CSCI 120A – Fall 2014
Computer Science I

Ben Coleman
coleman@cs.moravian.edu
PPHAC 214

Office Hours: MWF 10:30 – 11:30
R 10:00 - 11:30
or by appointment

Course Description

The goal of this course is to introduce students to the techniques computer scientists use to answer questions and solve real-world problems. The course emphasizes the design and creation of computer programs to solve problems and the analysis of program capabilities. In general, what is the best way to solve a given problem? Computer programming will be used as a vehicle for learning computer science concepts. In particular, this course includes data types, control structures, functional abstraction, parameter passing, and structured data, including simple objects.

During the in-class portion of the course, students will learn programming skills and discuss applications of these ideas. Weekly laboratories give students the opportunity for hands-on exploration of the material and the chance to solve real-world problems.

Course Goals

Upon completion of this course, a successful student will be able to:

- Describe how the concepts of computer science are applied to solve real-world problems.
- Write programs in Python using assignments, conditions, loops, functions, and objects.
- Measure the performance of computer programs using appropriate mathematical notation.
- Break down problems using top-down / bottom-up design and functional decomposition.

Required Texts

In addition to the following text, supplementary readings will be given periodically during the semester.

- *Python Programing: An Introduction to Computer Science* Second Edition by John Zelle

You should expect to spend about an hour before each class session working through the readings. This means reading the text for detail, typing code into the computer, studying the syntax for new language features, and working to learn vocabulary – not just skimming through the material before class.

Graded Material

- **Homework** – The goal of homework problems is for you to practice using the current course content and to explore the topics in more detail. Problems will be assigned nearly every class session and will be due the next class.

  Each homework problem will be graded using the following scale:

  (3) "You got it" – The solution is perfect or near perfect.
  (2) "You mostly got it" – The solution has some errors or omissions but was headed in the right direction.
  (1) "You were far off" – The solution has significant errors or omissions, but a serious attempt was made.
  (0) "You didn't do much" – The solution shows little progress or the problem was not attempted.

  At the end of the semester, your homework grade will be computed as follows:

≥ 2.5	A
≥ 2	B
≥ 1.5	C
≥ 1	D
< 1	F

- **Laboratory Exercises** – Lab sessions will be held in the Computer Science Computer lab, PPHAC 114. During the lab, you will be given a series of activities to complete, individually or with a partner, depending on the lab. Labs are designed to utilize concepts of the past week's classroom session to answer real-world problems. In addition to writing programs to solve the problems, you will collect data from the program and write-up answers to questions.

- **Culture Points** – One of the goals of this course is for you to gain an appreciation of how computer scientists contribute to the world-at-large. The examples used in class will contribute toward this goal, but you are also expected to explore other applications on your own. There are no specific due dates for culture point submissions, but you are encouraged to submit write-ups regularly throughout the semester. See the handout on culture points for additional information.

- **Tests** – Three tests will be given during the semester on Monday, September 22, Friday, October 24, and Monday, November 24. You may only re-schedule a test for college-approved absences or documented illness. In either case, you must contact me *before* the beginning of the test.

- **Final** – The final will be cumulative and will be given in-class on Tuesday, December 9 at 8:30 a.m. Any change to the final exam schedule must be approved by both me and the dean of students.

- **Participation** – Half of your participation grade is determined solely on your attendance in class (irregardless of whether an absence is excused). The other half is based on active participation. I believe that we learn better when we are *actively* engaged in the material. Therefore, I expect you to participate in the activities in class and contribute on a regular basis.

Grade Determination

- (25%) Homework
- (20%) Laboratory exercises
- (5%) Culture Points
- (25%) Tests
- (15%) Final
- (10%) Attendance and Participation

All grades will be calculated on the standard scale using pluses and minuses.

