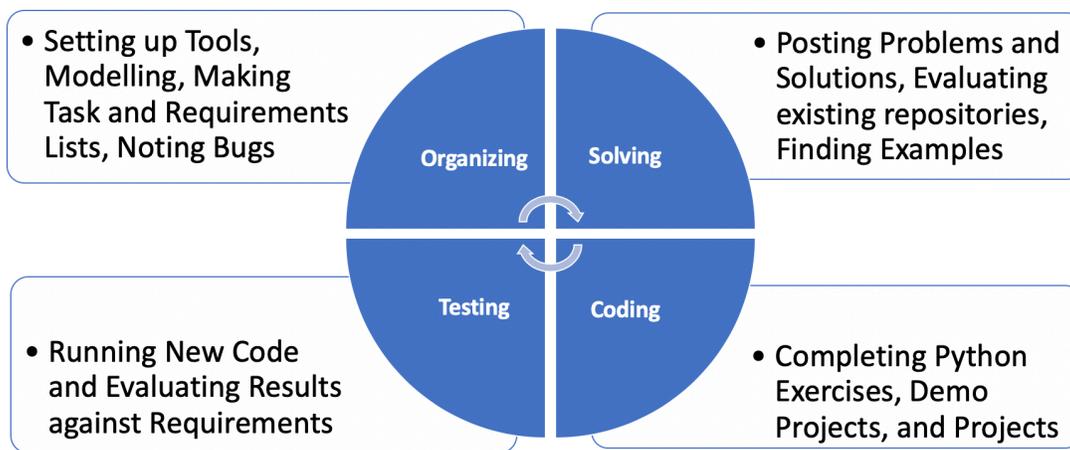
¹ Prerequisites: [IS 2200](#), 60 credit hours with a minimum GPA of 2.0, and Admission to the Coles College or student in a Coles College Partner Program that includes this course. This is a 3 “credit hour” course. See <https://www.carnegiefoundation.org/blog/the-carnegie-unit-revisited/>

<https://blog.rmotr.com/the-3-python-books-you-need-to-get-started-for-free-9b72a2c6fb17>

- **Anaconda Navigator development tool and enhanced Python** release. Free leading Python deployment for data analytics with Jupyter notebook support to run the course tutorials, examples, and coding exercises on your own machine. Available for Mac/Windows/Linux at <https://www.anaconda.com>

Course Model/Process

We will cycle through the Systems Development Life Cycle (SDLC) in this class. The SDLC typically includes 4 steps that happen over and over in the life of an application development project. These steps are often termed Planning, Analysis, Design, and Implementation (PADI) when projects go all the way through installation. In our case, we will not go that far, so, our steps are related but abbreviated. We will do the following:



Grading

This is a hands-on class. Students will be evaluated on the following items:

Course Item	Responsibility	Weight
Exercises (Modeling and Misc. Exercises)	Individual	20%
Intro. to Python: Absolute Beginner	Individual	20%
Intro. to Python: Fundamentals	Individual	20%
Attendance and Participation	Individual	10%
2 Demonstration Projects	Individual	10%
Project	Individual/Team	20%

Notice that at least 4/5 of your grade is individual. *Projects can be individual. Project teams can be no larger than 2 people.

Learning Goals, Objectives & Assessments

Goals: Upon successful completion of this course, students will...	Objectives: Upon successful completion of this course, students will be able to...	Assessments: These activities will show levels of progress...
1. understand how to capture the unstructured requirements for software into a structured model and plan distinguishing what the software needs to do based on priority.	a) Capture details from a narrative and model them in a cross functional flowchart identifying the main activity, the human agents and their tasks, the computer agents and their tasks, and data elements. b) Use appropriate symbols aligned with Business Process Modeling Notation (BPMN) standards for drawing and reading elements in a model. c) Distinguish digital and physical data as well as an activity flow in a flow model.	System Blueprinting Exercises and Quiz; Project (workplan and any models produced)
2. understand basics of coding in a modern programming language.	Define variables. Recognize and transform their data types. Use them in code. Create logic and encapsulate it into functions or methods of objects. Make logic conditional and nested into loops for repetitive tasks and scaling. Process strings and lists (aka arrays) using iteration and sequence references with methods. Access and store data using files and the filesystem.	Intro. to Python: Absolute Beginner (quizzes + coding assignments) & Intro. to Python: Fundamentals (quizzes + coding assignments)
3. know ways that application development can lead to errors and learn to document them and find help in solving them.	a) Describe coding problems clearly so others can understand what happened and assist in providing solutions. b) Describe coding solutions clearly so others can understand how to apply them to fix their problems. c) Recognize the need to take a break or try another means to solve a coding problem when current methods/approach is failing.	Attendance, Quantity and Quality of Posts to Microsoft Teams Tool as well as Github repository comments during projects
4. understand how different coding tools work and can augment or hinder productivity in	a) Install Python on a PC/Mac and use the included IDLE tool for coding. b) Create a cloud-based account and type and run code in it.	Coding checks during python coding modules; Project (code progress and

