AUSTIN COMMUNITY COLLEGE

**DEPARTMENT OF COMPUTER STUDIES AND ADVANCED TECHNOLOGY**

Course Syllabus: COSC-1336-026 (3-3-1) – Programming Fundamentals I

Synonym 08862, Fall 2016

**Lecture:** PIN 309, TTh 7:05PM-8:50PM

**Lab:** PIN 309, TTh 7:05PM-8:50PM

**Instructor**: Carl Gregory

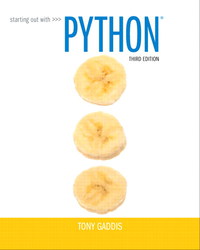
**Office Hours:** <http://www6.austincc.edu/directory/info.php?id=cgregory#OfficeHours>

**E-mail:** [cgregory@austincc.edu](mailto:cgregory@austincc.edu)

**Home page:** <http://www.austincc.edu/cgregory>

**Course Description:** Introduces the fundamental concepts of structured programming. Topics include software development methodology, data types, control structures, functions, arrays, and the mechanics of running, testing, and debugging. This course assumes computer literacy.

**Special note**: **this is not a “course in Python.”** It is a course that teaches the basic fundamentals of the art of computer programming, using the Python language. Other programming languages can be, and have been, used in the past. The objective is not to make you a Python wizard; it is to make you a good programmer in whatever language you choose.

**Pre-requisite:** This course requires the same math skills necessary for College Algebra. Students should either have taken or be currently enrolled in College Algebra or a course that requires College Algebra. TSI complete in reading.

**Approved Text and Teaching Materials:**

Tony Gaddis, *Starting out with Python, 3rd Edition, Pearson,*

ISBN:  *978-0-13-358273-4*

**Please note**: this is not the ISBN for the textbook that comes bundled with MyProgrammingLab. This curriculum does not use that online tool.

**Instructional Methodology:** This course will have 50% lecture and 50% laboratory. If the students are unable to finish the assigned lab work within the lab time, they will need to visit the CIS open labs.

**Course Rationale:** This is an entry level programming course designed to teach students the basic concepts of computer programming. The course will include designing, coding, debugging, testing, and documenting programs using a high level programming language. This course is intended to prepare students for a programming-oriented academic path. The course is included in several degree plans including:

* Associate of Applied Science – Computer Programming
* Associate of Applied Science – Web Programming
* Associate of Applied Science – Game and Visualization Programming
* Associate of Applied Science – Information Technology Application
* Associate of Applied Science – Software Testing
* Associate of Science – Computer Science

**Course Objectives / Learning Outcomes:**

|  |
| --- |
| 1. Demonstrate problem solving skills by developing and implementing algorithms to solve problems. 2. Derive problem specifications from problem statements. 3. Develop algorithms using modular design principles to meet stated specifications. 4. Create code to provide a solution to problem statements ranging from simple to complex. 5. Test and debug programs and program modules to meet specifications and standards. 6. Create programs that contain clear and concise program documentation. 7. Implement programs that use data types and demonstrate an understanding of numbering systems. 8. Incorporate both basic and advanced control structures appropriately into algorithms. 9. Demonstrate an understanding of structured design by implementing programs with functions, including parameter passing and value returning. 10. Implement programs using classes, including strings and files. 11. Implement algorithms using one-dimensional and indexed data structures. 12. Demonstrate an understanding of array searching and sorting algorithms by desk-checking and/or modifying algorithm implementations. 13. Design and implement simple classes. |

## SCANS (Secretary’s Commission on Achieving Necessary Skills):

Refer to <http://www.austincc.edu/cit/courses/scans.pdf> for a complete definition and explanation of SCANS. The following list summarizes the SCANS competencies addressed in this particular course:

|  |  |  |  |
| --- | --- | --- | --- |
| **RESOURCES**  1.1 Manages Time | **INTERPERSONAL**  2.3 Serves Clients/Customers | **INFORMATION**  3.1 Acquires and Evaluates Information  3.2 Organizes and Maintains Information  3.3 Uses Computers to Process Information | **SYSTEMS**  4.1 Understands Systems  4.2 Monitors and Corrects Performance  4.3 Improves and Designs Systems |
| **TECHNOLOGY**  5.1 Selects Technology  5.2 Applies Technology to Task  5.3 Maintains and Troubleshoots Technology | **BASIC SKILLS**  6.1 Reading  6.2 Writing  6.3 Arithmetic  6.4 Mathematics  6.5 Listening | **THINKING SKILLS**  7.1 Creative Thinking  7.2 Decision Making  7.3 Problem Solving  7.4 Mental Visualization  7.5 Knowing How to Learn  7.6 Reasoning | **PERSONAL SKILLS**  8.1 Responsibility  8.2 Self-Esteem  8.3 Sociability  8.4 Self-Management  8.5 Integrity/Honesty |

## Grade Policy:

This is a long description, primarily because it differs in some significant ways from the simple number/letter system we are all accustomed to since grade school. **It is important to read and understand this description in detail**.

