CSCI 204: Introduction to Computer Science II

Syllabus
Fall 2013

Class

<table>
<thead>
<tr>
<th>Lecture</th>
<th>MWF 11:00 - 11:52 a.m., Dana 115</th>
</tr>
</thead>
<tbody>
<tr>
<td>204.02</td>
<td>MWF 1:00 - 1:52 p.m., Dana 115</td>
</tr>
<tr>
<td>Lab</td>
<td>Wed 3:00 - 4:52 p.m., Brki 164</td>
</tr>
<tr>
<td>204.60</td>
<td>Thurs 10:00 - 11:52 a.m., Dana 213</td>
</tr>
</tbody>
</table>

Instructor

Professor Anastasia Kurdia  
office: Dana 210, phone: 570-577-2345, email: ak034@bucknell.edu  
Office Hours: MWF 2 p.m. - 2.52 p.m. Quick drop-ins are welcome during other times. Questions on class material during labs are very welcome. Extra help is available in person or via Google hangout, by appointment.

Teaching Assistants

- Brigitte Hofmeister bth007 at bucknell dot edu
- Evan Gingrich, ecg009 at bucknell dot edu
- Nick Marshall, nam021 at bucknell dot edu. Nick also conducts a TA help session on Sundays from 5 to 7 pm in Dana 213.

Websites

Course Moodle site (all course information):  
http://moodle.bucknell.edu/course/view.php?id=8795

Course website (public course information, including course schedule):  
http://www.eg.bucknell.edu/~csci204/

Please check course Moodle site, course website and your Bucknell email regularly (at least daily).

Required text


Learning Objectives

Students succeeding in this course will be able to:

1. Explain, utilize, implement and analyze fundamental data structures and abstract data types.
2. Explain and apply introductory algorithm analysis and object-oriented programming techniques.

3. Design, implement, and test programs for problems using algorithms and data structures.

**ABET Student Outcomes Addressed**

- an ability to apply knowledge of computing, mathematics including mathematics appropriate to the discipline, science, and engineering (a);
- an ability to design and conduct experiments, as well as to analyze and interpret data (b);
- an ability to identify, formulate, and solve engineering problems (e).

**Overview: What is this course about?**

**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 be 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 to 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 with 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 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).
Assignments

Reading

Reading assignments are associated with nearly every class and are listed on the course schedule. For the most part it’s up to you whether to complete the reading assignment before or after class (note that every reading assignment should be completed no later than the following class). Occasionally there will be reading assignments essential for in-class discussion, those will be explicitly marked as such and will need to be completed before a corresponding class.

Projects (Individual)

There will be several individual programming assignments during the semester. Each is due at 11:59 p.m. on the specified date via Moodle. Most projects will have an intermediate deadline. The grading scheme for each project along with the scope of work expected by the intermediate deadline will be posted along with the project description and can be used as a checklist when completing the project assignment.

Late projects At the beginning of the semester you receive three late cards, each of which may be used for one 24-hour extension on any one project deadline. If you submit your project after the deadline, you will automatically use a late card. You do not need to tell your instructor or TAs that you’re using a late card. We will notice based on your submission time and automatically deduct a late card. After late card quota is exhausted, late projects will receive a 10% penalty per (non-free) day overdue. No late assignments that are over one week late will be accepted regardless of any free late-days. The feedback for the late work may be delayed as well.

Quizzes (Individual)

There will be frequent closed-book/closed-notes quizzes given during the first 15 minutes of your lab section. Arrive to lab on time, or you will not have adequate time to complete your quiz. The quizzes will cover reading or lecture topics. The lowest two quiz grades will be dropped. If you miss a quiz with permission, that quiz will not count toward your final grade. Make-up quizzes won’t be offered.

Exams (Open Study Sheet Exams)

There will be two hourly exams during the semester and a comprehensive 3-hour final exam at the end of the semester. All exams will be in closed-book/closed-notes format; one handwritten letter-size study sheet will be allowed on each of the exams.

Labs (Collaborative Discussion)

Weekly labs will serve as both lab and homework assignments for that week. You will be starting your work in the lab and finishing it at home. Lab attendance is mandatory. You should be able to finish the labs within 2-4 hours. Labs will be submitted individually, but you are welcome to collaborate with your colleagues during the labs (please see Collaboration Rules). All labs will use Moodle for online submission by 11:59 p.m. of the due date (typically the following Monday except for holidays or semester breaks). The first lab has a written portion; that written portion is due at the end of the first lab, the remainder of that lab is due the first Monday of the semester.

Lab Grading The labs will be graded 1 for satisfactory, .5 for unsatisfactory, or 0 for missing. Typically the feedback for the labs is available within a week from the submission date on Moodle.

Revised labs Students who received an un-satisfactory grade for a particular lab may revise the work and re-submit it once more for the other half credit. If the re-do is not satisfactory, you will not receive the extra .5 points. All lab redos are due a week after we grade the lab. There will be no re-do opportunity on the last lab.
Late Labs  Lab work that is turned in late or is missing will count as a 0 on the initial handin. The maximum re-do score possible in this case is .5 points.

Grading scheme

Course grades will be distributed as follows:

<table>
<thead>
<tr>
<th>Grading</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Projects</td>
<td>25%</td>
</tr>
<tr>
<td>Quizzes</td>
<td>15%</td>
</tr>
<tr>
<td>Labs</td>
<td>10%</td>
</tr>
<tr>
<td>Mid-term Exams</td>
<td>30%</td>
</tr>
<tr>
<td>Final Exam</td>
<td>20%</td>
</tr>
</tbody>
</table>

In order to receive a passing grade for the course, you must receive a passing grade (60% or above) both in the lab/project/quiz portion and in the exam portion. Missing a significant number of classes (over 10%, what equals to 4 lectures) will result in lowered final grade. In order to foster team spirit, grades are not curved: teamwork (within the boundaries of Honor Code) is warmly encouraged in this class, and helping your colleagues succeed won’t negatively affect your chances of receiving good grades. All grades will be posted on Moodle.

