General Information

Instructor: Ben Schafer
Email: schafer@cs.uni.edu  [Note, please use this address over my official UNI address]
Office: 316 ITTC, phone 273-2187
Office Hours:
  - MWF, 9:00-9:50 AM, 11:00-11:50 AM, 1:00-1:50 PM
    - To reserve a time use:
  - Anytime my office door is open or by appointment

Time and Place: MWF 12:00-12:50, ITTC 328

Class Website: http://www.cs.uni.edu/~schafer/1140/

Course Information:

Course Description

I can hear many of you right now:

“I’m going to be a [high school / middle school] [math / science/ communications/ …] teacher, NOT a computer science teacher. Why should I learn how to program?”

Truthfully, that is a fair question! However, I think I have some good reasons why you might want to take this course and learn how to program.

1) To become more computer literate/savvy. More and more computer programs are being created to allow (or even require) “end user programming.” Your use of these tools will depend on your ability to read, modify, and write basic computer programs. This course is designed to give you the skills to understand the structure and logic of programming.

2) To keep up with your students. Kids are being exposed not only to computers, but to simple programming environments at earlier and earlier ages. As someone studying education you are very likely to end up working with kids who know (or at least think they know) how to program.

3) To learn some of the tools you might get to use on the job. We can’t teach you every tool that you will see on the job. Even if it isn’t your primary job, your ability to help with an after school program might just be the difference that gets you hired at your first teaching job.

4) To make your life easier. The computer is really good at performing repetitive (and often monotonous) tasks. Suppose that you have a bunch of scientific data in a file, or available for you to download off a website. But, unfortunately, it isn’t in the format that you need. Wouldn’t it be great to write your own script that downloads the data and modifies it for you?

CS 1140 is designed to be an introduction to the basic logic and structure of computer programming in environments that a K-12 teacher might use either with their students or for their own personal benefit. It assumes no previous experience with computer programming. While access to a computer outside of a university computer lab will be beneficial, it will not be
required for this course – all course materials are available from any computer with internet access and most programming environments are available in most College of Humanities, Arts, and Sciences computer labs.

**Required Materials**
No single textbook fits our needs. Instead, all required readings and other materials will be selected from legally available resources on the internet or from instructor produced materials. Everything you will need to complete this course is either directly contained within the class website (see above) or is available on the internet from other sources. In the case of materials in this latter category, links to these materials will be provided from the class website. Thus, it will be to your benefit to become familiar with the class website and pay close attention for changes and additions.

**Course Learning Objectives**
By the end of this semester students taking this course should be able to meet the following course objectives:

**Programming-oriented Outcomes:**
Students should be able to:
- trace a segment of code to determine the result produced or state achieved by given code
- modify a provided piece of code to accomplish a given task
- choose and sequence action statements to accomplish a given task
- develop and use selection statements (if-then, if-then-else, etc.) to control selection between actions
- develop and use iteration statements (for, while) to control repetition of actions
- explain the concepts of sequence, loops, parallelism, events, conditionals, operators, variables, and lists within the context of computer science.

**Teaching-oriented Outcomes:**
Students should be able to:
- discuss resources for learning about several programming environments
- discuss which of several programming environments would be appropriate in a given classroom
- explain the concepts of sequence, loops, parallelism, events, conditionals, operators, variables, and lists within the context of a K-12 classroom.

**Course Structure and Policies**
This course consists of the following activities and assessments to assist you in achieving the course and instructional objectives.

**Out-of-class activities (~75 points)** – The goal of these activities is to get you thinking about one or more concepts from the weekly materials. While the exact format may vary they will typically involve having you exploring some existing code or investigating one or more ideas from the preparation materials and considering their application in your future classroom. These activities will normally be completed as an individual.

**In-Class Activities (~150 points)** – The goal of in-class, guided practice activities is to give you an opportunity to practice the material you have been working with. Each week you will be presented with one or more smaller programming activities – often started for you – that you must complete following a set of guided instructions. These are designed to focus on one or more building block concepts from the week’s material and are intended to help you gain practice with the material before you complete a full-blown programming assignment. Think of guided practice as a lab activity in a traditional science classroom.

In almost all situations you will complete in-class activities with a partner. 20% of your grade for that activity is based on my observation of your ability to cooperate with your partner to complete the assignment. If you sit back and let your partner do most of the work you will not receive full credit for the lab. In the event that you are not in class during a practice lab, it is your responsibility to complete the assignment outside of class. In most cases you will not receive points for the 20% of your grade reserved for partner work since you did not work with a partner.

If you do not complete the assignment in class (either because you are absent or because you and your partner do not finish in time) you may continue to work on the assignment. All work is due no later than the START of the next class session.
unless alternate arrangements were made with me prior to that deadline. If you are working with a partner it is expected that you and your partner will schedule a time to get together outside of class to complete the work together.

