AUSTIN COMMUNITY COLLEGE
DEPARTMENT OF COMPUTER STUDIES AND ADVANCED TECHNOLOGY

Course Syllabus: COSC 1336 (3-3-1) – Programming Fundamentals I
Section 9 Synonym 69302 **'MW19'** Spring 2019 Northridge

January 22 to May 19 (16 week semester)

Lecture:	NRG 4207	Monday, Wednesday	7:05 pm – 8:25 pm (1:20)
Lab:	NRG 4207	Wednesday	8:35 pm – 9:30 pm (:55)

Instructor: Paul Thayer
Office Telephone: (512) 223-1790 ext: 26414 (voicemail only)
Office: classroom/lab (NRG 4207), open computer lab (NRG 4232), adjunct office (NRG 4216A)
Office Hours: Monday and Wednesday: 6:50 pm – 7:05 pm in NRG 4207
Monday: 8:25 pm – 8:40 pm in NRG 4207
Wednesday: 9:30 pm – 9:45 pm in NRG 4207
(15 minutes before and after each class meeting)
E-mail: pthayer@austincc.edu *Please put* **MW19** *in the email subject line!*
Home page: http://www.austincc.edu/pthayer

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.

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:
Starting out with Python, Fourth edition, Tony Gaddis, Pearson, 2018
ISBN-13: 978-0-13-454366-6 w/Access; (on-line access code not required)

Instructional Methodology: This course has 75% lecture and 25% lab time. If students are unable to finish assigned lab work during lab time, they can use CIS open labs or their own computer.

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 Competencies:

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 course:

RESOURCES 1.1 Manages Time	**INTERPERSONAL**	**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: Grades will be assigned based on concepts and practical application. Exams, quizzes, and lab projects will be a part of the grade. There is no curve. Your grade is based on your individually earned point total. Point totals ***are not*** "rounded up" to the next higher grade level. Extra credit opportunities are included in many assignments. See the Blackboard MyGrades area for your current point total on graded assignments. An overall grade is assigned based on this scale:

Points Percentage	90% - 100%	80% - 89.9%	70% - 79.9%	60% - 69.9%	0% - 59.9%
Grade	A	B	C	D	F

Each student's grade for this course consists of attendance (3%), class exercises (2%), quizzes (12%), programming labs (30%) and three exams (53%) as indicated in the table below:

(30) Attendances	1 point/class	30 points total	3%
(1) In-class exercise	20 points/exercise	20 points total	2%
(10) Chapter Quizzes	12 points/quiz	120 points total	12%
(10) Programming Labs	30 points/lab	300 points total	30%
Exam 1 (Ch 1 – 4)	100 points (30-35-35)	100 points total	10%
Exam 2 (Ch 1 – 4, 5 – 7)	180 points (60-60-60)	180 points total	18%
Exam 3 (Ch 1 – 7, 8 – 10)	250 points (80-85-85)	250 points total	25%
TOTAL		**1000 points**	**100%**

Each quiz or lab ***is due two weeks after the date assigned.*** Quizzes are short, on-line (Blackboard) multiple-choice tests that can be taken at the student's convenience, during lab or remotely. They are a check on the student's reading and comprehension. Quizzes are open-book, open note. The student should complete the quiz without help from another person prior to the upcoming exam. The student may save a quiz attempt over multiple sessions, but it must be submitted for grading and credit. ***Quizzes are due within two weeks of being assigned, but no late penalty is assessed.***

Lab assignments are considered late if submitted after two weeks of the date assigned, but will be accepted with a 20% late penalty if submitted within four weeks; and will be accepted with a 50% late penalty if submitted within six weeks of the date assigned. Six weeks after a lab is assigned, it will disappear from Blackboard and the student will not be able to submit that lab assignment.

lab submitted within:	two weeks	four weeks	six weeks	past six weeks
considered, penalty	on-time, none	late, -20%	very late, -50%	not available

Each exam tests the student on skills learned from the preceding quizzes and labs. You should complete assignments prior to being tested on them. The exam tests concepts from all prior labs. ***A common penalty for late work is that you lose points on the exam because you were not prepared.***

The last date to submit assignments for consideration this semester is Friday, May 17, 2019 (two days *after* the last day of class, by 11:59 pm).

