

Course title: Introduction to Computer Programming, Summer 2012

Instructor: Radu Paul Mihail, 312 Davis Marksbury Building, Email: r.p.mihail@gmail.com

Class meeting times and location:

- **Lecture:** F Paul Anderson Tower, Room 267, MWF at 9:10 AM to 10:10 AM
- **Lab:** Ralph G Anderson, Room 103, TR at 9:10 AM to 10:10 AM

Office Hours: MWF, 312 Davis Marksbury Building at 10:30 AM to 11:30 AM and by appointment.

Required Textbook: Python Programming: An Introduction to Computer Science , Second Edition by John Zelle, Publisher: Franklin, Beedle & Associates, Copyright Year: 2010, ISBN: 978-1-59028-241-0.

Software: The main programming environment will be Python. This software is free and available from www.python.org. The software will be available on computers in many labs on campus and is available to all students in the class for free download. Make sure that you get version 3.x of Python, not version 2!

Course Description: This course teaches introductory skills in computer programming using an object-oriented computer programming language. There is an emphasis on both the principles and practice of computer programming. Covers principles of problem solving by computer and requires completion of a number of programming assignments. Lecture meets three times per week; lab sections meet twice per week. This course covers introductory skills in computer programming using Python. The course assumes NO programming experience. You should already have basic computing skills, like being able to copy files from one place to another, renaming files, making folders. You should expect to spend at least 3 hours outside class for every hour in class on this course, on average.

Learning Outcomes: Students will understand basic programming terminology and techniques. More specifically students will:

1. acquire an understanding of computer architecture and data representations (variables, representation of numbers and character strings)
2. learn basic algorithmic problem-solving techniques (decision structures, loops, functions)
3. be able to use and understand objects used in programming
4. be able to design, document, implement and test solutions to programming problems

Course Outline:

- No class on Wednesday July 4th.
- Week 1:
 - Syllabus, Chapter 1 (Basics)
 - Chapters 1 and 2 (Basics, Simple Programs)
 - Lab 1: Introduction to Python, Knowledge Survey, Lab 2: Meet your team
- Week 2:
 - Chapters 2 and 3 (Simple Programs, Computing with Numbers)
 - Chapters 3 and 4 (Numeric Operators and Data Types, Graphics)
 - Program 1 handed out
 - Lab 3: Introduction to Testing, Lab 4: Introduction to Design
- Week 3:
 - Chapters 4 and 5 (Python Graphics, Lists and Files)
 - Chapter 5 (Strings, Lists and Files)
 - Program 1 due, Program 2 handed out
 - Lab 5, program 1 design & test cases, Lab 6: Python Graphics
- Week 4:
 - Chapters 5 and 7 (lists, files, ifs)
 - Program 2 due, Program 3 handed out
 - Review
 - Midterm exam
 - Lab 7, program 3 design & test cases and Lab Exam
- Week 5:
 - Chapters 6 and 7 (ifs, functions)
 - Chapters 6 and 8 (functions, loops)
 - Program 3 due, Program 4 handed out
 - Lab 8, program 4 design & test cases
- Week 6:
 - Chapter 8 (loops)
 - Review, Chapter 9 (simulations, random numbers)
 - Program 4 due, Program 5 handed out
 - Lab 9, program 5 design & test cases
- Week 7:
 - Chapter 11 (arrays)
 - Part of Chapter 10
 - Program 5 due
 - Lab 10 and 11
- Week 8:
 - Chapter 10 (objects/OOP), Review
 - Review
 - Final Exam (comprehensive)
 - Lab Review and Lab Exam 2

Assessment:

The grade for this course will be calculated as follows:

- Attendance and pop quizzes at labs and/or lectures: 5%

- Lab assignments: 10%
- Programming assignments: 35%
- Midterm exam: 20%
- Lab tests: 5% each, both: 10%
- Final Exam (comprehensive):20%

Grades will be assigned according to the following scale:

90-100%	= A
80-89.99%	= B
70-79.99%	= C
60-69.99%	= D
Below 60%	= E

Exams:

- Midterm exam: TBD
- Final exam: Thursday August 2, 2012

What to do if you miss...