Regrade requests  If your grade is less than 100% of points and you have questions why points were deducted, please see your lecture instructor as soon as possible but no later than two weeks after the graded assignment or exam was returned. Timely resolution of all questions will help you avoid making the same mistake in another assignment! If you believe that points were deducted incorrectly, please submit regrade request addressed to lecture instructor via email within two weeks after the graded homework is returned (late regrade requests will not be considered).
Success strategies

Academic Responsibility

Please read the Collaboration Rules policy and the Computer Science Department policy and Bucknell’s Academic Responsibility policy carefully. Each assignment in this course has a specific collaboration policy which is explained in the Collaborations Rules posted on the course website and Moodle site. The Computer Science Department policy is posted on the department website at http://www.bucknell.edu/Documents/Engineering/ComputerScience/student-conduct-policy.pdf. Students are also expected to read and abide by Bucknell’s policy in the Student Handbook at http://www.bucknell.edu/x1324.xml. Please read and follow all policies carefully. If you are unsure, ask your instructor for help.

Use of technology

Standard meeting courtesy rules apply to class meeting times: cell phone notifications should be switched off; cellphones, tablet and laptop computers should be put away, and no unauthorized technology used should occur during lectures and labs. Occasionally we might need to use laptops, tablets and/or smartphones; those instances will be explicitly arranged by the instructor. While in class, you should be 100% committed to learning the subject. As agreed by the class in the beginning of semester, the person who engages in unauthorized technology use will be asked to stand up and sing a song in front of the class (this is an experimental measure and it might be revised if found ineffective).

Emergencies and Special Circumstances

In case of a medical or family emergency please notify your Associate Dean as soon as possible, notify your professor as soon as possible and keep original documentation justifying your special circumstances. You will very likely get extensions or excused absences if your instructor receives Associate Dean’s approval of your extenuating circumstance before the work is due. Conversely, no special arrangements will be made without proper documentation justifying extenuating circumstances that caused you to miss the exam or the deadline.

Sports, performance, conference, job interview trips are not an emergency. If you are going on such trip, you must follow appropriate school policy. If approved, assignment can be given to you early, an exam may be sent to your coach for you to complete on your trip.

If you have a physical, mental or learning disability, either hidden or visible, which may require classroom, test-taking, or other reasonable modifications, please see your Associate Dean as soon as possible. If you have been granted special accommodations, it is your responsibility to contact your instructors in a timely manner (a week before an exam would be considered timely).

Missed work Material of missed lecture is a responsibility of the student. If you missed a class, follow the reading schedule and complete the reading for the missed class; get in touch with your classmates to figure what else you might have missed; resolve questions that you may have about new material with your instructor asap.

Professionalism

This class is one of your first professional environments, so we will define and practise high quality behavior. Behaviors that are of low quality, such as arriving or turning in work late, being unprepared, missing class, texting, shirking your own or group work, or other less desirable behaviors, lay the groundwork for mediocrity. My intent is rather help you lay the groundwork for excellence, and so your professional demeanor is crucial to your success in this class. In other words, you are expected to taking charge of your own learning and behaving as you would in your dream job.

For example, if you didn’t know how to perform a part of your job, you wouldn’t just sit and do nothing for a week. Likewise, if you are having a tough semester and discover you have fallen behind, please visit with your professor and TAs sooner rather than later. I will be very happy to work with you and help you catch up.
You are expected to stay in the classroom while class is in session, and you are allowed to leave the classroom only in the event of emergency. Unfounded leaving of classroom will be treated as unexcused absence.

A friendly reminder of Bucknell University’s expectations for academic 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. Half and quarter unit courses at Bucknell should have proportionate expectations for student engagement.

Analytic approach

To get the most of this class, approach all new information analytically and critically.

When facing a new problem, think how you would solve it. Initially you may not know what to do, or what others have done to solve the problem. First, think where you would start if you really-really had to solve the problem yourself (using the job world analogy, what if the problem appeared on a job interview?) If an elegant solution does not immediately appear, begin by describing a straightforward solution and see if you can optimize its parts. If you absolutely can’t solve a problem in its original formulation, try to formulate a very simple version of the problem that you know how to solve, and then add conditions and assumptions back into the problem formulation, one by one, augmenting your solution every time to reflect the changes.

When learning about new solution, i.e. a new algorithm or data structure, you first goal is to be able to explain to others how it works. Next, try to understand the assumptions and the system for which an algorithm/data structure has been developed, consider the various ways in which the assumptions could be modified, and ask yourself if the solution would still work with the changed assumptions and why.

The questions that were not asked in class will make it to the exams. So be sure to ask a lot of meaningful questions: the more we discuss the more familiar the assignments and tests will look to you! Also feel free to ask questions that might seem too simple - if you don’t, you may have a harder time absorbing more sophisticated concepts.

Important disclaimer

The instructor promises the best effort in adhering to the above rules but reserves the right to change them if deemed necessary. For instance, slight alterations to the course schedule are possible if the class needs more/less time for a certain topic; the midterm exam dates can be changed by unanimous agreement of the class and professor more that two weeks before the exam in question; additional readings may be assigned during the semester as needed; and so on. Updates will be announced in class and posted on Moodle; most current version of the syllabus will be posted on course website.