software development.	c) Download, install, configure, and use Visual Studio Code to code software.	methods); Demo Projects
5. understand the importance of managing code bases in sharable tools (repositories) and tracking changes to improve productivity.	<p>Create a repository in Github and set it up correctly with a readme.md file.</p> <p>Submit changes to code as documented commits in an online, shared repository.</p> <p>Download and sync an online, shared repository to a personal computer to enable local development.</p>	Project (Github repository activities)
6. know how to go through the process of developing a targeted piece of software	<p>a) Decide on a target software idea to develop and source and evaluate existing solutions related to it.</p> <p>b) Design additional features, code them, and commit them to a software repository with documentation.</p> <p>c) Describe a software project so others can understand it and its intended audience as well as its value.</p> <p>d) Test changes to a software code base and analyze the results to define next steps.</p>	Github commits to codebases plus progress report meetings in Project; project marketing slide; final presentation

Course Components & Policies

Exercises/Projects

We will begin the semester learning how to design software. The technique we will use is called System Blueprinting. It is a form of cross-functional flow charting (see details in D2L) that captures the actions of the software as well as the people involved in completing a targeted activity. It notes how data are produced and consumed as well (Goal 1). Additionally, we will learn how to lean on and support each other as we learn process modeling and coding (Goal 3). This is how collaborative, modern software development works. We will all be grouped into a team in Microsoft Teams, and we will expect you to post questions and answers concerning emerging challenges and solutions in our team space (Goal 3). The quality of your posts matter. Post screenshots and details!

We will be doing some demonstration exercises beyond what you see listed in the Python tutorial modules that include making web pages with Python, web scraping with Python, and others. Shorter/smaller exercises will focus on some deeper coding concepts like how to write functions, make conditionals, conduct sort algorithms, coordinate a code repository for a team with a tool like Git and Github (Goal 5), and what causes code to slow down (Goal 6). The Modeling quiz will be included in this part of your grade. It focuses on knowledge of the shapes and where they go. The final projects come toward the end of the class, the goal is for you alone or with a partner (a maximum team of two) to create a working piece of software using Python that accomplishes some business purpose in line with an initial idea/design (Goal 3 and Goal 6).

The final projects you complete will be managed in sprints, *which account for most of the credit in the project* (Goal 6). Also, they must be completed on time, as the process matters in application development and takes time for issues to emerge and get resolved (Goals 1 and 3). Sprints are a concept drawn from a very popular, current project management methodology termed “agile.” The idea is that you will set short goals for your sprint time-period and evaluate how you did on them at the end. Then, you plan and start the next sprint. Typically, sprints last about two weeks in industry. In our class, they will be faster than that, since we do not have that much time.

Sprint 1 Requirements

Here are things that you should have completed in the first sprint.

1. Each person should have an approved project topic (Turn in Ex 6 and complete your project listing in the Projects spreadsheet in the Projects channel in Teams).
2. Each person should have established a Github account and joined our class organization (KSU-IS) in Github by accepting the invite email.
3. Each person should also be associated with ownership of a code repository in our Github organization. If on a team, ensure each has Owner or Maintainer Role.
4. Each person should have edited and committed changes in a readme.md file in their repository in our Github organization. The commits should show in Github.
5. Each person should have explored at least one existing Python code base related to their project to evaluate using it in their project and submitted updates to the projectroadmap.md file explaining their findings. In evaluating your codebase, you should have read the code and attempted to run it yourself.

Sprint 2 Requirements

During sprint 2, you should be coding and using git to manage your code.

1. Each person should be making small updates and adding commit messages with them in your repository in our Github organization so that your work is visible. Do this with Github desktop. Otherwise, it will appear you are not working, and you will fail to demonstrate your knowledge of the application development process, which is one of our course goals. Thus, even if you claim to be working on your own machine but fail to post updates in Github in our shared organization where the professor can see them, you will fail on this grading item. Each person should have committed at least 6 code changes of significant size (not just a spelling error fix for example). Each commit should include a comment that explains what you did. The comment must be specific (ie. not just Updated README.md but rather explain what your update was and why you did it briefly). You will have to paste URLs to 2 code commits.
2. Also, you need to be tracking your progress on planned and emerging tasks in your projectroadmap.md document. We will look for changes in this document as signs of your progress. Using checkboxes, adding new tasks as they emerge, or adding “DONE” to mark done items all could count here. You will be asked to paste a URL to one of your commits.