- a lecture - find out what the material covered was, read the book, borrow someone's notes, find out what any announcements or assignments were. If attendance was taken and you have a documented excuse as described in the attendance policy, contact your instructor within one week of your absence.
- a lab - the points for attendance and for the demonstration can not be made up unless you have a documented, excused absence. If you have an excuse, make sure you contact your instructor within one week. You **MUST** still turn in your labwork by the deadline for the lab.
- a test - if you know ahead of time you must miss a test, contact your instructor and make arrangements for an alternate time. If circumstances force you to miss an exam unexpectedly, you **MUST** contact your instructor within a week after the test, in order to have a chance to be allowed to make the exam up.
- a deadline on an assignment - see the late policy. Programs are accepted up to 5 school days late, with penalty.

Due dates

- Lab Assignments - Every lab assignment will be posted on the page several days before the lab period when it will be due.

- Late Policy for Labs - The individual part of the lab will be due the night before the lab meets. It must be submitted by midnight. If it is submitted later than midnight, but before the actual lab period, it will be accepted with a late penalty. It will not be accepted after the individual's team has submitted their team product. There is a deadline for the team work, which will usually be during the lab period. There will be links on the lab assignment page for submitting individual and team lab assignments.
- The electronic submission of program assignments will be done via a script on the class web page; it can be done as soon as you have your program ready. It is not acceptable to email your submission. Each assignment will have a list of items to turn in at the bottom of the page.
- Late Policy for Programs - Programming assignments may be turned in late, but they lose 10 percent of the points possible per school day late, up to 5 school days late. That is, work that is one school day late loses 10 points out of 100, work that is two school days late loses 20 points, and so on.
- Free days - You have 2 "free days" to extend any of the programs' due date. They can be used consecutively on one program or separately.
- Some assignments may have bonus parts. If a program is turned in late, it is NOT eligible for any bonus points. This means if you are tempted to turn in an assignment late so you can finish the bonus, don't. The bonus will not be counted. Even if you use the 2 available days to extend your due date, the bonus will NOT be counted in that case.
- Programs will not be accepted more than 5 school days late. This means that work turned in more than 5 school days after the deadline will get a grade of zero. This allows your instructor to grade most of the submitted programs at the same time, increasing consistency and improving return times for assignments. If you have an excused absence, some deadline extension will be allowed, determined by your instructor, contact him immediately.

Late assignments will not be accepted once the graded work has been returned to the class! Programming projects require time to prepare and a way to do POORLY on them is to wait to start work the evening before they are due. A program need not be fully functional to receive some credit, but we would much rather see a working program turned in late than one on time which does not work. A program that does not run because of syntax errors should not be turned in. It will NOT be graded; you will be asked to continue working on it, and there will be NO extension of the due date, meaning you will be losing points for the late penalty. This is another reason to start work on your assignments early - so you will have time to ask questions before the due date! This means that even if you turn in the program on time or early, if it does not run without errors, you will be losing points to the late penalty until you turn in one that DOES run! It is easy to accidentally delete a character or comment out a line at the last minute. After you submit your program, check to make sure what you just turned in DOES run! YOU are responsible for making sure your program as turned in will run cleanly!

File Formats in Electronic Submissions: You need to know the difference between a text file and a doc file and an HTML file and an executable file. Very often, the type that is

acceptable in electronic submissions is a text file (which include .py files) or a doc file (test cases). We will explain how to generate them. Occasionally other files may be required (e.g., a jpg file or a csv file if your program generates one or uses one for input). The assignment will specify what format each file should be in. It is YOUR responsibility to make sure the files are in the correct format. If we have time and find files of incorrect formats in a submission, we will try to notify you by email as soon as possible, BUT the submission will be counted as getting later, up until you submit the files in the correct format. Make sure you have the files you need and NOTHING else in your submission. When in doubt, show it to your instructor ahead of time! This also means check your email regularly!

