What to do on Day One

College:	CUNY School of Professional Studies
Program:	Online Bachelor's Degree in Information Systems
Course Name:	Software Application Programming I
Course Code:	IS 210

A brief introduction to the next fifteen weeks of learning adventure!

Welcome Students!

Welcome, students, to **IS 210: Software Application Programming I**. I'm excited to share the fun and creative world of application programming with each of you.

Throughout the next fifteen weeks of courses, you will be regularly challenged to think about problems in new and different ways. The tools and techniques you learn here can help you in your career, at home, and as a lifelong learner. All you need to start is an open mind.

An Appetite for Knowledge

Before we delve too deeply, I'd like you to just take a moment and think about your kitchen. Think about all the tools and ingredients sitting there, just waiting to be turned into something delicious.

Will it be anchovy pizza? Perhaps some roasted vegetables are in order. Or are you in the mood for a challenge like soufflé?

Your kitchen is filled with numerous tools and ingredients that could all be counted individually but it represents so much more than the sum of its parts; it's a reservoir of creative possibility. How many new treats could your kitchen produce if you set your mind to it?

Python is just another kind of kitchen. In this course we'll learn the basic tools and techniques of cooking with code, from melon ballers to blenders and `integers` to `if`'s. Just as with any culinary instruction, I encourage you to bring and try your own recipes; make the dishes you want to eat and the software you want to use.

Expectations

Warning

Success in this course will require a personal commitment of approximately eleven hours per week with eight hours or more of that time devoted to completing homework assignments throughout the week.

I'll be frank with you: this will be a difficult course for some of you but, perhaps, for reasons different from what you expect. Success at programming requires both practice and extreme attention to detail. You will be regularly challenged in both of these aspects and it is expected that you might need some regular interaction with your fellow students and your instructor to discuss problems of particular complexity. In prior semesters, students who waited until the end of the week to begin their homework almost universally found it difficult to achieve a passing grade. As the great Thufir Hawat said,

> "The first step in avoiding a trap is knowing of its existence!"

I urge you to take this advice to heart and plan a weekly schedule that includes an early start to homework activities.

Graded assignments will form the primary means of instruction. Completed assignments from one week will sometimes be used as the basis for work in a following week and examples from student work may be used to illustrate key points during discussions. Because of this fact, this course does not award credit for late work and you are encouraged to review the lateness policy in-detail in the **Syllabus**.

Each week, you will visit the course site and that week's content folder where I'll give an overview of the concepts we'll be covering in that week's reading and assignments. You will then have an opportunity to practice honing your new skills by completing the assignments and completing quizzes that reinforce course materials. These assignments will be submitted to a public repository that could later become part of your software development portfolio.

For more information regarding course content or scoring policies, please see the **Syllabus**.

Communication

As a fully-online course, the this Blackboard Course site will serve as the primary means of communication for the following content types:

- major announcements
- lesson content
- quizzes and exams
- assignments
- extra credit discussions
- help with tools or technologies

Comments specific to assignments will appear on GitHub, where assignments will be submitted for testing, and in graded submissions on Blackboard.

Private communications will be conducted over e-mail. Students are expected to use their SPS e-mail accounts for **all** e-mail communications. E-mails sent from other accounts may not receive responses. For more information on how to use your CUNY SPS e-mail account, contact the CUNY SPS Help Desk.

In additional to e-mail communication, there will also be opportunities for synchronous sessions where we'll go over emerging topics and challenges. If you require additional support, private appointments may be made in upon request. Requests must be made in advance of the desired appointment time. Please see the **Instructor Bio & Contact** page for more details regarding private consultations.

Course Schedule

The standard course week will commence at 9:00am each Monday and run until 9:00am the following Monday. Weeks impacted by holidays or other events will have the course schedule and load adjusted accordingly.

Please take a moment to read the full academic calendar at https://sps.cuny.edu/academic_calendar.html and familiarize yourself with such key dates as the final exam period, term end, and last day to withdraw.

More details regarding the course schedule may be found in the *Syllabus*.

Lessons and Assessment

This course uses five major categories of assessment which, when combined, are weighted to represent your final grade. These categories are:

- Homework
- Quizzes
- Mid-Term
- Final

- Course Project

Complete details regarding the lessons and assessments can be found in the *Syllabus*.