Grades will be assigned based both on concepts and practical application. Homework exercises, , exams, quizzes, lab projects, and attendance are **Regularly Assigned Work** and are a *part* of the grade. An overall grade will be assigned according to some extensions to the usual categories most people are used to.

When submitting official letter grades to the college, these are the associated official descriptions:

A Excellent

B Good

C Satisfactory

D Minimum Passing

F Failing

In some courses, assigning those letters to particular scores can be fairly fluid. That cannot be so in this class. COSC 1336 is the prerequisite for all later programming classes. When an instructor records a letter grade of A, the teacher in the follow-up class is being told, “This student did *Excellent* work.” The student’s next instructor must be able to trust the first instructor’s description; if a student barely understands programming concepts but is *given* a B, all future teachers will expect Good performance and be disappointed. Over time, the first instructor’s evaluations become suspect, and finally ignored.

Accordingly, in this class, doing well on all the **Regularly Assigned Work** but just meeting expectations is a Satisfactory performance and earns a letter grade of C. Doing very well on all the regularly assigned work is Good, and earns a letter grade of B. Notice that the letter grades are not “given”, they are earned.

**Earning an “A” – Xtra Problems to demonstrate Excellent performance**

To demonstrate Excellent work, a student in this class must do more than the regularly assigned work. Students who undertake the task are expected to choose and evaluate a problem, draw up a design to solve it, research and find any supplemental information that is needed, and develop a rigorous testing sequence – meaning that *it is the student who decides when the program is correct and successful*, and ready for submission.

The instructor provides almost no guidance whatsoever. *The student will get no answer to the question “What do you want?”* The student is demonstrating that he or she has mastered the course material to the extent that the student can now use that mastery to tackle a problem alone, test it thoroughly, and present it “shrink-wrapped” and declaring it ready to be used.

The instructor evaluates whether the program 1) truly solves the problem, 2) has been well tested 3) the code has style, clarity, and good design. Think of it as buying an App or not. Or think of it as a job interview with the program as the student’s “portfolio.” The Xtra Problem earns one of two grades: Satisfactory or Unsatisfactory. Sold or not, hired or not. There is no further discussion, no further work, no second interview.

## Overall Letter Grade achievement

The “regular” scoring is done according to the following scale:

under 40% F

under 50% D

under 75% C

75% and over: B

Working an Xtra Problem successfully will raise a “regular” score of B or C one letter grade:

**Regularly Assigned Work** qualifies for Xtra Problem Score Overall Grade

B S A

C S B

If your score on the regularly assigned classwork has not earned a C, the Xtra Problem will not improve that grade. The Xtra Problem is not “extra credit.” It does not add more points to your overall total so that you can achieve a higher average. Please avoid this common mistake.

## Course work: These are the Regularly Assigned Work items

* **Attendance** is graded: **5 points** per class meeting
  + I am pretty lenient about excusing absences or having to leave early, so you can save the points *but you will still lose the information and/or activity gained in the class*.
  + You are responsible for making up any material missed in class. I will usually post the notes I use in the lecture.
  + Any **quizzes** or **in-class activities** which are scored *may* ***not*** *be made up*.
* **A lecture** is usually a “work-along” activity with the instructor writing pseudocode or code and the class doing the same. Then in **lab time** the students will finish the exercise or complete additional work based on the lecture.
* There will be **3 examinations**. Each will consist of one or more *problems* to be worked within the class period.
  + The examinations are worth **200 points** each.
  + If you miss one of the first two examinations, your score on the next examination will also be your score on the missed exam. *You may not miss more than one examination*.
  + You **must** take the final examination. Whatever else your numbers add up to, **it is not possible to pass this course without taking the final examination**.
* There will be **Program Assignments** (as many as 5), each worth **100 points** and covering the topics and programming tools that have been discussed to that point in time. The assignments are “cumulative” – that is, the second program starts with the first program and either
  + adds more tasks to the problem,
  + asks the student to use a different method to solve the same problem, or
  + a combination of both: more tasks, new methods

## Calculating the Regularly Assigned Work score

It’s rather simple, really: divide the total points you have earned by the total points it is possible to earn. Evaluate the resulting percentage using the top of this page.

# How Learning Behavior translates into my Teaching Method

* I particularly stress Program Design, perhaps more than other instructors and even the textbook.
  + This means writing programs in ***pseudocode******first***, and then building the specific Python code from the design.
* I also stress Program Clarity (good style, readability, well-commented).
  + I will note that "stress" translates into "grade points."
* I also believe that learning anything should be (even *must* be) *enjoyable* if one really wants to retain it.
* Finally, computer programming, like mathematics, is **learned by doing**. That means writing a lot of programs.

All computing machines receive input, process the input, and produce some output. It is both the simplest and most complex of behaviors. It is commonly said that “Information is processed data.” I make the analogy that “Knowledge is processed facts.” I note that as our “distance” (in space and time) from our accumulated and shared knowledge vanishes (from hours and days spent walking to libraries, checking out and reading books, to minutes and seconds googling with a smart phone or a wearable device) the way we learn, and what we learn, is changing before our eyes and ears.

