TCSI201 Honors Introduction to Computer Science

Univ. at Albany Fall 2009 Course Prospectus

4 Honors College Credit Hours: 3 hours lecture and 2 hours laboratory per week. Out-of-class reading, homework, writing and projects. Students will learn how computers and computer software work. In contrast to most other introductory Computer Science courses that use abstract examples to show how software works and should be written, this introductory course will use small personal robots. Students will write programs to control the robots. We will get the robot, with increasing sophistication to sense its environment and to move about, explore and change this environment. As we get the robots to do more and more, we learn more about the computer hardware and software that underlie them. We will develop, practice and improve the skills in solving problems through the use of careful logical thinking, patience and debugging, which is characteristic of CS and programming.

Although numerics and some geometry and algebra will be involved, and these skills are common to mathematics and other physical sciences, little explicit prior knowledge in high-school mathematics or calculus will be required. However, mastery of certain exact rules and logical thinking will be required in order to use them creatively. Besides responsive robot control, some graphics, image processing and elementary data structures and algorithms topics and projects will be included. See samples of materials from similar courses. [http://wiki.roboteducation.org/Educator_Resources]

Using the robots, students will learn three important things. First, they will learn the first semester’s lessons in developing and writing computer software. Second and third, the students will learn group technical communication and teamwork skills through the class exercises, team projects and presentations of solutions. Feedback from employers and potential employers of UA graduates indicates that these are the basis of skills they are looking for in the technically savvy graduates they wish to hire. Specifically, the programming they learn will form the nucleus of skills they need to take the follow-on course (CSI310, Data Structures), and that they will build on throughout the Computer Science curriculum.

Every first CS major/minor course like ours (known as CS1 in our subject’s curriculum standards) teaches principles and practice in the context of programming with a particular programming language and some target problems. Whereas the regular version utilizes the Java language and elementary graphics, the honors version will feature a new robot programmed in the Python language and an associated new curriculum and textbook (see below). We will introduce the core computer science ideas in an attractive but challenging context to talented students who might not have already decided to take more computer science after learning programming before. It will allow more individualized attention to be paid to greater challenges, to presentation and discussions of solutions and of the problem solving process, and to the introduction of somewhat more sophisticated ideas than in a regular version. Besides the concreteness and depth from the robotics, this honors course will include some of the Java language to prepared for our follow-on major and minor courses. The lesson is that core computer science ideas and skills are independent of any particular programming language. The laboratory meetings will provide needed instructional time for efficient usage of the technology and for better debugging and problem solving skills; plus they will enable the lectures to better convey the principles of the computer science disciplinary outlook.

Some innovative teaching methods will be employed, including "team-based learning" to use in class what is learned from readings, with both individual and group quizzes, and in-class group exercises.
Robot Kit—see the images at the end.

Use of the IRPE Robot System for homework, projects and practice outside of the scheduled classes and lab is a required element of this course. The system has the following components:

- The Scribbler Robot. It is most convenient to order this new or used through Amazon using the link from Georgia Robotics. On Amazon, be sure to specify promotion code IOIPREIO to get the 10% student/educator discount, as indicated on the ordering information web page. Although Amazon shows only one unit available, I've been told that many are in stock. Also, buying the textbook is not essential because the digital version is available free.

- The IPRE Fluke board which plugs into the Robot. This also can be ordered new or used, with the discount, from Georgia Robotics through Amazon as above. If you try to get a used Fluke board from say a student or ebay, (rather than from Amazon), make sure the version/revision printed on the board or a sticker is at least 6.2 because older boards fail after being "paired" with different computers multiple times. (All used boards from Amazon are working and up-to-date, and bear the 6.2 sticker on the box.)

- A computer to run the MYRO software with a Bluetooth adapter. Windows or Linux laptops/PCs or MacOSX systems are suitable if the Bluetooth adapter works on the system.