Sprint 3 Requirements

1. Continue coding and refining and testing in Sprint 3 as in Sprint 2. We will look for additional progress. If you had minor scope for your project and finish already, you need to add more scope and work on it.

In addition, coders typically think little about marketing and sustaining their projects. Inside a business there are many stakeholders who are not technical. As project managers and communicators who span the gap between the business and the coders, we need to create ways to communicate our projects quickly and accurately. This is just such an activity.

2. Create **one** PowerPoint slide introducing your project and upload it. If you have more than one teammate, each needs to turn-in a copy in D2L to ensure everyone has a copy and gets credit. The slide needs to be attractive and informative. Be sure to include this information:
 - List your project team members.
 - Show the title of your project.
 - Show a tag line that introduces the main concept of what it does/will do.
 - Show 1-2 screenshots or pictures demonstrating the idea or parts (optional) Each person must upload a PPT slide in D2L. Each team must ensure there is a copy in their Github repository so that future coders can quickly grasp the idea.

Project Presentation Requirements and Samples

You will present your project at the end of the semester. The presentation targets 5 minutes and should focus on your showing your code and talking about what you did and tried. You can use up to 3 PowerPoint slides to frame your work. You will be recording and uploading your presentations to our Teams presentations channel. Then, the class and we can see them. This ensures we have enough time and every presentation is high quality. Be sure to put a link to your presentation in your Github repo too. Sample projects from prior offerings of this course include (you can download and try these from Github at: <https://github.com/ksu-is>):

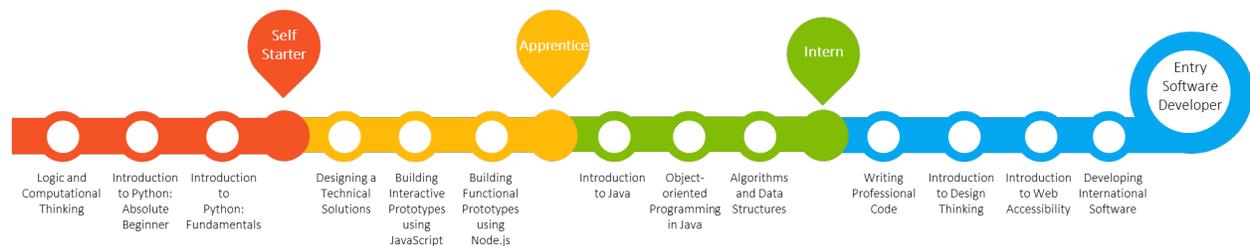
- Doggy Tracks -> Flask (ie. WWW), dog adoption system
- Remnants -> complex decision tree adventure game w/ functions
- Simple Inventory -> Flask to track inventory
- Unist Hitbox Viewer -> Tkinter .pack() method w/ graphics and buttons
- Recip Manager -> Django dependencies and setup
- Sports Stats -> script/package customization, scraping
- RHA Inventory System -> Flask + RaspberryPi barcode scanner
- Tic-Tac-Toe -> PyGame w/extra rows and cols, timed
- Georgia Brewery DB -> Flask app for brewery lookup w/Excel import
- Flappy Bird + -> PyGame – modified to add enemy objects
- Receipt Parser -> PyTesseract and OpenCV, embedded C++
- An Unfortunate Fright Night -> Tkinter text-based horror game w/text animation
- Operation Sharpshooter -> RaspberryPi quick reaction game
- Google Shade -> Jupyter based map shading and selection
- Guess The Number -> Flask, guess the number game for 3rd graders
- Inedible -> text-based scrape yelp for lowest rating, randomize choice
- Soccer Manager -> text-based, assigns teams based on experience
- Gradebook -> read/write to Excel
- Snake Game -> PyGame – timer + pop-up messages

Intro. To Python Tutorials

We will use materials from two of the three required courses (Absolute Beginner and Fundamentals) that Microsoft uses to train new software developers in their business. To verify your progress, we will require you to complete the 8 modules in the two Python courses

(Absolute Beginner and Fundamentals) and post screenshots of your progress in D2L (Goal 2). Four-fifths of your grade will be for timely, earnest completion of the coding tutorials and the final code exercise in each Python course.

