CS 204, Fall 2014
Intro to Computer Science II
Syllabus

<table>
<thead>
<tr>
<th>Instructors</th>
<th>Classes and Labs</th>
<th>Grade Calculation*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Lecture MWF 11-12am, Dana 115 (Minhaz Zibran)</td>
<td>Projects 25%</td>
</tr>
<tr>
<td></td>
<td>Lecture MWF 1-2pm, Dana 115 (Evan Peck)</td>
<td>Quizzes/Homeworks 15%</td>
</tr>
<tr>
<td></td>
<td>Lab Wed 3-5pm, BRKI 164 (Steve Guattery)</td>
<td>Labs 10%</td>
</tr>
<tr>
<td></td>
<td>Lab Thurs 10-12pm, Dana 213 (Steve Guattery)</td>
<td>Exams 30%</td>
</tr>
<tr>
<td></td>
<td>Lab Thurs 1-3pm, Dana 213 (Evan Peck)</td>
<td>Final Exam 20%</td>
</tr>
</tbody>
</table>

* Failure in any one portion of the course grade will result in failing the overall course at the instructors discretion.

Course Overview

Consider all of the tweets on Twitter or all of the status messages on Facebook. For a moment, think of yourself as a programmer for one of these companies. If someone asked you to design a method to search or sort every tweet or message ever sent, how would you begin to think about such a problem? How do you know the difference between an approach that takes 200 milliseconds and 200 years? How would you organize the data in a way that helps you meet your goals? How do you even know where to start?

In this course, we will give you the tools to answer these questions. We will study the organization of data and learn how to analyze the impact of algorithms not only in our own programs, but in programs that must handle thousands, millions, or billions of data points. We will learn about the following topics:

**Standard Data Structures**  A data structure is a particular way of organizing and manipulating data (for instance, Python’s list introduced in CSCI 203 is an example of a data structure). Smart choice of data structures can dramatically speed up program’s execution (and poor choice of data structures may render a program so slow that it becomes unusable). We are going to learn organization and proper use of classical data structures (stack, queue, array, linked list, tree).

**Data Abstraction**  Throughout the course, we will familiarizing ourselves with a very powerful notion of separating abstract properties of data types from their concrete implementations in Python. We will learn to view complex programs as combinations of individual pieces and reason about the behavior of those pieces independently of their actual implementation. This will ultimately help you develop more versatile software that is easy to update and reuse.
**Algorithms and algorithm analysis**  Programs are often judged by how fast they run and how effectively they utilize the available memory space. Interestingly, how fast one program runs compared to another program can be established by looking at their underlying algorithms in theory only (without bothering with the actual implementation, or characteristics of a particular computer on which the programs are run). We will study the way to describe the efficiency of algorithms and evaluate various approaches to implementing data structures and algorithms. This will allows us to make an intelligent choice and alter of a data structure suitable for a particular program.

**Advanced Python programming**  This course builds on the foundation established in CSCI 203 and provides you with further programming experience in Python. You will be implementing the abstract data types that we discuss, and algorithms involving them, and utilize object-oriented programming concepts (such as instance variables, methods, constructors, inheritance, and overriding).

**Course Outcomes**

1. Students will be able to explain and use fundamental data structures, data types, and programming techniques. (ABET a, b, e)

2. Students will be able to explain and use introductory algorithm analysis techniques. (ABET a, b, e)

3. Students will be able to design, implement, and test programs for problems using algorithms and data structures. (ABET a, b, e)

**Locations for course information**

- The course has a Moodle site at [http://moodle.bucknell.edu/course/view.php?id=12829](http://moodle.bucknell.edu/course/view.php?id=12829) where all the information about the course is posted, please check it daily.

- The course textbook is Rance D. Necaise, *Data Structures and Algorithms Using Python*. Wiley 2011

- The course has an accompanying website: [http://www.eg.bucknell.edu/~csci204/](http://www.eg.bucknell.edu/~csci204/)

You should complete any reading assignments by the due date. Be prepared to discuss the reading material in class. Readings are listed on the course schedule.

**Academic Responsibility**

Students are expected to read and comply with the following policies.

- The Course Collaboration Rules policy (on Moodle)


- Bucknell’s Academic Responsibility policy ([http://www.bucknell.edu/x1324.xml](http://www.bucknell.edu/x1324.xml))

If you are unsure, ask your instructor for help.

**Bucknell University Honor Code**

As a student and citizen of the Bucknell University community:

1. I will not lie, cheat, or steal in my academic endeavors.

2. I will forthrightly oppose each and every instance of academic dishonesty.
3. I will let my conscience guide my decision to communicate directly with any person or persons I believe to have been dishonest in academic work.

4. I will let my conscience guide my decision on reporting breaches of academic integrity to the appropriate faculty or deans.

Course engagement

Courses at Bucknell that receive one unit of academic credit have a minimum expectation of **12 hours per week** of student academic engagement. Student academic engagement includes both the hours of direct faculty instruction (or its equivalent) and the hours spent on out of class student work. This course meets 5 hours a week so you should expect to spend at least **7 hours a week** working on it outside of class.

Handing in work

You will submit work on Moodle, unless mentioned otherwise for the particular work. Your grades will be posted on Moodle.

Projects (Individual Writeup)

There will be several programming assignments during the semester. Each project is broken into two phases. Each phase is due by 11:59pm on the due date.

  Late projects will receive a 10% penalty per day overdue. I will not accept projects that are over 3 days late. You get 3 free late days to use on projects as you see fit. You do not have to inform me when you are using a late day, I will assume it when your work comes in late.

Quizzes / Homeworks (Individual Work)

There will be sudden quizzes during the first 15 minutes of your lecture section. Do not be late to class or you will not have adequate time to complete your quiz. You will also be given homeworks (i.e., take-home quizzes), which will also be counted as quizzes. They will cover reading or lecture topics. I will drop your lowest quiz/homework grade.

  If you miss a quiz/homework and I excuse you, that quiz/homework will not count toward your final grade. There will be *no* make-up quizzes/homeworks.

Exams (Open Study Sheet Exams)

There will be two hourly exams during the semester and a final at the end.

Labs (Collaborative Discussion)

You should be able to finish the labs within 4 hours. Labs are due on the following Monday by 11:59pm. The labs will be graded on a 0 to 10 scale. I do not accept late labs. I will drop your lowest lab grade.

  Lab attendance is mandatory. If you do not attend a lab, you maximum grade on it is an 8.

Emergencies, Special Circumstances, etc.

Emergencies happen (medical, family, or otherwise). If one occurs, please contact your Dean as soon as possible. I am very likely to give extensions or excused absenses if you contact me before things are due.

  Sports trips are not an emergency. If you are going on one, you must follow school policy reguarding them. If necessary, I can often give you an assignment early or send an exam with you on the trip.
If you are having a tough semester and discover you have fallen behind, please see me sooner rather than later. While I won’t just excuse you from assignments without a compelling reason, I’m very happy to work with you and provide extra assignments to help you catch up.

If you have been granted accommodations, it your responsibility to contact me in a timely manner to request them. A week before an exam would be considered timely. I am happy to work with you to provide extra time and quiet space, etc as needed.