## CS-135 CDU1 – Spring 2014

# Programming for Non-CS Majors

This is a LASC course and can be taken to satisfy the *Quantitative Reasoning* requirement.

### Credit Hours

3 credits (3 lecture hours/week)

### Catalog Course Description

*Introduction to programming. Emphasis on practical skills, working with data sets, doing analysis and visualization. No prior programming experience required.*

### Instructor

Dr. Karl R. Wurst

See [http://sharepoint.worcester.edu/faculty/kwurst/](http://sharepoint.worcester.edu/faculty/kwurst)   
for contact information and schedule.

### Meeting Time and Location

Thursdays, 3:30pm-6:00pm

ST 105

### Textbook

*Python Programming in Context, 2nd Edition*  
Bradley N. Miller and David L. Ranum  
Jones and Bartlett, 2014  
ISBN-13: 9781449699390

<http://www.jblearning.com/catalog/9781449699390/>

### Required Materials

In addition to the textbook, to successfully complete this course you will need:

1. **Computer:** Our class will be held in a computer lab, and you will be able to use one of the computers in the lab for in-class work, but you will need access to a computer that you can use for homework. That computer can be your own computer – either a laptop or a desktop – or some other computer that you can use on a regular basis (at the library, your school, or somewhere else). The brand and operating system (Windows, Mac OS X, Linux) is unimportant – the software we will be using runs on all major operating systems and can be downloaded for free. The computer you plan to use will need:
   1. **Web Browser** – You will need a web browser to access the Blackboard course site, Piazza, and QuizStar (see Internet Access below.)
   2. **PDF Reader** – All course materials will be distributed as PDF files, and all graded work will be returned electronically as PDF files. You must have software to read PDFs.
   3. **Python 3** – The Python 3 software is Free and Open Source Software and can be downloaded from <http://www.python.org/download/> and is available for Windows, Mac OS X and Linux operating systems. If you are using your own computer, you will be able to install Python on your computer. If you are using some other computer, you will be able to run Portable Python off of a USB stick.  
        
      If you are going to be using your own laptop, you may want to bring it to class to use so that you can install the software in class, and work in the same environment both in- and out-of-class.
2. **USB Stick/Flash Drive:** You will need some kind of storage device to keep your class files on. It does not have to be high capacity – 1GB should be plenty if you are only going to use it for this class, and you might even be able to get by with less space.
3. **Internet Access:** You will need internet access for access to:
   1. **Blackboard** – All course materials and announcements will be made available through the course site on Blackboard. Students will be required to use Blackboard as the course tool and be familiar with uploading files.
   2. **WSU Gmail** – You must check your WSU Gmail account on a regular basis. All communications to the class, such as corrections to problem sets or changes in due dates, will be sent to your WSU Gmail account.
   3. **Piazza** – This is an online service that lets you ask (and answer other students’) questions about the course assignments, materials, and topics.
   4. **QuizStar** – We will be using this service for online quizzes.

### Course Workload Expectations

You should expect to spend, on average, at least 6 hours per week *outside of class* during the semester reading the textbook, studying, doing assignments, working on projects, and practicing in order to master the concepts and use of the language and modules covered in the course. (See *Definition of the Credit Hour*)

### Definition of the Credit Hour

Federal regulation defines a credit hour as an amount of work represented in intended learning outcomes and verified by evidence of student achievement that is an institutional established equivalence that reasonably approximates not less than –

1. One hour of classroom or direct faculty instruction and a minimum of two hours of out of class student work each week for approximately fifteen weeks for one semester or trimester hour of credit, or ten to twelve weeks for one quarter hour of credit, or the equivalent amount of work over a different amount of time; or
2. At least an equivalent amount of work as required in paragraph (1) of this definition for other academic activities as established by the institution including laboratory work, internships, practica, studio work, and other academic work leading to the award of credit hours.

### Prerequisites

This course has a math placement score prerequisite of 3 or above. This prerequisite can also be met by having taken a college-level math course for credit. A partial list of those math concepts that I expect you to understand is:

* Fractions, decimals, percentages, rates, ratios, proportions, word problems, graphs and tables, signed numbers, variables, expressions, functions.

*If you do not have this background, you should not take this course.*

### Course Objective

From having taken *one* course in programing, a student – any student ([CS] major or not) – should walk away from [that] course and somewhere later in his or her career address a problem by thinking, “Hey, I’ll just write a program to solve that.” – Punch and Enbody, *The Practice of Computing using Python,* 2011, p. xxi.