- A preferably 100 foot (class 1) Bluetooth adapter. The AZiO BTD603-132 USB 2.0 Bluetooth Adapter (Class 1 v. 2.0) is recommended because it works with all the above systems and requires no drivers. It can be purchased at Newegg and other online vendors. You probably won't need this if your computer already runs Bluetooth.

- 6 or preferably 12 rechargeable AA NiMH batteries and a charger. Experience shows that two or three sets of alkaline batteries are needed for a semester's work, and the rechargeable ones make the robot work better because they maintain a more constant voltage (for more constant robot speed) while a useful charge remains. (A supply of charged batteries will be maintained in the lab., but you will need your own for out-of-lab work.)

- The MYRO software and Python software. The Windows version comes on the CDROM that is provided with the Fluke board, and Windows, MacOSX and Linux users can freely download and install it by following the instructions in the Myro Installation Manual

Each student should purchase his/her own Robot Kit or work out to purchase and share one kit among two or three classmates. Since all course textbooks and other readings are free, the cost is not out of line. Although students in previous semesters often kept their Robots and Fluke boards, they may be resold personally or maybe ebay, exchanged for gift cards (if they are in working order), and the CS Department might purchase working kits at the end of the semester pending funding and demand, as it did last year.

Some Robot Systems will be available both in the scheduled lab and in the TA or instructor tutorial meetings, but these are reserved for demonstrations, lab exercises, and to a limited extent for use in the tutorial meetings by students whose Robots have broken down and sent for repair or otherwise unavailable. If there is a genuine extenuating circumstance that prevents you from acquiring your own or a shared Robot Kit, or textbooks for other courses, contact the course professor or Dean Haugaard.
We will be using three online books, some online reference manuals, and assigned electronic or library reserve readings in other literature to be announced. They include:

4. Elkner et al.'s *How to Think Like a Computer Scientist: Java Version, 4th edition* is also available free electronically. Our introduction to Java will refer to this to emphasize that learning a 2nd programming language is easy once one has learned a first.

The course web page is www.cs.albany.edu/~sdc/tcsi201 (now under construction.) It is the primary resource for announcements, information, etc. for this course. When in doubt, check the web page. We will also be using Blackboard for parts of the course, such as submitting assignments, online discussions and grade records, and we might use a Wiki.

Class meetings: There are three lectures per week: M, W and F from 9:20 until 10:15 pm in the Arts and Sciences Bldg. rm. Basement 13 (AS B13), and one laboratory meeting per week: Monday from 4:15 to 6:15 in the Earth Sci. and Math. Bldg. rm. Basement 19 (ES B19). These rooms have computer desks. It is very important that you attend class - it is where most of the material that you need to know is presented, demonstrated and discussed. Although there is substantial overlap with the material in the books, additional material and techniques are presented in the classes. *Also, you are responsible for all announcements made in classes.* The laboratory ES B19 will open a fair number of hours per week, to be coordinated with student schedules, for the professor or TAs to provide tutorial sessions where individual students or groups can get hands-on assistance with subject questions, problems or additional exploration. Please contact me for further information or questions, and see you in class!

Professor Seth Chaiken, email: sdc@cs.albany.edu office: LI-96H, hours TBA; Voice: 518-442-4282 (If I'm not in, please leave me voicemail - email may be faster though.)

**Survival in CS courses:** In these subjects, “'knowing how' matters more than 'knowing what' “ [Gian-Carlo Rota], as in exact sciences like core mathematics or physics, or in musical performance, foreign languages, and even sports. Demonstrable skills based on concepts are developed in a sequence in which new ones depend critically on the ability to use previously new ones. In order to pass CSI201 you must take an active part in learning the course material. You must do the readings on time, attend the class consistently, and do your assignments with diligence and patience. This is especially important in an honor's course because of the faster pace. If you do not do these things, you will almost certainly get a weak grade or even fail the course. However, a hint, personal discussion, an explanation in different words, and working in a group work wonders after you learned “what” from reading and listening, and then spent a limited amount of time patiently trying hard to practice “how” to apply it successfully.