To identify files submitted for grading, the instructor adopted a file naming convention. The student shall prepend course designator ('**MW19**'), underscore ('_'), L# or H#, and last name before the filename. Example: Jessica Jones attends a Monday/Wednesday class starting at 7:05 pm. For lab 3, she submits the file: **MW19_L3_Jones.py** for grading. Files are submitted via the Blackboard assignment link. To **undo** and **redo** a submission before grading, ask the instructor to clear the submission. In an emergency, if the Blackboard link is unavailable, use ACC email.

Exams 1, 2 and 3 consist of one at-home review portion and two in-class portions (concept and lab). There are **NO** makeup exams given in this course. If a student misses an exam, it may be taken late, but a late penalty will apply. Only one exam may be missed and there will be **NO** make up for Exam 3. If you cannot take an exam on the regularly scheduled day in class due to illness, family emergency, or some other excusable, unavoidable conflict, contact the instructor ASAP about making special arrangements.

Academic Integrity

A student is expected to complete his or her own projects and tests. Students are responsible for observing the policy on academic integrity as described in the current ACC Student Handbook.

Students are encouraged to work together to learn to program. During lab time, you may collaborate on the development of programs, **but each student's program is expected to be uniquely coded, tested and submitted independently**. If you collaborate on a lab with another student and your lab is similar to another student's lab, ***both*** students who worked together should record who they worked with in the comments. If substantially similar labs are submitted without a declaration of collaboration, the score for the labs may be reduced to zero on suspicion of cheating (submitting another student's work as your own). **The reduction to zero applies to *both* the sender and receiver of similar lab code.**

The penalty accessed will be in accordance with the current ACC policy. See: http://www.austincc.edu/handbook for more information.

Incomplete
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 unforeseen circumstances beyond their control.
2. The student must have earned at least half of the grade points (500) offered by the course 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 discussions.

Tutoring Free tutoring is provided for this course. For schedules and details please refer to http://www.austincc.edu/support-and-services/tutoring-and-academic-help/tutoring-services-and-schedules.

Attendance / Withdrawal
Students are expected to attend classes and will be held responsible for all material covered in class. Regular attendance helps ensure satisfactorily progression towards completion of the course. **Attendance is recorded for each class!** Students are expected to initial the attendance roster each day of class attended. Failure to initial the roster may result in losing attendance credit. Monday, January 21, 2019 (Martin Luther King, Jr. Day) is a Holiday (college closed).

It is the student's responsibility to follow the Course Withdrawal/Drop Instructions if they wish to withdraw from this class. See: http://www.austincc.edu/online-services/course-drop-withdrawal. **The last date to withdraw for this semester is Monday, April 29, 2019**. 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. ***Typically, the instructor does not withdraw students!*** On the last day to withdraw, if it appears a student cannot pass the course, the instructor may withdraw a student. ***Do not rely on the instructor to withdraw you from the course!***

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. See: http://www.austincc.edu/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. See current ACC Student Policies at: http://www.austincc.edu/support-and-services/services-for-students/student-accessibility-services-and-assistive-technology.

Communication
The ACC online Blackboard system http://acconline.austincc.edu and ACC Gmail 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 ACC Gmail accounts. All students are expected to check both Blackboard and their ACC Gmail accounts on a regular basis. For information on how to log onto Blackboard and ACC Gmail please visit the following sites:

http://acconline.austincc.edu	(Blackboard logon)
http://irt.austincc.edu/blackboard/StudentSupport.php	(first time Blackboard user)
http://www.austincc.edu/google/ or http://www.austincc.edu/accmail	(ACC email login)

Instructors/Lab Technicians will conduct a brief lab orientation during the first class laboratory period. If this is your first semester at ACC, you have some extra work to set up your ACC email account, your Blackboard account, etc. There is help available to accomplish these tasks. First check the ACC website: http://www.austincc.edu and other austincc.edu links (see above). Also ask lab assistants (available in every CIT lab), or call ACC help line at: (512) 233-HELP (4357).