### Course-Level Student Learning Outcomes

After successful completion of this course, students will be able to:

* Understand the basic syntax and semantics of the Python programming language. (Emphasis)
* Model an abstraction as a Python program. (Emphasis)
* Manipulate integer and floating point values. (Mastery)
* Code, test and debug simple procedural programs given a program design/specification. (Emphasis)
* Write simple functions to access and manipulate data. (Emphasis)
* Accept user input from the keyboard, as command-line arguments and read data from a text file. (Emphasis)
* Generate output to the screen and to a text file. (Emphasis)
* Trace, code, test, and debug simple recursive functions and procedures. (Introduced)
* Manipulate and compare strings. (Emphasis)
* Store and access data in lists and dictionaries. (Emphasis)
* Perform simple exception handling. (Introduced)
* Write code representing conditional and repetition control structures. (Mastery)
* Trace code to determine the behavior of the code (Emphasis)
* Draw a diagram representing memory and the object/reference relationships (Emphasis)
* Write simple unit tests (Introduced)
* Use a version control system to commit and revert changes on a single-user repository (Introduced)
* Document code for other programmers, users of code, and users of programs (Emphasis)
* Write programs that can display data in a variety of formats including tables, graphs and charts. (Emphasis)
* Find and evaluate third-party program modules and incorporate them into their own programs. (Introduced)
* Understand and modify code written by others. (Emphasis)

### LASC Student Learning Outcomes

This course will contribute to the students’ ability to:

* Demonstrate effective oral and written communication.
* Employ quantitative and qualitative reasoning.
* Apply skills in critical thinking.
* Apply skills in information literacy.
* Understand the roles of science and technology in our modern world.
* Understand how scholars in various disciplines approach problems and construct knowledge.

### LASC Quantitative Reasoning Objectives

This course will:

1. Acquaint students with formal systems, procedures and sequences of operation.
2. Strengthen students' understanding of variables and functions.
3. Apply mathematical techniques to the analysis and solution of real-life problems.
4. Emphasize the importance of accuracy, including precise language and careful definitions of mathematical concepts.
5. Understand both the underlying principles and practical applications of one or more fields of mathematics.
6. Strengthen understanding of the relationship between algebraic and graphical representations.

### Course Topics

The course outline will be covered on a best-effort basis, subject as always to time limitations as the course progresses.

* How computers represent and process information
* Algorithms
* Strings Operations and Comparisons
* Functions
* Lists and Dictionaries
* Text Processing
* Input/Output
* Exceptions
* Testing and Debugging
* Programming Languages vs. Natural Languages
* Recursion
* Translation, Compilation, Interpretation
* Syntax and Semantics
* Testing and Debugging
* Simple Version Control
* Documentation
* Variables, Assignment, Operators
* Control Flow
  + Repetition
  + Selection
  + Tracing code

### Course Philosophy

During this semester we will be learning to design, write and test programs that can be used to manipulate data in a number of ways. The emphasis will be on solving real-life data analysis problems in a variety of disciplines. We will be using the Python programming language which has the advantage of being (relatively) easy to learn and has a large collection of modules contributed by other users. We will spend some of our class time working on the computer trying out new concepts and techniques. This will give us all a chance to learn new material with the support and help of the instructor and your classmates. Larger projects completed outside of class will lead toward more complete mastery of the material. Finally, we will work on individual or group final projects, developing a program to solve some problem in our own disciplines and learning from each other’s experience on these projects.

You are encouraged to help each other out, in and out of the classroom, as long as you do your own work. (See Academic Conduct below.)

### Instructional Methods

This class will not be a traditional “lecture” class, and will incorporate some teaching methods that may be unfamiliar to you.

### *Flipped Classroom*

You will be given reading assignments in the textbook and from other sources, and you will be expected to complete a short quiz on the reading assignment(s) before the class meeting. Your answers from the quiz will be used to tailor what we do in the classroom. If everyone understands a topic or concept, we will not spend time on it in class. If a large portion of the class is having difficulties with a topic or concept, we will work on it in class to make sure everyone understands.

Your quizzes will be graded on the effort you put into explaining your answer. If you do not complete the quiz by the specified time, you will not receive any points for that quiz.

This flipped classroom technique means that I will spend much less time “telling” you things that you already understand from the reading, and more time on the concepts that you find difficult.

### *Pair Programming*

Often, during class time, you will be working on hands-on assignments. For these assignments, you will work in a pair with one of your classmates. One of you will type while the other makes suggestions, watches for errors, reads the assignment, and thinks ahead. You will switch roles frequently during the lab session.

You will be randomly assigned a new partner each lab. This will allow you to get to know the other members of the class, work with partners of different abilities and programming styles, and develop relationships that may extend beyond to classroom.