Course Policies

- **Late Policy** – I understand that life sometimes gets in the way of getting work done. Consequently, late assignments will be accepted without penalty in the class after the assignment is due. However, this policy should not be used as a crutch, and if you frequently use it I will deduct from your grade. After the next class session, late work will not be accepted unless there are exceptional circumstances.
- **Extensions** – In a similar vein, I am generous with extensions on work if you approach me *before* the day the assignment is due.
- **Absences** – Your attendance is expected at each class meeting, but I understand that students occasionally get sick, have obligations outside Moravian, and even over sleep. If you do miss class, please send me an email explaining your absence – preferably before the class session. Regardless of your reason for missing class, you are responsible for the contents of reading assignments, handouts, class activities, and class email.
- **Academic Honesty** – Except on tests, you are *encouraged* to discuss the material and work with other students in the course. Specifically, on homework and labs you may discuss any portion of the assignment with your fellow students. This policy does not allow you to copy another student's work verbatim – you must produce your own code or write-up of the material. Work together to learn the concepts, but keep in mind that you are ultimately responsible for the material on the tests.
- **Disabilities** – Students who wish to request accommodations in this class for a disability should contact the Assistant Director of Academic and Disability Support in the Academic Support Center, Monocacy Hall, lower level, or by calling 610-861-1401. Accommodations cannot be provided until authorization is received from the Academic Support Center.

Course Outline

Date	Reading	Topic
M Aug 25		• Day 1 Activities
W Aug 27	• Zelle Chapter 1	• Computer Hardware and Software • Compiling
F Aug 29	• Zelle Sections 1.6 – 1.9 • Zelle Section 2.1 – 2.5	• A First Program
M Sept 1	• Zelle Sections 2.6 – 2.8	• Input / Output • Assignment Statements • Definite Loops
W Sept 3	• Zelle Sections 3.1 – 3.3	• Numeric Programming • Using Libraries
F Sept 5	• Zelle Sections 3.4 – 3.6	• Numeric Representation and Type • Type Conversions
M Sept 8	• Zelle Sections 4.1 – 4.5	• Using Objects • Graphics
W Sept 10	• Zelle Sections 5.1 – 5.2	• Strings
F Sept 12	• Zelle Sections 7.1 – 7.3 • Zelle Sections 7.5 – 7.6	• Decision Statements
M Sept 15	• Zelle Sections 5.2 – 5.6	• Lists
W Sept 17	• Zelle Sections 5.7 – 5.8	• List Processing
F Sept 19		• Slip Day / Review
M Sept 22		• Test #1
W Sept 24	• Zelle Sections 8.1 – 8.3	• Looping Statements
F Sept 26	• Zelle Section 8.4	• Loop Design
M Sept 29	• Zelle Section 8.5	• Loop Patterns
W Oct 1		• Loop Practice
F Oct 3	• Zelle Sections 6.1 – 6.3	• Functions
M Oct 6	• Zelle Sections 6.4 – 6.5	• Parameters and Return Values
W Oct 8	• Zelle Section 6.6 • Zelle Sections 9.1 – 9.3	• Top-Down Design
F Oct 10	• Zelle Sections 9.4 – 9.6	• Bottom-Up Design
M Oct 13 W Oct 15		• Fall Break
F Oct 17	• TBD	• Unit Testing
M Oct 20		• More Unit Testing

Date	Reading	Topic
W Oct 22		• Slip Day / Review
F Oct 24		• Test #2
M Oct 27	• Malik pp. 548 – 555	• Big-Oh Notation • Algorithm Analysis
W Oct 29	• Malik pp. 568 – 575	• More Analysis
F Oct 31		• Algorithm Analysis Examples
M Nov 3		• Selection Sort
W Nov 5		• Insertion Sort
F Nov 7		• Bubble Sort
M Nov 10		• Sorting Analysis
W Nov 12	• Zelle Sections 10.1 – 10.3	• Defining Classes
F Nov 14	• Zelle Sections 10.4 – 10.5	• Encapsulation
M Nov 17	• Zelle Sections 12.1 – 12.2	• Object-Oriented Design
W Nov 19	• Zelle Section 5.9	• File Processing
F Nov 21		• Slip Day / Review
M Nov 24		• Test #3
W Nov 26 & F Nov 28		• Thanksgiving Break
M Dec 1	• TBD	• The Prisoner's Dilemma
W Dec 3		• The Prisoner's Dilemma Competition
F Dec 5		• Review

The details of this syllabus and schedule are subject to change based on our progress through the material.