CSCI 120 – Fall 2015
Computer Science I

Ben Coleman
coleman@cs.moravian.edu
PPHAC 214

Office Hours: MW 2:30 – 3:30
R 10:00 - 12:00
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.
- 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 average homework problem score will be translated to a grade as follows:

≥ 2.5	A
≥ 2	B
≥ 1.5	C
≥ 1	D
< 1	F

- **Programming Assignments** Three programming assignments will be assigned during the semester. These are larger programming tasks, and you will be give one to two weeks to complete the work. Each programming assignment will be graded out of 100 points.

- **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. Each lab will be graded out of 25 points.

- **Tests** – Two tests will be given during the semester on Friday, October 9 and Monday, November 23. 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 Thursday, December 17 at 8:30 a.m. Any change to the final exam schedule must be approved by both me and the dean of students.

- **Attendance and Participation** – At the beginning of each class session I will pose at least one question to the class using plickers. Each class session will be worth one plicker point plus one point for each question that has a correct answer. This will serve as both a way to take attendance and a simple mechanism to evaluate where you are with the course material.

  If you lose your plicker card or forget to bring it to class you will not earn any plicker points for that day. If your card is lost, come to my office, and I will print you another one.

You cannot make up plicker points, but I will give every student two free days where you earn credit for the day's questions even if you are late, absent, or have lost/forgot your card.

Grade Determination

- (12%) Homework
- (17%) Laboratory exercises
- (22%) Programming Assignments
- (22%) Tests
- (15%) Final
- (12%) Attendance and Participation

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, programming assignments, 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 31		• Day 1 Activities
W Sept 2	• Zelle Chapter 1 • Zelle Sections 2.1 – 2.5	• A First Program • Tracing Programs
F Sept 4	• Zelle Sections 2.6 – 2.8	• Input / Output • Assignment Statements • Definite Loops
M Sept 7	• Zelle Sections 3.1 – 3.3	• Numeric Programming • Using Libraries
W Sept 9	• Zelle Sections 3.4 – 3.6	• Numeric Representation and Type • Type Conversions
F Sept 11		• Problem Day
M Sept 14	• Zelle Sections 4.1 – 4.5	• Using Objects
W Sept 16	• Zelle Sections 7.1 – 7.3 • Zelle Sections 7.5 – 7.6	• Decision Statements
F Sept 18		• Problem Day
M Sept 21	• Zelle Sections 5.1 – 5.2 • Zelle Section 5.5	• Strings
W Sept 23	• Zelle Sections 5.3 – 5.4 • Zelle Section 5.6	• Lists
F Sept 25		• Problem Day
M Sept 28	• Zelle Sections 5.7 – 5.8	• List Processing
W Sept 30	• Zelle Section 5.9	• File Processing
F Oct 2		• Problem Day
M Oct 5		• Slip Day / Review
W Oct 7		• Test #1
F Oct 9	• Zelle Sections 8.1 – 8.3	• Looping Statements
M Oct 12		• Fall Break
W Oct 14	• Zelle Section 8.5	• Loop Design
F Oct 16		• Problem Day
M Oct 19		• Loop Practice
W Oct 21		• Loop Practice
F Oct 23		• Problem Day
M Oct 26	• Zelle Sections 6.1 – 6.3	• Functions

Date	**Reading**	**Topic**
W Oct 28	• Zelle Sections 6.4 – 6.5	• Parameters and Return Values
F Oct 30		• Problem Day
M Nov 2	• Zelle Section 6.6 • Zelle Sections 9.1 – 9.3	• Top-Down Design
W Nov 4	• Zelle Sections 9.4 – 9.6	• Bottom-Up Design
F Nov 6		• Problem Day
M Nov 9	• Zelle Sections 10.1 – 10.3	• Defining Classes
W Nov 11	• Zelle Sections 10.4 – 10.5	• Encapsulation
F Nov 13		• Problem Day
M Nov 16	• Zelle Sections 11.1 – 11.2	• Problem Solving with Lists
W Nov 18	• Zelle Sections 11.3 – 11.4	• Lists of Records
F Nov 20		• Problem Day
M Nov 23		• Test #2
W Nov 25 F Nov 27		• Thanksgiving Break
M Nov 30	• Zelle Section 11.6	• Dictionaries
W Dec 2	• Zelle Sections 12.1 – 12.2	• Object-Oriented Design
F Dec 4		• Problem Day
M Dec 7	• TBD	• The Prisoner's Dilemma
W Dec 9		• The Prisoner's Dilemma Competition
F Dec 11		• Review

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