### Grading Policies

There will be in-class work, two exams, programming projects, and a final project and presentation. Your grade in the course will be determined as follows:

* Attendance and In-Class Lab Work 20%
* Homework Assignments and Pre-class Quizzes 20%
* Exams (2) 20%
* Programming Projects 20%
* Final Project and Presentation 20%

Grading Scale

93-100% A

90-93% A-

87-90% B+

83-87% B

80-83% B-

77-80% C+

73-77% C

70-73% C-

67-70% D+

63-67% D

60-63% D-

less than 60% E (Failure)

Each range includes the lower value, but not the upper value. For example, the range of 80 to 83 includes all grades up to, but not including 83. The highest range does, however, include 100%.

### Attendance and In-Class Lab Work

You are expected to attend every class. Class time will be divided between lecture and hands-on in-class work. Past experience has shown that students who do not attend class do not do as well on exams and projects.

The in-class lab work will give you a chance to apply the material from the lecture in an environment where you can benefit from the help of the instructor and your classmates. Once you have developed a level of comfort and confidence with the material, then you can be expected to apply it on larger projects outside of class.

Missed in-class work cannot be made up after the fact. If you know that you will have to miss class, let me know beforehand and we can make arrangements for you to do the in-class work at another time.

### Homework Assignments and Pre-class Quizzes

We will use an online site that contains assignments you will complete to go along with your reading and will help you master the material. These assignments provide immediate feedback, so that you can be sure that you understand the material. You will also complete problems in the textbook. Finally, you will also take quizzes before class that will guide the development of the in-class sessions.

### Exams

There will be two exams given during the semester. Both exams will be given during a regular class period.

Exam 1 is tentatively scheduled to be given during class on 20 February 2014.

Exam 2 is tentatively scheduled to be given during class on 3 April 2014.

### Programming Projects

The programming projects will give you a chance to apply the material to larger tasks. These projects will require you to interpret the problem description, do some planning and design, possibly research and evaluate third-party modules, implement your program, and test and debug it.

### Final Project and Presentation

Near the end of the semester we work on individual or group programming projects. You will work on some data from your particular major or area of interest. With the instructor’s guidance you will determine what analysis you wish to perform on your data, and develop a program to perform that analysis.

During the scheduled final exam period, you will present your final project to the class. Our Final Exam period is scheduled for Thursday, 8 May 2014.

### Late Submissions

**Late assignments are not accepted**. This is so that we may discuss the assignment soon after the due date.

I understand that occasionally circumstances beyond your control (work, family emergencies, illness, etc.) may prevent you from completing an assignment on time. If that is the case, please e-mail me before the assignment is due and I will certainly consider your case.

### Deliverables

All work must be submitted electronically. The submission method, due date and time will be given on the assignment. The submission date and time will be determined by the timestamp on the website used to submit the work.

**Please do not submit assignments to me via email.** It is difficult for me to keep track of them and I often fail to remember that they are in my mailbox when it comes time to grade the assignment.

It is strongly recommended that you keep copies of your projects. Students are responsible for reproducing any lost work including unreadable files.

Graded assignments (with comments and solutions) will be returned to you electronically. *Please make sure that you review the comments and solutions provided, so that you can improve your future assignments.*

### Academic Conduct

Each student is responsible for the contents of the readings, discussions, class materials, textbook and handouts. All work must be done independently unless assigned as a group project. You may discuss assignments and materials with other students, but you should never share answers or files. ***Everything that you turn in must be your own original work, unless specified otherwise in the assignment.***

All work must be done individually. Students may help each other understand the language and the development environment but students may not discuss actual solutions, design or implementation, to their programming assignments before they are submitted or share code or help each other debug their programming assignments. The assignments are the primary means used to teach the techniques and principles of computer programming; only by completing the programs individually will students receive the full benefit of the assignments. If you are looking at each other’s code before you submit your own, you are in violation of this policy. All students that collaborate on an assignment will receive a 0 for that assignment if that is the first offense. Students that collaborate a second time will receive an ‘E’ for the course.

Students may not use solutions to assignments from any textbooks other than the text assigned for the course, or from any person other than the instructor, or from any Internet site, or from any other source not specifically allowed by the instructor. If a student copies code from an unauthorized source and submits it as a solution to an assignment, the student will receive a 0 for that assignment.

**Any inappropriate sharing of work or use of another's work without attribution will result in a grade of zero on that assignment for all parties involved. If you do so a second time, you will receive an “E” for the course.**

Academic integrity is an essential component of a Worcester State education. Education is both the acquisition of knowledge and the development of skills that lead to further intellectual development. Faculty are expected to follow strict principles of intellectual honesty in their own scholarship; students are held to the same standard. Only by doing their own work can students gain the knowledge, skills, confidence and self-worth that come from earned success; only by learning how to gather information, to integrate it and to communicate it effectively, to identify an idea and follow it to its logical conclusion can they develop the habits of mind characteristic of educated citizens. Taking shortcuts to higher or easier grades results in a Worcester State experience that is intellectually bankrupt.