I have worked as a programmer and Computer Scientist since 1966, with as much time or more spent developing computer-based instructional process, as teaching in classrooms. In all those years, I have never, not once, seen a working professional write a program without at least one manual in the room, close at hand, or open on the desk (I have an open $200 bounty for anyone who has seen such a thing). I have worked as an individual consultant, but my primary time in industry was spent on large, group-developed products (“system engineering”). I co-authored a curriculum to teach programming in high school in which teams of students worked problems in a method called “sandboxing” (referring to the way all children learn more by the age of 5, proportionally, than they will learn in the rest of their lives – by playing).

Therefore,

* **I highly encourage working together on homework**: helping and teaching are also learning. But I will not accept identical submissions with different names on them.
* **I love group projects**.
* **I avoid multiple-choice, fill-in-the-blanks, and true/false tests at all costs**. They test whether you have memorized facts, not whether you know what to do with them.
* **All examinations are open-book and open-notes**. I do not expect to separate students from what are becoming our primary source of information – the Internet.
  + Examinations, however, are the one situation where I must require that you not communicate with other people.
  + By definition, an examination is an assessment of what you, personally, have learned. But my examinations assess performance, not the retention of facts. Here’s the book – what can you do with it?

**But be warned:** my open book examinations are much harder than closed-book exams. The exam is not the time to learn something for the first time.

# Course/Class Policies:

## Academic Integrity

* A student may not represent someone else’s work as his or her own.
* A student may not represent his or her work as someone else’s.
* Students are responsible for observing the policy on academic integrity as described in the current Student Policies Handbook.

**The penalty assessed for violations of these policies will be in accordance with the** [**current policy**](http://www.austincc.edu/admrule/1.04.006Guideline.pdf)**.**

For this course, the penalty for scholastic dishonesty is a grade of ‘F’ for the course.

## Incomplete

An Incomplete cannot be used as a shelter against a potentially low grade in the class. A student may receive a temporary grade of “I” (Incomplete) at the end of the semester only if ALL of the following conditions are satisfied:

1. The student is unable to complete the course during the semester due to circumstances beyond their control.
2. The student must have earned at least half of the grade points needed for a “C” by the end of the semester.
3. The request for the grade must be made in person at the instructor’s office and necessary documents completed.
4. To remove an “I”, the student must complete the course by two weeks before the end of the following semester. Failure to do so will result in the grade automatically reverting to an “F”.

## Freedom of Expression Policy

It is expected that faculty and students will respect the views of others when expressed in classroom and/or Blackboard discussions.

## Tutoring

Free tutoring is provided for this course. For online schedules and details please refer to <http://www.austincc.edu/cit/>

## Attendance / Withdrawal

Regular and punctual class and laboratory attendance is expected of all students. If attendance or compliance with other course policies is unsatisfactory, the instructor may withdraw students from the class.

It is the student’s responsibility to complete a Withdrawal Form in the Admissions Office if they wish to withdraw from this class. The last date to withdraw is 11/28/2016. **It is not the responsibility of the instructor to withdraw the students from their class** even though the instructor has the prerogative to do so under the above listed circumstances.

I go one step further: ***I will never perform an Instructor-Initiated Withdrawal*** without a written (or e-mailed) request from a student, with a copy sent to the chair of the Computer Science Department.

Students who enroll for the third or subsequent time in a course taken since Fall 2002 are charged a higher tuition rate. State law permits students to withdraw from no more than six courses during their entire undergraduate career at Texas public colleges or universities. With certain exceptions, all course withdrawals automatically count towards this limit. Details regarding this policy can be found in the ACC College Catalog.

## Student Files – Privacy

The information that a student stores in his/her student volume in the Computer Studies Labs may be viewed by their instructor for educational and academic reasons.

## Students with Disabilities

Each ACC campus offers support services for students with documented physical or psychological disabilities. Students with disabilities must request reasonable accommodations through the Office for Students with Disabilities on the campus where they expect to take the majority of their classes. Students are encouraged to make this request three weeks before the start of the semester. (Refer to the current ACC Student Policies).

## Communication

The ACC online Blackboard system [http://acconline.austincc.edu](http://acconline.austincc.edu/) and the ACCmail accounts will be used as the official communication system during this semester. Lecture notes, handouts, changes to course schedule or assignments and your grades will be posted on Blackboard and all email communication will be via the ACCmail accounts. All students are expected to check both Blackboard and their ACCmail accounts on a regular basis. For information on how to log onto Blackboard and ACCmail please visit the following sites: <http://irt.austincc.edu/blackboard/StudentSupport.php> and <http://www.austincc.edu/google/>.

## Safety Statement

Each student is expected to learn and comply with ACC environmental, health and safety procedures and agree to follow ACC safety policies. Emergency posters and Campus Safety Plans are posted in each classroom. Additional information about safety procedures and how to sign up to be notified in case of an emergency can be found at <http://www.austincc.edu/emergency/> .

Anyone who thoughtlessly or intentionally jeopardizes the health or safety of another individual will be immediately dismissed from the day’s activity, may be withdrawn from the class, and / or barred from attending future activities.

**COURSE SCHEDULE**

This proposed schedule is subject to change, with notice, if circumstances require.

