CSCI 120 – Fall 2007
Computer Science I

Ben Coleman
coleman@cs.moravian.edu
214 PPHAC
Office Phone: 610-625-7781

Office Hours: MWF 9:00 - 10:00
or by appointment
AIM: bjcoleman15

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. We will consider the current state of the discipline and the implications of its application. 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, and objects.
- Read source code, critique its form, and describe its functionality.
- Represent information using appropriate encodings such as binary, two's complement, floating point, and basic linear data structures.
- Measure the performance of computer programs using appropriate mathematical notation.
- Break down problems using top-down design, functional decomposition, and stepwise-refinement.

Required Text

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

- *Python Programming: An Introduction to Computer Science* 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, 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.

- **Laboratory Exercises** – Each Thursday session will be held in the Computer Science Computer lab, PPHAC 114. During the lab, you will be given as 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 the 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 24, Monday, October 22, and Monday, November 19. 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 the date assigned by the registrar. 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 (excused absences do not count against you). 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

- (30%) Homework
- (20%) Laboratory exercises
- (5%) Culture Points
- (20%) Tests
- (15%) Final
- (10%) Attendance and Participation

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

Responsibilities

Your attendance is expected at each class meeting. You are also responsible for the contents of reading assignments, handouts, class activities, and class email.

If you have a disability that may affect your participation in this course, please contact me immediately to discuss academic accommodations.

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.

Schedule

Date	**Reading(s)**	**Topic(s)**
M Aug 27		• Day 1 Activities
W Aug 29	• Zelle Chapter 1	• Computer Hardware and Software • Compiling
R Aug 30		• Lab 1: Chaos and Randomness
F Aug 31	• Murtagh Chapter 1 • Zelle Sections 2.1 - 2.2	• Being Digital • The Software Development Cycle
M Sept 3		• Labor Day
W Sept 5	• Sections 2.3 - 2.8	• Input / Output • Assignments • Definite Loops

Date	Reading(s)	Topic(s)
R Sept 6		• Lab 2: I/O and GUI programming
F Sept 7	• Zelle Sections 3.1 - 3.3	• Numeric Programming • Using Libraries
M Sept 10	• Zelle Sections 3.4 - 3.7	• Numeric Representation and Type • Type Conversions
W Sept 12	• Murtagh Chapter 2	• Two's Complement • IEEE Floating Point
R Sept 13		• Lab 3: Monte Carlo Simulation
F Sept 14	• Zelle Sections 4.1 - 4.3	• Strings and String Processing • Simple Lists
M Sept 17		• More String Processing
W Sept 19	• Zelle Sections 4.6 - 4.7	• File Processing
R Sept 20		• Lab 4: Pop Mail Reader
F Sept 21	• Zelle Sections 7.1 - 7.3 • Zelle Sections 7.5 - 7.6	• Decision Statements
M Sept 24		• Test #1
W Sept 26	• Zelle Sections 8.1 - 8.3	• Looping Statements • Common Loop Patterns
R Sept 27		• Lab 5: More Simulation
F Sept 28	• Zelle Section 8.4	• Loop Design
M Oct 1	• Zelle Section 8.5	• Program Design
W Oct 3	• Murtagh Chapter 3 pp. 25 - 41	• Variable-Length Codes
R Oct 4		• Lab 6: Morphing
F Oct 5	• Murtagh Chapter 3 pp. 41 - 48	• Huffman Encodings
M Oct 8		• Fall Break
W Oct 10	• Zelle Sections 6.1 - 6.3	• Functions
R Oct 11		• Lab 7: Image Processing
F Oct 12	• Zelle Sections 6.4 - 6.5	• Function Parameters • Function Return Values
M Oct 15	• Zelle Section 6.6 • Zelle Sections 9.1 - 9.3	• Top-Down Design
W Oct 17	• Zelle Sections 9.4 - 9.6	• Bottom-Up Design
R Oct 18		• Lab 8: Steganography
F Oct 19	• Zelle Sections 10.1 - 10.3	• Defining Classes

Date	Reading(s)	Topic(s)
M Oct 22		• Test #2
W Oct 24	• Zelle Sections 10.4 - 10.5	• Encapsulation
R Oct 25		• Lab 9: The Prisoner's Dilemma
F Oct 26	• Zelle Sections 11.1 - 11.3	• List Processing
M Oct 29		• More List Processing
W Oct 31	• Malik pp. 548 - 555	• Big-Oh Notation • Algorithm Analysis
R Nov 1		• Lab 10: More Dilemmas
F Nov 2	• Malik pp. 568 - 575	• Sorting Analysis
M Nov 5	• Zelle Sections 13.1 - 13.2	• Recursion
W Nov 7	• TBD	• Recursion and Recurrences
R Nov 8		• Lab 11: Game Trees
F Nov 9	• Zelle Sections 12.1 - 12.2	• Object-Oriented Design
M Nov 12	• Weisfeld Chapter 6	• Designing with Objects
W Nov 14		• Object-Oriented Example
R Nov 15		• Lab 12: More Games
F Nov 16		• Object-Oriented Example
M Nov 19		• Test #3
W Nov 21 - F Nov 23		• Thanksgiving Break
M Nov 26	• TBD	• Traveling Salesman Problem
W Nov 28	• TBD	• TSP: Brute-Force Solutions
R Nov 15		• Lab 13: Traveling Salesman Problem
F Nov 30	• TBD	• TSP: Greedy Algorithms
M Dec 3	• TBD	• Evolutionary Algorithms
W Dec 5	• TBD	• TSP: Evolutionary Algorithms
R Dec 6		• Lab 14: TSP – Evolutionary Algorithms
F Dec 7	• TBD	• TSP: Other Approximation Algorithms
M Dec 10		• Review

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