The final 20% will derive from coding quizzes on D2L for each module. Be sure to turn in both the screenshots and required code in a .py file to verify module completion as well as complete the coding quizzes in D2L. The coding quizzes are related to each other. So, you need to complete them in order. Some will only appear as available to you after you complete the prior ones. This ensures that you learn the skills in order and do not skip any. The modules do assume prior knowledge and build off of each other. Some quizzes do expire. So, if you do not complete them on time, you get no chance to make them up.



Late Work

The timing of your work in our class is very important. We use your knowledge from your work to enable our discussions and sharing with each other. When you do not keep up, you and your colleagues lose out on learning. Thus, we want you to turn things in on time. At the same time, we know life happens. We want you to complete our work during the semester even if it is late, because you will get value from it. Thus, we accept late work for most items, but we penalize it. If it is just a little late (within a week), you will lose 10%. After that, the losses continue.

Attendance & Participation

Attendance and participation is important. Ask and help each other. This skill is practice for technology company culture, wherein employees help each other and ask questions (Goal 3). The only content you cannot directly discuss/screenshot is the quizzes and required code. We will check our logs to see when you last logged in. We will be monitoring your online presence looking for you to post in Teams, because participating exposes you to our discussions and topics. That experience gives you learning potential. We dedicate 5% of your grade to this aspect. In the face-to-face version of our course we multiply your attendance percent times 3% then add points for additional participation in Teams. Generally, you will earn for each meaningful post (question, event write-up, or answer) in Teams.

The other 5% comes from IS Engagement event attendance or by attending Python/application development events. PyAtl (<https://www.meetup.com/python-atlanta/>) and PyData Atlanta (<https://www.meetup.com/PyData-Atlanta/>) events are pre-qualified as are other application development webinars. Others require an instructor to approve them though we will approve almost any event related to application development. To earn the credit, share a brief write up of what you learned and how the event went to our Microsoft Teams General channel with a selfie you took during the event. For each IS Engagement event you earn 1%. Find a list of IS Engagement events here: <https://owllife.kennesaw.edu/events?query=%23ISENG> Keep

track of these to report them in D2L. You can also view videos of past events for credit at: <https://mediaspace.kennesaw.edu/category/Coles/24065201>

Statement on Physical/Emotional Health

A range of issues can cause barriers to learning, such as strained relationships, increased anxiety, health issues, alcohol/drug problems, feeling down, difficulty concentrating, lack of motivation or feeling ill. These concerns or other stressful events may lead to diminished academic performance or may reduce your ability to participate in daily activities. University resources can help you address these and other concerns. You can learn more about KSU's broad range of medical and confidential mental health services at: <http://studenthealth.kennesaw.edu/> and <http://counseling.kennesaw.edu/>

Classroom Environment and Expectations



What you can expect of me:

What we expect of you:

-
- | | |
|---|---|
| <ul style="list-style-type: none">• Come prepared to every class.• Stay focused on your learning.• Exhibit only professional behavior.• Design course and activities to achieve stated goals and learning objectives.• Listen, guide, and create environment for learning.• Foster mutual respectful learning environment in classroom.• Provide a model of how experts in the field diagnose and solve problems.• Consider that it's not always your fault if you do not understand the material.• Use best professional judgment to evaluate performance fairly, and not be capricious or prejudiced in any way.• Preserve fairness of performance evaluation by upholding high standards of academic integrity. | <ul style="list-style-type: none">• Come prepared to every class.• Stay focused on your learning.• Exhibit only professional behavior.• Complete all work required on time, and with proper attention and thought.• Listen, stay actively involved, study, and learn.• Treat classmates and instructor with appropriate respect.• Respect instructor's expertise in this field and emulate the model provided.• Consider that it's not always the instructor's fault if you do not understand the material.• Recognize that instructor uses best professional judgment to evaluate performance and is not "out to get" students.• Preserve fairness of performance evaluation by adhering to high standards of academic integrity. |
|---|---|

We are bound by the KSU Academic Integrity Statement (listed below). There is a **zero tolerance** policy for violations in this course.

Statement on Accommodations for Students with Disabilities

If you anticipate issues with the format or requirements of this course, please meet with me—we would like to discuss ways to ensure your full participation in my classroom. If you determine that you need formal, disability-related accommodations, it is very important that you register with the Office of Student Disability Services (<http://sds.kennesaw.edu/>) and notify your instructor of your eligibility for reasonable accommodations. We can then plan how best to implement your accommodations.