Academic Honesty:

Students are expected to do their own work. Cheating is considered a serious offense by the University. Any form of "seeking an unfair academic advantage" is considered cheating.

If an assignment is designated as "cooperative learning" or "partner work", then you are allowed, encouraged, in fact required, to work with your partner or team. These are the ONLY students you are allowed to work with. Of course you can still ask questions of your instructor.

Any other assignment is individual work. That includes programs, lab tests, lecture tests, and quizzes.

Learning to program is an individual task; you are expected to do the programming assignments on your own. One person taking any part of another person's work (with or without their permission) and claiming it as his or her own is plagiarism and will not be tolerated. Any occurrences will be dealt with according to the University policy stated at:

URL: <http://www.uky.edu/Ombud/acadoffenses/index.htm>. This policy allows for a minimum penalty of zero on an assignment AND a warning letter in the student's file. Repeat offenders (in ANY class) face increasing penalties with each offense.

The only way to LEARN programming is to DO programming. You may think you have gotten the grade very easily by using someone else's work if the copying is not detected, but you have lost that much experience and will be that much further behind on the next assignment.

If you would agree that "he/she and I worked together" on a program, then we would consider it cheating.

If you and your roommate share a computer, be VERY careful. Your work MUST be your OWN. Discuss your design or algorithm or logic in GENERAL terms, but write your own design and your own code, your own implementation. Advice: if it makes you uneasy, then it's probably NOT ok. When in doubt, ASK your instructor before submitting work.

Do not show your source code to ANY other student.

It may seem an easy way to "show them how it's done" or "help them understand the problem". It is a recipe for trouble. It is a temptation to copy the other person's work without figuring out how to solve the problem.

It is just as dishonest to allow someone to represent your work as their own as to do the reverse. This also means YOU are responsible for making sure that your code does not accidentally fall into someone else's hands. Don't leave floppy disks or memory sticks or printouts in a lab; don't leave source code files on a hard drive somewhere. Be aware that files that you put on the local hard drive (C or D or E) in a computer lab on campus STAY there until they are deleted. They do NOT automatically go away when you log out! If someone else finds your code and turns it in, YOU are responsible too!

Do not post your code on the Internet.

This is an open invitation for someone else in the class to copy it and turn it in as theirs!

If you get help from a person who is not in the class, be extremely careful. Do not take code from anyone! Make sure the help you get is using the material covered in THIS class. You can be penalized in this situation also. If you work with a tutor, make sure you understand what the tutor is telling you. If they just "transplant" code into your program, you are being cheated of the understanding you need to do the next program and to take the Lecture tests. This is also considered cheating.

All programs will be checked by plagiarism detection software.

Withdrawing:

If you decide to leave the class, please do it officially. There is a date on the Academic Calendar past which you are not allowed to drop for academic reasons. We'd much rather give a W grade than an E. Don't just stop coming to class - you WILL get an E! Take care of your transcript!

All policies associated with this course are subject to revision. Reasonable notification will be provided to students prior to any major changes.

Extra Help: Do not hesitate to come to my office during office hours or by appointment to discuss a homework problem or any aspect of the course.

Attendance Policy: Please keep in mind that this is a summer course. The material that is taught over 16 weeks in a regular semester is condensed into 8 weeks for a summer course. Attendance is **CRITICAL** for doing well in this class and you are expected to show up for both lab and lectures.

Accommodation for Disabilities: If you have a documented disability that requires academic accommodations, please contact your instructor as soon as possible. In order to receive accommodations in this course, you must provide a Letter of Accommodation from the Disability Resource Center (<http://www.uky.edu/StudentAffairs/DisabilityResourceCenter/>). If you have not already done so, please register with the Disability Resource Center for coordination of campus services available to students with disabilities (Room 2, Alumni Gym, 257-2754, Jake Karnes, jkarnes@email.uky.edu)

Accommodations can be made for all parts of the course.

These letters are NOT retroactive! This means that we only make special arrangements for class activities after we receive the letter.