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/. In the event of an extreme emergency or impending threat, ACC Emergency Alert can send critical voice and text messages to your cellphone. You can register or update your emergency contact information starting at the link above.

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.

COSC 1336 – Programming Fundamentals I – SPRING 2019 – Syn 69302 – Sec 9 'MW19' NRG 4207 Mon/Wed 7:05p – 8:25p (lec); Wed 8:35p – 9:30p (lab)					Assignments: Quizzes and Labs
Wk #	**Date**	**Day**	**Topic**	**Read Gaddis**	***listed on date assigned on-time <= 2 weeks -20%<=4; -50%<=6 wks***
1	**1/21**	**Mon**	**Martin Luther King, Jr. Day Holiday (College Closed)**		
	1/23	**Wed**	Course Intro: policies, grading, meet classmates Lab Orientation (logins: network, email, Blackboard)		
2	**1/28**	**Mon**	Ch 1: Introduction to Computers and Programming Hardware: CPU, RAM, hard drive; Intro to Python	Ch 1	**Quiz 1 – Chapter 1** **Lab 1 (ch1)**
	1/30	**Wed**	"Being A Bit" – binary data storage and manipulation		**In-class exercise 1, Be-A-Bit**
3	**2/4**	**Mon**	Chapter 2: Input, Processing and Output Using Python with IDLE	Ch 2	**Quiz 2 – Chapter 2** **Lab 2 (ch2)**
	2/6	**Wed**	Chapter 2: Input, Processing and Output, continued Introduction to Turtle Graphics		
4	**2/11**	**Mon**	Chapter 3: Decision Structures and Boolean Logic	Ch 3	**Quiz 3 – Chapter 3; Lab 3 (ch3)**
	2/13	**Wed**	Ch 3: Decision Structures and Boolean Logic, cont'd		
5	**2/18**	**Mon**	Chapter 4: Repetition Structures (loops)	Ch 4	**Quiz 4 – Chapter 4; Lab 4 (ch4)**
	2/20	**Wed**	Chapter 4 Repetition Structures, (cont'd); Turtle Graphics		
6	**2/25**	**Mon**	Review for EXAM 1 (Chaps 1 - 4)		
	2/27	**Wed**	Chapter 5 Functions (void: define, call) (preview)		**EXAM 1, start**
7	**3/4**	**Mon**	**EXAM 1 finish** (Chaps 1 - 4) (30-35-35)		
	3/6	**Wed**	Chapter 5 Functions (input: parameters and arguments) Turtle Graphics	Ch 5	**Quiz 5 – Chapter 5**
8	**3/11**	**Mon**	Chapter 5 Functions (output: returns and side-effects)		**Lab 5 (ch5)**
	3/13	**Wed**	Chapter 6: Files and Exceptions	Ch 6	**Quiz 6 – Chapter 6**
Congratulations! You made it to *SPRING BREAK!* Monday, March 18 to Friday, March 22 (College closed)					
9	**3/25**	**Mon**	Chapter 6: Files and Exceptions (continued)	Ch 6	**Lab 6 (ch6)**
	3/27	**Wed**	Chapter 7: List and Tuples	Ch 7	**Quiz 7 – Chapter 7**
10	**4/1**	**Mon**	Chapter 7: List and Tuples (continued)		**Lab 7 (ch7)**
	4/3	**Wed**	Combine: loops, functions; lists, tuples, files		
11	**4/8**	**Mon**	Review for EXAM 2 (Chapters 5 -7)		
	4/10	**Wed**	Chapter 8: More about Strings (preview)		**EXAM 2, start**
12	**4/15**	**Mon**	**EXAM 2 finish** (Chapters 5 - 7) (60-60-60)		
	4/17	**Wed**	Chapter 8: More about Strings (continued)	Ch 8	**Quiz 8 – Chapter 8** **Lab 8 (ch8)**
13	**4/22**	**Mon**	Chapter 9: Dictionaries and Sets	Ch 9	**Quiz 9 – Chapter 9** **Lab 9 (ch9)**
	4/24	**Wed**	Chapter 9: Dictionaries, Sets (continued), and Pickle		
Notice: Monday, April 29, 2018 is the ACC Final Withdrawal Date!					
14	**4/29**	**Mon**	Chapter 10 Classes and Object-Oriented Programming	Ch 10	**Quiz 10 – Chapter 10**
	5/1	**Wed**	Chapter 10 Classes and OO Programming, cont'd		**Lab 10 (ch10)**
15	**5/6**	**Mon**	Chapter 10 Classes and OO Programming, cont'd		
	5/8	**Wed**	More on Python – Combining features		
16	**5/13**	**Mon**	Review for EXAM 3 (Chapters 1 – 10)		**EXAM 3, start**
	5/15	**Wed**	**EXAM 3 finish** (Chapters 1 – 10) (80-85-85)		