Institutional Policies

Academic Affairs - Federal, BOR, & KSU Policies

http://curriculum.kennesaw.edu/resources/ksu_student_resources_for_course_syllabus.php - this link provides information and current policies on additional technology resources including how to get help; academic resources for disabilities, tutoring, ESL, the library; wellness and how to report any concerns about other students; and grade appeals.

Academic Integrity Statement

Every KSU student is responsible for upholding the provisions of the Student Code of Conduct, as published in the Undergraduate and Graduate Catalogs. Section 5c of the Student Code of Conduct addresses the university's policy on academic honesty, including provisions regarding plagiarism and cheating, unauthorized access to university materials, misrepresentation/falsification of university records or academic work, malicious removal, retention, or destruction of library materials, malicious/intentional misuse of computer facilities and/or services, and misuse of student identification cards. Incidents of alleged academic misconduct will be handled through the established procedures of the Department of Student Conduct and Academic Integrity (SCAI), which includes either an "informal" resolution by a faculty member, resulting in a grade adjustment, or a formal hearing procedure, which may subject a student to the Code of Conduct's minimum one semester suspension requirement. See also KSU Student Code of Conduct (<https://web.kennesaw.edu/scai/content/ksu-student-code-conduct>). **We will use the Respondus Lock-down browser for online Python quizzes and check for any improper behavior via the webcam and logs.**

COVID-19 Special Notes

KSU has updated the KSU's Coronavirus website (<https://coronavirus.kennesaw.edu/>) with new information and FAQs for fall 2020. Students, faculty and staff are expected to visit the site for additional, detailed information. Brief answers to frequent questions:

- Attendance to class in person is optional in IS 3020. Students may choose to attend and participate virtually instead via Teams. Once classroom capacity is reached on any given day, no one else will be allowed to sit in the room. Some students have already asked to be all virtual. So, there are likely to be plenty of seats most days.
- KSU partnered with Wellstar to deliver Covid-19 testing on both the Marietta and Kennesaw campuses for students, faculty and staff.
- Classrooms will be cleaned 2-3 times per day. Wipes will be provided for students to clean their table/desks. Wipes will also be provided so that the instructor can clean their workstation.

- Some classrooms are being modified to allow for social distancing.
- Masks are required in all public spaces including classrooms.
- In cooperation with the Department of Public Health, KSU has trained a contact tracing team that will be in place at the beginning of the fall term

Face Masks in the classroom

As mandated by the University System of Georgia, the university requires the use of face masks in the classroom and in KSU buildings to protect you, your classmates, and instructors. Per the University System of Georgia, anyone not using a face covering when required will be asked to wear one or must leave the area. Repeated refusal to comply with the requirement may result in discipline through the applicable conduct code.

Reasonable accommodations may be made for those who are unable to wear a face covering for documented health reasons. Please contact Student Disability Services at sds@kennesaw.edu for student accommodation requests.

Shifting Modalities

Please note that the university reserves the right to shift teaching modalities at any time during the semester, if health and safety guidelines require it to do so. Some teaching modalities that may be used are F2F, Hyflex, Hybrid, or online, both synchronous and asynchronous instruction.

Staying Home When Sick

If you are ill, please stay home and contact your health professional. In that case, please email the instructor to say you are missing class due to illness. Signs of illness include, but are not limited to, the following:

- Cough
- Fever of 100.4 or higher
- Runny nose or new sinus congestion
- Shortness of breath or difficulty breathing
- Chills
- Sore Throat
- New loss of taste and/or smell

Seating Plans

Students will sit in the same seat for every F2F class so that the instructor can use a seating plan for contact tracing if a student contracts Covid-19.

Web Cameras

Instructors may require web cameras in their respective courses.

Electronic Communication

Use Teams to communicate. You can post to channels and open direct chats there. It is an officially-supported KSU tool. The University provides all KSU students with an “official” email account with the address “students.kennesaw.edu” or “kennesaw.view.usg.edu” (in D2L). As a

result of federal laws protecting educational information and other data, this is the sole email account you should use to communicate with your instructor or other University officials. Note that the D2L email often fails. Please avoid it. Please use dominic.thomas@kennesaw.edu directly instead.

Schedule

Notice that we have a lot of **Work Days** (with **orange backgrounds**). These are days when we work and interact via Teams. In part, they also duplicate the current application development work happening in industry. Much of that work is conducted virtually. You need to learn to use a tool such as Teams to document issues and questions effectively as well as to help your colleagues. We will be looking for that communication and participating too. All actual due dates are posted in D2L. Some items listed here are earlier and prompt you to get moving and are relative to when they are actually due in D2L, which might be a day or more later.