Academic integrity is important to the integrity of the Worcester State community as a whole. If Worcester State awards degrees to students who have not truly earned them, a reputation for dishonesty and incompetence will follow all of our graduates. Violators cheat their classmates out of deserved rewards and recognition. Academic dishonesty debases the institution and demeans the degree from that institution.

It is in the interest of students, faculty, and administrators to recognize the importance of academic integrity and to ensure that academic standards at Worcester State remain strong. Only by maintaining high standards of academic honesty can we protect the value of the educational process and the credibility of the institution and its graduates in the larger community.

**You should familiarize yourself with Worcester State College’s Academic Honesty policy. The policy outlines what constitutes academic dishonesty, what sanctions may be imposed and the procedure for appealing a decision. The complete Academic Honesty Policy appears at:** <http://worcester.edu/Academics/Shared%20Documents/AcademicHonesty.aspx>

**If you have a serious problem that prevents you from finishing an assignment on time, contact me and we'll come up with a solution.**

### Code of Conduct/Classroom Civility

All students are expected to adhere to the policies as outlined in the University’s Student Code of Conduct.

### Getting Help

If you are struggling with the material or a project please see me as soon as possible. Often a few minutes of individual attention is all that is needed to get you back on track.

By all means, try to work out the material on your own, but ask for help when you cannot do that in a reasonable amount of time. The longer you wait to ask for help, the harder it will be to catch up.

I am here to help you understand the material and be successful in the course.

### Contacting Me

You may contact me by email ([Karl.Wurst@worcester.edu](mailto:Karl.Wurst@worcester.edu)), telephone (+1-508-929-8728), or see me in my office. My office hours are listed on the schedule on my web page (<http://sharepoint.worcester.edu/faculty/kwurst>) or you may make an appointment for a mutually convenient time.

If you email me, please include “[CS-135]” in the subject line, so that my email program can correctly file your email and ensure that your message does not get buried in my general inbox.

If you email me from an account other than your Worcester State email, please be sure that your name appears somewhere in the email, so that I know who I am communicating with.

You may expect that I will get back to you within 24 hours of your email or phone call (with the exception of weekends and holidays), although you will likely hear from me much sooner.

### Schedule

Our first class will be on Thursday, 23 January 2014.

Our last class will be on Thursday, 1 May 2014.

We will not have class on 6 March 2014, because I will be away at a conference, and on 20 March 2014, because of WSU’s Spring Break.

Final presentations will be given during our scheduled final exam period – Thursday, 8 May 2014.

### Americans with Disabilities Act

Worcester State University and this instructor are committed to the full participation of all students, and will provide accommodations for any student with documented disabilities who are registered with the Disability Services Office (DSO). Please contact the instructor as early as possible to discuss necessary accommodations. All information regarding disabilities will be treated with confidentiality. The DSO is located in the Administration Building, Room 131 and can be reached by phone (508-929-8733) or email ([dso@worcester.edu](mailto:dso@worcester.edu)).

### Tutoring Services/Academic Success Center

Tutoring Services are offered through the Academic Success Center (ASC).  The ASC is located on the first floor of the Administration building, A-130.  Tutoring services are provided to students FREE of charge.  Students seeking academic assistance should visit the center as soon as possible or contact the Tutoring Coordinator at 508-929-8139

### The Writing Center

The writing center provides free assistance to students in the areas of research and writing.  It is located on the third floor of the Sullivan Academic Building, S306.  To schedule an appointment, please call 508-929-8112 or email the Center at [writingcenter@worcester.edu](mailto:writingcenter@worcester.edu).  To find out more information about the Writing Center including the Center’s hours and the Center’s Online Writing Lab, visit their website at [www.worcester.edu/writing](http://www.worcester.edu/writing)

### Worcester State Library

Worcester State Library has access to many articles through online databases including J-STOR. In addition many articles and book chapters are available to students through Inter-Library Loan (ILL). With a little planning, ILL expands your ability to get credible information sources about topics you pursue in your course work. Finally WSU students are free to use many of the library resources within the consortium. Given all of these resources it is extremely unlikely that you should have to pay for access to individual articles. Please work with the reference librarians to find the appropriate way to access materials you need. You have already paid for these resources through your fees—please make use of them.

### Student Responsibilities

* Attend every class. Past experience has shown that students who do not attend class do not do as well on exams and assignments.
* Come to class on time
* Contribute to a classroom atmosphere conducive to learning for everyone in the class: by asking/answering questions, participating in class discussions and coming to the class prepared. Don’t be just a warm body in the class
* Seek help when necessary
* Start assignments as soon as they are posted. Do not wait until the due date to seek help/to do the assignments
* Make use of the academic success center (see below)
* Expect to spend at least 2 hours outside of class (reading, doing homework assignments etc.) for every hour spent in class
* Each student is responsible for the contents of the lectures, the textbook, handouts, and homework assignments.