Extra credit will also be available in the form of Discussion Boards. Please read the extra credit policy available in the **Getting Started** section of the course.

Tips for Success

1. Become oriented with your tools quickly. This goes for both development tools as well as the learning tools we'll use in this course like Blackboard. Sometimes it's worth losing half an hour to read through some help pages or a how-to manual. The confidence you gain in understanding how your tool functions will remove doubt later on when other bugs or issues manifest.
2. Be fearless. Take ownership of the computer as your tool and use it with a sense of responsibility and purpose. Click things just to find out what they do. Read the tooltips and manual pages. If something's not clear, try it out and see if you can glean its function from the result. Have faith in your own ability to clean up any messes you might make. Don't be afraid of making a turkey just because you're worried you might not get it cooked all the way through. Take the risk and improvise if it all comes out pink; what you learn in the process of improvisation can sometimes be much more valuable than if everything had gone according to plan.
3. Don't try to get it right the first time. Application programming is a perfectionist's game, without a doubt, but it does so through repetition and refinement. The first cut of any product is usually quite rough but it can give you a sense of whether or not you've started in the right direction. Test early and test often, before you've wasted hours on a product that was heading in the wrong direction. This course will enable you to bounce your code off our our automated tests as often as you'd like during the grading window; take advantage of that. Submit your proof of concepts just to see if they work, then refine them down till all the tests pass and you're happy with the work. Instead of thinking of each revision as a failed attempt, think of it as an improvement; a work with fifty revisions is sometimes a lot more trustworthy than one with just two.
4. Engage in short bursts of activity. If you can fit it into your schedule, try to access the site and the materials once or twice each day. This will not only help with retention but it will give your mind time to relax and assimilate the material. This rule is also true for completing assignments. If you find yourself stuck on a particular problem, take an active break. A common issue for developers is getting entrenched on thinking about a particular problem in a particular way. We all need shakeups from time to time just to give us another angle of attack. A walk or a meal can do wonders for that. If I had a dollar for every time I had a "Eureka!" moment standing in a subway car or out on a walk in the park, I could quit my day job.
5. Collaborate with each other. Contrary to popular belief, programming is a highly social activity. Some projects have hundreds of thousands of individual contributors whose work builds upon each others to form incredible pastiches of form and function. Within this course you have an authoritative resource in me, certainly, but you also have each other and you should take advantage of that. The discussion boards are available for general discussion of course topics and for comments specific to code I encourage you to use GitHub where you'll quickly gain a reputation for being a helpful coder.
6. Believe in yourself; every one of you can be an awesome Python programmer. Programming is just telling a computer to do something in a language it understands. Each of you has the ability to speak that language; you just need to find the right words. If you need help, don't get frustrated, just ask; we're here to help you succeed.

Resources

Syllabus

The syllabus should function as your primary resource for answering any questions you may have about the course. In it you will find such details as required texts, software, grading and classroom policies, and the course schedule. If you have not already done so, please take a moment to read it in its entirety.

The syllabus is available in the lefthand navigation of the course site under **Getting Started**.

Q&A

For general questions regarding this course or online education, please use the **Ask Your Instructor** discussion board.

Development Environment

As this is a practical programming course, you will be quickly introduced to a number of new technologies and concepts. For this reason, the first few weeks are sometimes the most challenging in the course. If you are having difficulty early on, please don't hesitate to reach out. We introduce these tools early in the hopes of benefiting from their use throughout the course. The pace of adoption will slow shortly after this initial period.

Getting Started

So now that we've covered most of the basics it's time to get cooking! Be certain you've read the **Syllabus**, then head over to the **Discussion Board** and post a little about yourself in the **Getting Acquainted** forum. After that, click the **Weekly Materials** link and start in on Week 01.

Oh, and be certain to set-up your own avatar in Blackboard, if you've not already done-so. For help in creating your avatar, visit:

http://screencast-o-matic.com/watch/c2V3cgniFS

Closing

In closing, I want to thank each of you for getting this far. I love having the opportunity to share my craft with engaged learners and look forward to a great semester of creative play.