Date	Day	Topic	Assignment(s)
17 Aug	Mon	Get code and Use it: Application Development = Coding + Management	VideoQuiz1: Watch the intro video. (code.org, Thomas intro) - setup Python and Teams
19 Aug	Wed	Get code and Use it: Setup and use Python 3 and Teams. Prepare to manage your time effectively.	Ex1. Turn in IDLE screenshot; Ex2. Intro in Teams VideoQuiz2: Github Desktop Tutorial; VideoQuiz3: Intro to Code via Gradetracker.py
21 Aug	Fri	Modeling: Intro	Ex3. Hack GradeTracker.py; VideoQuiz4: Code Logic and Modeling
24 Aug	Mon	Modeling: Modeling Software Makes It Easier Focus: TacoCopter Q & A + Modeling Software	Ex4. TacoCopter (follow the video and use software to make your model)
26 + 28 Aug	Wed, Fri	Modeling: Interpret Scenarios into Structure Conveyor Sushi Q & A	Ex5. Conveyor Sushi; Conveyor Sushi part 2 is extra practice + review Modeling Quiz in D2L: Know the shapes with their specific uses and the exact reasons why things go in which rows!
31 Aug + 2 Sep	Mon, Wed	Python I: Setup Focus - Post Questions/Issues to Teams	VideoQuiz5: How to Learn to Code in this Class (setup Anaconda, get code, learn what we will learn, etc.)
4 Sep	Fri	Python I: Basics Focus: KSU Scrape Demo	Watch Variables video. Complete P1M1 - post questions/difficulties/tips in Teams;

			complete P1 Quiz 1; Watch Project Demo KSU Scrape video
9 + 11 Sep	Wed, Fri	Python 1: Functions Focus - Post Questions/Issues to Teams	Watch Functions video.
14 Sep	Mon	Python 1: Functions Focus: Functions and Intro Conditionals	Complete P1M2 - post questions/difficulties/tips in Teams; complete P1 Quiz 2
16 + 18 Sep	Wed, Fri	Python 1: Conditionals Focus - Post Questions/Issues to Teams	Complete P1M3 - post questions/difficulties/tips in Teams; complete P1 Quiz 3;
21 + 23 Sep	Mon, Wed	Python 1: Nesting and Loops Focus: Nesting and Loops	Complete P1M4 - post questions/difficulties/tips in Teams; complete P1 Quiz 4
25 Sep	Fri	Python 1: Programming Paradigms & User Interfaces Focus - Post Questions/Issues to Teams	Watch Project Demo: calcGUI Walkthrough
28 Sep	Mon	Python 1: Final Code Focus: CalcGUI demo, Q & A, Troubleshooting	Do final coding evaluation activity for Python 1 and submit P1M5.
30 Sep + 2 Oct	Wed, Fri	Python 2: Sequence, Index, Finite Loops Focus - Post Questions/Issues to Teams	Watch Project Demo: Flask Webpage Walkthrough
5 + 7 Oct	Mon, Wed	Python 2: Web Application Focus: Webpage with Flask	Complete P2M1 - post questions/difficulties/tips in Teams; complete P2 Quiz 1
9 Oct	Fri	Python 2: Sequence Manipulation Focus - Post Questions/Issues to Teams	Watch Project Demo: DB with Python Walkthrough
12 + 14 Oct	Mon, Wed	Python 2: Database Focus: DB in Python	Complete P2M2 - post questions/difficulties/tips in Teams; complete P2 Quiz 2
16 Oct	Fri	Python 2: Demo Work Focus - Post Questions/Issues to Teams	

