The College of Charleston
Computer Science Department
CSCI 220 Computer Programming I
Fall, 2016

Instructor George J. Pothering (potheringg@cofc.edu)

Office Harbor Walk East, Room 327

Office Hours TR 11:30 -12:30
Other hours by appointment

Telephone 953-8156 (office)
884-4583 (home)

Course Meeting TR 9:55 - 11:10 Harbor Walk East, Room 300

Course Description: An introduction to programming and problem solving using Python. Topics include data types, variables, assignment, control structures (selection and iteration), arrays, methods, classes and an introduction to object-oriented programming.

Textbook: Zelle. John. *Python Programming: An Introduction to Computer Science* (2nd ed)
Franklin, Beedle & Associates, 2010. ISBN:978-1-590218-241-0.

Important Dates for This Course:

August 23	First day of class
November 7-8	Fall Break – no class
November 23 - 27	Thanksgiving Break – no class
December 1	Last day of class
December 8	Final Exam (8:00 am – 11:00 am)

Class Policies

I. ***Classes***: Attendance will be taken each day. Students who arrive after attendance has been taken will be counted as absent. You are responsible for all work done or required for any class that you miss, i.e. doing the preparatory work for class, getting notes, turning in assignments, etc. On average you should expect to work six to nine hours *outside of class* each week on this course.

II. ***Exams:*** You will take two semester exams in addition to a comprehensive final examination. The dates of the semester exams are

Tuesday, September 27
Tuesday, November 1

The final examination is currently scheduled for:
Thursday, December 8 from 8:00 am to 11:00 am.

III. ***Assignments***: Programming assignments will be issued throughout the semester and will normally be due the second class after the one in which they are assigned. Your code must be submitted via Oaks **and a printed copy of your work handed in to me**. Any work which is submitted after the due date will be accepted under the Russian roulette policy. If more than one page is being submitted please staple the pages together.

"Well begun is half done" (Aristotle). Although not explicitly listed as an outcome of the course, one of the habits we strive to instill in you is a realization that programming requires a ***disciplined*** approach to problem solving and that successful problem solving itself is founded on principles which require ***sufficient time*** to be fruitful. Work that is done at the last minute just for the sake of turning in work for some credit is not consistent with a disciplined approach.

You are expected to do your own work on each assignment. Anyone who submits work that is not his or her own may be taken before an honor board. If found guilty of an honor code violation, you will receive a grade of "XF" for the course

indicating failure of the course due to academic dishonesty. This grade will appear on your transcript for two years after which you may petition for the X to be expunged.

IV. ***Grading Scale:*** I do not use a strict grading scale such as the 10-point scale (90-100, 80-89, etc) commonly used by other instructors Instead, scales derived from clusters of student performance will be given for each test. Your status after every test can be determined by adding the boundaries of the individual scales and finding where your accumulated scores lie within these intervals. The exact way this works will become clear after the first test.

V. ***Course Grades:*** Students with 3 or fewer absences and who complete every assignment will have their best test or exam grade counted again in determining their course grade. Your efforts in the course will be weighted as follows:

Semester Exams	2/6 each	(2/5 each without best test grade being counted again)
Final Examination	2/6	(2/5 without best test grade being counted again)
Assignments	1/6	(1/5 without best test grade being counted again)
Best Test Grade	1/6	only if you qualify

VI. Classroom Behavior: To maintain a classroom environment that is conducive to learning, I expect certain behavior of students in my classes. Students who, during class, check and send e-mail or text messages, browse the Web, Tweet, giggle, sleep, yawn audibly, whisper, whine, groan, arrive late, leave early, or come unprepared are disrupting to me and detract from their fellow students' learning experiences. At the end of the semester you may be just below the cut-off for a higher grade. If the impression of you that comes to mind is of someone who is uninterested, disruptive, rude or otherwise lacking in classroom etiquette and deportment, do you really think anyone is going to reward you for this?

VII. ***Honor Code and Academic Integrity:*** Lying, cheating, attempted cheating, and plagiarism are violations of our Honor Code that, when identified, are investigated. Each instance is examined to determine the degree of deception involved.