The key to success is to stay current. That is, stay up to date in the readings and assignments. If you attend the class, do the preparation and follow-up readings on time, and do the exercises when assigned, you should have relatively few problems. Do not wait until just before a test or assignment is due to try and cram several weeks worth of material in one night. *It will not work* - there are just too many different dependent concepts to master together at once. If you do find that you are having trouble with this course - seek help, SOON!. The longer you wait, the tougher it will be to get back on track. If you wait until the end of the semester, do poorly, and then tell us, there's nothing we can do.
Syllabus Details:

**Grading:** Most assessments here are based on what you **do** rather than on what you **know**. Quality of expression including choice of variable and function names, use of accurate, useful and grammatical comments, etc. is **counted** in programs, exams, written explanations and presentations in addition to logical content, functionality and correctness. The midterms, final and homework assesses individual performance mostly in conceptual problem solving and in programming skills.

<table>
<thead>
<tr>
<th>Assessment</th>
<th>Due Date</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Midterm Exam 1</td>
<td>Fri, Oct. 9, in class</td>
<td>10%</td>
</tr>
<tr>
<td>Midterm Exam 2</td>
<td>TBA, in or Lab.</td>
<td>15%</td>
</tr>
<tr>
<td>Final Exam</td>
<td>Fri, Dec. 11, 3:30PM-5:30 from Univ. Schedule</td>
<td>25%</td>
</tr>
<tr>
<td>Homework</td>
<td>Due on paper in class or online via Blackboard</td>
<td>15%</td>
</tr>
<tr>
<td>Questionnaires, readiness and summary quizzes</td>
<td>in class</td>
<td>10%</td>
</tr>
<tr>
<td>Lab indiv. grades</td>
<td>in Lab meetings</td>
<td>7%</td>
</tr>
<tr>
<td>Lab team grades</td>
<td>in Lab meetings</td>
<td>7%</td>
</tr>
<tr>
<td>Individual or two person project (your choice)</td>
<td>class presentation, video and written report</td>
<td>11%</td>
</tr>
</tbody>
</table>

The final letter grades are based on cutoffs applied to the average score calculated with the weights given above. I will base the exact cutoffs on performance clusterings and the following maximums: A: 90%, B: 80%, C: 70%, D: 60%. (So I have the option to make these cutoffs lower but not higher.) There is no class curving. That is, a pre-defined grade distribution will **not** be used. As a consequence, it is possible for everyone to get an A! Attendance is necessary though mere attendance doesn't count, but each unexcused absence from a questionnaire, readiness or summary quiz or lab. individual quiz results in a zero grade for that assessment. Questionnaires are graded Pass/Not Done.

**Subject Topics (they are cumulative, subject to minor change):**

<table>
<thead>
<tr>
<th>Basics 3 weeks</th>
<th>direct and programmed robot and other operations, robot motors and movement, problem solving, procedures, function return values, expressions, variables, values, basic syntax, iteration, numeric types, conditionals, simple I/O, editing/saving/loading programs, debugging/rerunning, some intellectual structure and goals of Computer Science.</th>
<th>1-5</th>
<th>1-3, 5-7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Robotics 4 weeks</td>
<td>Robot sensors, normalization, non-linear functions, Python lists and strings, complete programs, mathematical models, images, insect and combined reactive behavior, game playing.</td>
<td>6-7</td>
<td>5-7, 8-9</td>
</tr>
<tr>
<td>Graphics Games Java 4 weeks</td>
<td>Graphics, animation, color and sound, image processing, fractals, image analysis, game trees. Beginning programming lang. generalities and Java skills. Recursion. CS articles, classic research or review papers.</td>
<td>8-11</td>
<td>5-8 8-9 +Java vers.</td>
</tr>
<tr>
<td>Moving On 3 weeks</td>
<td>Java practice, object oriented programming basics, topics and projects CS articles, research or review papers.</td>
<td>12</td>
<td>10-17, +Java vers.</td>
</tr>
</tbody>
</table>
Incompletes: As per the Undergraduate Bulletin, the grade of Incomplete (I) will be given only when the student has completed the course requirements at the level of at least C up to and including the first midterm exam (Oct. 9) but because of genuine extenuating circumstances beyond the student’s control that occurred after the drop day, the work is not completed. A student granted an incomplete makes an agreement specifying what material must be made up, and a date for its completion. The incomplete is be converted to a normal grade on the agreed upon date based upon whatever material is submitted by that time. Incompletes will not be given for the purpose of avoiding a failing grade or for improving an anticipated grade when there is no genuine extenuating circumstance.