19 Oct	Mon	Python 2: Sequence Iteration Focus: Q & A, Troubleshooting	Complete P2M3 - post questions/difficulties/tips in Teams; complete P2 Quiz 3
21 Oct	Wed	Python 2: IoT - physical + digital Focus: RaspberryPi - build devices and run them with our code. Get a kit in advance to follow along.	Watch Project Demo: RaspberryPi Physical Digital Merge Walkthrough
23 Oct	Fri	Python 2: Files and Final Code Focus: File Handling + Excel, Q & A, Troubleshooting	Complete P2M4 - post questions/difficulties/tips in Teams; complete P2 Quiz 4; Turn in Demo Project Code 1 and 2. Turn in Project Idea Ex6. (Join our KSU-IS Github organization by accepting the invite emailed to you. If you do not see the email invite, check spam and/or contact the Professor. They expire if not opened in 2 weeks.)
26 Oct	Mon	Focus - Post Questions/Issues to Teams	Do final coding evaluation activity and submit P2M5.
28 Oct – 4 Nov	Wed-Wed	Project: Sprint 1 Focus: Launch Projects	Ensure everyone is started and has themselves setup. Check the spreadsheet in Teams to ensure your project details are there and correct and that your row is marked as approved. If not, work out the issues with the Professor.
6 Nov	Fri	Project: Sprint 1 Focus - Post Questions/Issues to Teams	Setup your Github repository and associated files. Make sure it is in KSU-IS. Find and evaluate approaches/code. Update your documents in Github (README, projectroadmap, etc.)
9 Nov – 18 Nov	Mon-Wed	Project: Begin Sprint 2 Focus: Q & A, Troubleshooting	Complete the Sprint 1 Review and Log in D2L (Quiz): Make lots of small commits as you develop your code. Pro developers commit 6+/hour
20 Nov	Fri	Project: Sprint 2 Focus - Post Questions/Issues to Teams	Test and edit your code noting your learning in the commits you make in Git. Note your learning through updates to the project roadmap. Use Github desktop to post changes.

30 Nov – Dec 4	Mon- Fri	Project: Sprint 3 Focus: Who is the Github repository for? Why do I need to package my work and integrate and test it as it precedes?	Complete the Sprint 2 Review and Log in D2L (Quiz); Design and Turn in your PPT marketing slide for your project.
7 Dec	Mon	Project: Sprint 3 Focus: What makes my project complete? Testing and Q/A	Complete your work. Package and plan your presentation. Submit your presentation materials in D2L (could be screenshot from presentation in Teams or your PPT).
8 – 14 Dec	Finals Week	Project Presentations	Schedule your presentation. Then, present and record it in the Projects channel in Teams. Invite friends and faculty.

Why learn to code with Python? 6 Reasons

March 13, 2020 (excerpted article found at:- <https://codingnomads.co/blog/business/why-learn-python/>)

1. Python is versatile

What is Python used for?

Just to name a few, Python is used in Data Mining, Data Science, AI, Machine Learning, Web Development, Web Frameworks, Embedded Systems, Graphic Design applications, Gaming, Network development, Product development, Rapid Application Development, Testing, Automation, the list goes on.

Python is used as an easier and more efficiently-written alternative to languages like C, R, and Java, and is growing in popularity as the primary language for many applications.

Python.org lists various **success stories** from different industries and applications that use Python. The length of this list alone shows you how versatile Python uses are. One interesting story is of **BATS**, which aims to provide access to maps for the visually impaired. Another story is a **workflow automation system for NASA** which was written in Python by one person in less than a year.

What types of companies use Python?

Python is great for quick prototyping, hence is used extensively by startups to build their first minimum viable product (MVP). As a highly scalable language, Python is also used in large companies. In a recent blog post, **Netflix** discussed its uses of Python in everything from their Content Delivery Network (CDN) to their monitoring systems.

Google also loves Python programming for its solutions. Peter Norvig, director of research at Google, **said that**,

Python has been an important part of Google since the beginning, and remains so as the system grows and evolves. Today dozens of Google engineers use Python, and we're looking for more people with skills in this language.

According to **iDataLabs** 69% of the companies that use Python are small (<\$50M in revenue), 9% are medium-sized (\$50M – \$1000M in revenue) and 16% are large (>\$1000M in revenue).

So how can you maximize the number of industries that would be open to hiring you? Choose a language that is used across the board. Whether you want to work for Google building their next worldwide application or for Pixar to help produce their next feature film, Python opens the doors to any industry.



Image source: [CodingNomads](#)

2. Python is the fastest growing programming language

According to Codecademy, 2018 saw a 34% increase in users studying Python. Stackoverflow.com also determined Python to be the **fastest growing major programming language** by analyzing the number of visitors vs. questions about a certain topic.

Growth of Python Queries in Stack Overflow



Image source: [Stack Overflow](#)

The **TIOBE Index** is an indicator of a programming language's popularity based on quantity of search queries across 25 search engines. Currently, Python is #3 in the TIOBE Index, preceded only by Java and C.

Because of its surge in popularity, TIOBE named Python the Language of the Year in 2018, 2010 and 2007. In June 2019 Python's rating jumped by 2.77%, once again reaching an all time high in the TIOBE index. If it continues at this pace, TIOBE predicts Python will overtake Java and C in just 3-4 years' time.