**Programming Assignments (~250 points)** - While the previously discussed activities can be thought of as “informal” practice, this activity is your main “homework” assignment. Application activities will consist of one or more activities designed to allow you to use the materials you have learned for the week on an activity that you might encounter with students.

Discussion of programming assignments with your classmates should be limited. Discussion should be restricted to understanding the specification for the assignment or asking clarifying questions about the structure/format of a particular command. It should NOT involve looking at other student’s solutions to see how they tackled the problem.

**Final Exam (200 points)** – This exam will consist of both a written and a computer based component. The exact format and nature of the exam will be discussed in the last two weeks of the semester.

**Grading**

Grading for this course is on an absolute scale. All points earned will be summed and an overall percentage calculated. Final grades will be assigned based on cut-off points no “higher” than:

- 90% or above for some sort of A,
- 80% or above for some sort of B,
- 70% or above for some sort of C,
- 60% or above for some sort of D, and
- below 60% for an F.

However, there are a few exceptions to this:

- I reserve the right to adjust these levels if I deem that assigned activities were more difficult than I had expected. That is, a score of 88 is guaranteed to be at least a B+ but it could become an A- under certain situations.
- I reserve the right to lower your final grade in the course by one grade "level" (for example, from a B+ to a B) for each programming assignment for which you fail to submit at least some evidence of effort towards completion. If an assignment has you completely stumped but you gave it some effort, then turn in what you have to show me that you made that effort. I can tolerate confusion; I can't tolerate people too lazy to even attempt the assignment.

**Course and University Policies:**

I try to accommodate student needs whenever possible, but I can only do so if I know about them. If you ever have to make alternate arrangements for some activity please contact me in advance. The safest way to make such arrangements is by notifying me via e-mail or phone of your circumstances and of how you can be reached.

All assignments are due at their assigned date and time. In order to receive partial credit, always submit your best effort at that time. I may accept late work on a limited basis, but you should not expect this to be the case.

Incompletes are awarded only in very rare instances when an unforeseeable event causes a student who has completed all the coursework to date to be unable to complete a small portion of the work in the last week or two of the semester (typically the final project or exam). Incompletes will not be awarded for foreseeable events including a heavy course load or a poorer-than-expected performance. Verifiable documentation must be provided for the incomplete to be granted, and arrangements for the incomplete should be made as soon as such an unforeseeable event is apparent.

**Scholastic Conduct**

You are responsible for being familiar with the University’s Academic Ethics Policies:

https://www.uni.edu/policies/301
Copying from other students is expressly forbidden. Doing so on exams or assignments will be penalized every time it is discovered. The penalty can vary from zero credit for the copied items (first offense) up to a failing grade for the course. If an assignment makes you realize you don't understand the material, ask a fellow student a question designed to improve your understanding, not one designed to get the assignment done. Your final submission for assignments should be individual, original work unless otherwise specified. Any substantive contribution to your solution by another person or taken from a publication should be properly acknowledged in writing. Failure to do so is plagiarism and will necessitate disciplinary action. In addition to the activities we can all agree are cheating (plagiarism, bringing notes to a closed book exam, etc), assisting or collaborating on cheating is cheating. Cheating can result in failing the course and/or more severe disciplinary actions.

Remember: Discussing assignments is good. Copying code or answers is not.

Class Distractions
While you are welcome to own and use electronics such as cell phones, tablets, and laptops, the use of these, and other, electronic devices in the classroom is forbidden without my explicit permission (This is a University-wide policy). A few exceptions do exist, and I reserve the right to approve these situations on a case-by-case basis with prior notification. Unless we have discussed it in advance, all electronic devices should be turned off and left out of sight during class time.

Technology Requirements
Students in this course will rely heavily on the use of the computer. Fortunately, all of the preparation materials and most of the programming environments for this course are available from any computer with a web browser and internet access. Furthermore, all of the required assignments can be completed using software available in most CHAS computer labs or available for free download to a personally owned machine.

If you do not own a computer than you can find appropriate machines (with all the correct software) in several different labs on campus.  While many will work, I suggest the following:

- Wright 339. This is a public lab and is arguably your best bet.
- Wright 112. This is a teaching lab used for several classes and may not always be available.
- ITTC 335 is a small general purpose lounge. It may be crowded, but its advantage is that it is close to my office if you have questions.

Accessibility
The Americans with Disabilities Act of 1990 (ADA) provides protection from discrimination for qualified individuals with disabilities. Students with a disability, who require assistance, will need to contact the Office of Disability Services (ODS) for coordination of academic accommodations. The ODS is located at 213 Student Services Center. Their phone number is 319/273-2676. Additionally, please contact me immediately if you have a learning or physical disability requiring accommodation.

Academic Learning Center Services
You are encouraged you to utilize the Academic Learning Center's free assistance with writing, reading, and learning strategies. UNI’s Academic Learning Center is located in 008 ITTC. Visit the website at http://www.uni.edu/unialc or phone 319-273-2361 for more information.