Make-ups for exams and other activities: Make-up midterms and final exams will be given only for valid and verifiable extenuating circumstances. In such a case, it is your responsibility to contact the professor ahead of time if at all possible. Readiness and summary quizzes, labs and other in-class activities are to be done with the class regardless of previous absences or lack of preparation, even if excused. Such activities missed due to an excused absence will not be made up but they will be excluded from the average score. If you miss a class activity, it is expected that you consult a trusted classmate, and ask the professor for how to get equivalent experience, and study the material after it is posted on the course Web site. An equivalent experience might be scored at the option of the professor. (This policy might be modified in the event of unusual conditions due to such swine flu, etc.) Excused missed homework assignments may be replaced by equivalent ones if answers have been provided.

Extenuating circumstances: Upon request, verification must be presented either directly to the professor or, at the option of either the professor or the student, via the Office of Undergraduate Studies or other UA officials.

Disabilities, etc: Accommodations are made for clients of the Office of Disabled Student Services upon adequate prior notice and according to that office's policies. Students with genuine continuing hardship situations, or any disability related problems with Lab or computer usage should consult that office and confer with the professor before October 16.

Student Responsibilities and Cheating: On individual, written homework assignments: DO the problem solving, coding, testing, debugging, refining ON YOUR OWN STEAM, with consulting BUT NOT COPYING. When you get help: Question until you (think you) understand, then DO IT YOURSELF on blank paper or a new file. (Then, you'll know for sure. If you ever get a tutor or other helper for this or subsequent courses, demand that he or she explains things until you understand enough to solve the assigned problems yourself, and not just tells you the answers.) If you are sharing a Robot Kit or even a computer, you must not share, copy or look at your partner's homework files. Any individual homework files on our lab or the local disk of Information Commons computers should be deleted (after they are backed up!) so that others cannot copy them.

Cheating is intellectually dishonorable, inhibits your learning and makes grading unfair. I will try to inquire and warn you if I see some similarities in submitted homeworks that suggest copying. Major or repeated copying, from classmates or others, will result in a zero score for the assignment. Cheating on a midterm or final exam, or deliberate computer hardware, software or network vandalism will result in course failure. A report of the every cheating incident will also be made to the Office of Undergraduate Studies in accordance with the University regulations concerning "Penalties and Procedures for Violations of Academic Integrity" in the Undergraduate Bulletin, and may be referred to the University Judicial System. Violations of Computer and Network Usage policy on laboratory and other University systems are regarded as academic integrity violations, like cheating.

Other responsibilities: Do assigned readings and exercises on time. You will be responsible for learning from readings some motivational and some factual background and some concepts that might be tested on quizzes or exams, and some details needed for assignments that will NOT be gone over in class. Besides doing readiness quizzes, you are expected PROVIDE the class and professor with QUESTIONS about what you need to know or what you find confusing. This is important for an honors and active learning style course to be effective. I've been thinking in terms of both principles and practices of programming (among other things) for some 40 years. It's hard to tell what beginners find easy or hard, or how long they need to use something obvious to me.