The last date to submit assignments for consideration this semester is Friday, May 17, 2019

The instructor has the prerogative to change the course schedule as required. Students are strongly encouraged to read over the current chapter, per the course schedule, before each class. More information on quizzes, labs, and specific sections to read in the textbook will be available in class and on Blackboard

What I look for when grading labs

1) At the top of each program, I expect a heading with a short comment (a few lines) which has the author's name, the course name, the lab number, and a brief description of what the program does. This practice will be introduced for Lab 1, but should be continued for all labs.

2) Use a consistent, easy-to-read indenting format. Python requires correct indenting as part of the language, which helps a lot. However, you still have to keep the indenting levels consistent. You may choose your own style and change it over time, but it should be consistent within a single program.

3) At the bottom of your program, include a comment block that shows the output results of testing your program. Programs will generate text that is displayed in a window. This text can be copied and pasted into your program, and placed in a comment block. I will demonstrate how to do this for lab 1.

4) Include brief comments to explain what your code is doing if not clearly indicated. There is **no need to comment:** CurrentYear = 2019 # create a new variable for the current year, set it to 2019. However, it is important to comment complex code that performs many non-obvious tasks. The best "commenting style" is self-documenting code that uses accurate and descriptive names.

5) Name your files correctly. Use the course designator (**MW19**) as described above. For example: **MW19_L3_Jones.py**. For labs, submit one (or more) ASCII text (.py) files.

6) Submit your assignments on time. You have two weeks OR up to the next scheduled exam, whatever comes first. Work time will typically be two weeks, but it can be less than two full weeks for labs that are assigned just prior to the upcoming exam. There is always a full week and one review class with no new assignment prior to an exam.

An otherwise excellent lab may have several points deducted if items 1 to 6 are missing. This applies to all labs. I will describe these items in detail early in the course, but may not mention them after a few weeks.

The first thing I do with your lab is run the program. I pretend I'm a "dumb user". That is, I try to use your program to do useful work. Don't assume I know all about your program, how it works, what it does, etc. **Provide a good user interface.** That includes: useful prompts, well-formatted output, and a graceful way to quit. The prompt: "Enter number" is very unfriendly. Much better is: "Enter a positive integer; I will tell you if it is a prime number (0 to quit)". It is best if a program displays a brief explanation when it begins and says: "good-bye" when it ends.

Test your code. Failure to test the code you submit can result in point reduction, even if it works! Proof of testing is the comment block at the bottom of your lab (see item #3 above). Good programs detect and handle invalid user input. **Test for failure as well as success.** Reasonable and expected user input should not crash or hang your program. The program should handle attempts to divide by zero, read from missing file or invalid file name. Check output for invalid results. As the course progresses, I am more particular about user friendliness, input validation and error handling. You won't want to use your own code if it is poor quality.

After running the program, I look at the code. Things I look for include: consistent and meaningful indenting, short, descriptive, accurate variable names, helpful comments, identification and reduction of redundant code, good organization, ease of testing (loops are better than code that just runs once and stops) . Correct output is very important. Some very nice looking code can produce incorrect results!

Programming is an art form. It is a creative act that varies significantly between different practitioners. With some practice and constructive feedback, I hope you can improve your skills and gain confidence. It is very satisfying to get a computer to do what you want.