Incidents where the professor believes the student's actions are clearly related more to ignorance, miscommunication, or uncertainty, can be addressed by consultation with the student. We will craft a written resolution designed to help prevent the student from repeating the error in the future. The resolution, submitted by form and signed by both the professor and the student, is forwarded to the Dean of Students and remains on file.

Cases of suspected academic dishonesty will be reported directly to the Dean of Students. A student found responsible for academic dishonesty will receive a XF in the course, indicating failure of the course due to academic dishonesty. This grade will appear on the student's transcript for two years after which the student may petition for the X to be expunged. The student may also be placed on disciplinary probation, suspended (temporary removal) or expelled (permanent removal) from the College by the Honor Board.

It is important for students to remember that unauthorized collaboration--working together without permission-- is a form of cheating. Unless a professor specifies that students can work together on an assignment and/or test, no collaboration is permitted. Other forms of cheating include possessing or using an unauthorized study aid (such as a PDA), copying from another's exam, fabricating data, and giving unauthorized assistance.

Remember, research conducted and/or papers written for other classes cannot be used in whole or in part for any assignment in this class without obtaining prior permission from the professor.

Students can find a complete version of the Honor Code and all related processes in the Student Handbook at http://www.cofc.edu/studentaffairs/general_info/studenthandbook.html.

VIII. ***Disability Accommodation:*** Any student who feels he or she may need an accommodation based on the impact of a disability should contact me individually to discuss your specific needs. Also, please contact the College of Charleston, Center for Disability Services http://www.cofc.edu/~cds/ for additional help.

Course Outcomes [numbers in brackets is the Bloom's Taxonomy level of mastery expected] ***:***

The order is intended to be "logical", and does not imply "chronological".

1. Understand the **software development process** and the **activities** included within it. [2, understand]
2. Apply **top–down design** in program development (functional decomposition for algorithmic design) as a mechanism for handling problem complexity as well as facilitating team programming and software reuse [3, apply]
3. Apply elements of **good style**: documentation, formatting, meaningful identifiers, capitalization [3, apply]
4. Apply **constants** and **variables** in program development (including knowing when to use which) [3, apply]
5. Analyze **assignment** including semantics of LHS versus RHS (eg. What does "x = x + 1" do? Why does it work in a program, but not in algebra?) [4, analyze]
6. Apply **I/O operations** from standard input and output to different data types [3, apply]
7. Apply **file I/O operations** to text files [3, apply]
8. Design expressions using **arithmetic operations** including understanding their limitations, such as integer truncation, round–off error, division by zero, narrowing and widening conversions, casting, precedence, and standard math library functions (eg. Write the expression to convert from Celsius to Fahrenheit) [5, synthesize]
9. Design expressions using **relational operators** (including proper floating point equality comparison) [5, synthesize]
10. Design expressions using **logical operators** (including short–circuit) [5, synthesize]
11. Design **selection statements** (including nested selection) [5, synthesize]
12. Design **repetition statements** (including count–controlled versus event–controlled, sentinel–controlled) [5, synthesize]
13. Design applications using **strings** [5, synthesize]
14. Design applications using simple data structures using **lists** (including using loops with lists, parallel lists, multi–dimensional lists (lists within lists), shallow versus deep comparison, common task such as finding max/min, summing, and other aggregate operations) [5, synthesize]
15. Design **functions** (including pass–by–value versus pass–by–reference, formal versus actual parameters, value–returning vs. nothing-returning and when to use which, overloading) [5, synthesize]
16. Apply **sequential search** in program development (eg. find the index of the first occurrence of 'A' in a list) [3, apply]
17. Apply **some N^2 sort** in program development (not simply call a built-in sort) [3, apply]
18. Write code using **objects** [5, synthesize]
19. Write code defining **classes** [5, synthesize]

Bloom's Taxonomy of Levels of Mastery:

1. **Recall of Data, Knowledge:** Students can recite memorized information about *the concept*.
2. **Comprehension:** Students can explain memorized information about *the concept* in their own words.
3. **Application:** Student can demonstrate his comprehension by applying *the concept* as a whole.
4. **Analysis:** Students can separate *the concept* into its component parts and apply these parts in another situation.
5. **Synthesis:** Students can design *an artifact* by combining components of *the concept*.
6. **Evaluation:** Students can judge *a novel artifact* created by combining components of *the concept*.