3. Python tops the list of the most in-demand programming languages

Most in-demand programming languages of 2019

Based on Indeed.com job postings in the USA - Feb 1, 2019

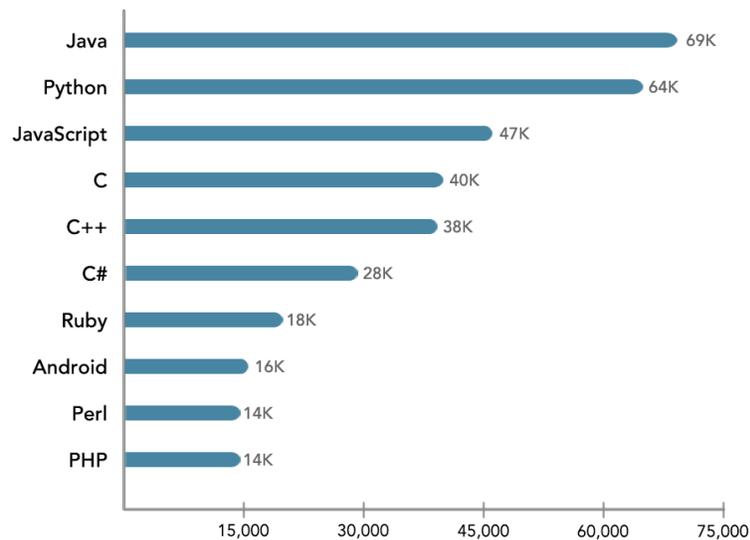


Image Source: CodingNomads

4. Python is easy to read, write, and learn

Python was built with the goal of getting rid of the complex and keeping only the necessary. Because of this, Python is easy to read, write, and most importantly, learn

WP Engine surveyed 909 developers across the US on which programming language is the easiest to learn. Python came in second place after HTML, and was noted for its “high readability and simple syntax that is easy to learn.” The article also states that, “for new coders especially, the consistency and simplicity of Python both contribute to its ease of use and approachability.”

Python is also an interpreted programming language. This means that unlike compiled languages like C++ or Java, you can run each line of code as soon as you finish writing it and see the results immediately. This is especially great for new programmers because you can get instantaneous feedback instead of waiting for the whole program to compile and run before you figure out that you forgot one character somewhere!

5. Python developers make great money

Python developers are some of the highest paid developers in the market, particularly in data science, machine learning and web development.

On average, a **Python developer earns \$110,021 per year** in the US. Additionally, the average salaries from 2017 to 2019 show that Python is consistently the second highest paying language.

Average Python Developer Salary Trends in Comparison to Other Programming Languages | 2017-2019

Skill	Average salary 2017	Average salary 2018	Average salary 2019
Ruby	US\$108,065	US\$115,005	US\$114,600
Python	US\$103,492	US\$116,379	US\$110,021
C++	US\$101,561	US\$108,123	US\$105,716
Java	US\$99,951	US\$112,592	US\$112,017
Perl	US\$99,857	US\$111,928	US\$109,099
JavaScript	US\$95,902	US\$103,503	US\$103,331
C#	US\$94,653	US\$101,715	US\$102,232
PHP	US\$93,169	US\$94,690	US\$96,231
ASP.NET	US\$92,150	US\$95,551	US\$96,467

Image source: [Daxx.com](#)

Indeed's salary calculator gives an even larger figure—a whopping \$123,642 per year. It's not exactly surprising to see Python rank so highly, given that it ranks second among the most in-demand programming language in 2019.

6. Python has an incredibly supportive community

While programming is often misinterpreted as a solo-sport, one of the greatest tools a programmer will ever have is the support of their community. Thanks to online forums, local meet-ups, and the open source community, programmers continue to learn from and build on the success of their predecessors.

Stack Overflow

Stack Overflow is a programming question and answer platform critical for all developers when they are stuck, or wanting to share wisdom with the community. According to **Stack Overflow's annual software developer survey**, Python is the most wanted programming language, and

tagged in 1,188,566 questions. This demonstrates an incredibly robust and active community for current and aspiring Python developers.

Stack Overflow's 2019 Most Wanted Programming Languages



Image source: [Stack Overflow](#)

GitHub

GitHub is where developers store project code and collaborate with other developers. With **over 1M repositories on GitHub** and over 59K users committing or creating issues in these repositories, Python is one of the largest developer communities on GitHub today.

Open source packages

Python offers a rich ecosystem of packages held within **The Python Package Index (PyPI)**. With Python, users can build modules for the ever-growing PyPi library. Users of this giant third party package library can find tools to help them with projects ranging from AI to web development and more.