Computer Science 1323

Fall 2017

Class Time: 3:00-4:15 MW  
Location: Dale 112  
Prerequisite: Math 1523 or equivalent or concurrent enrollment, some prior programming experience.  
This course is appropriate for students with some prior programming experience.

Instructor: Dr. Deborah A. Trytten (rhymes with mitten)  
Office: Devon 252  
Office Phone: 325-4299 (email usually gets a quicker response)  
Personal URL: http://www.cs.ou.edu/~trytten  
Email: dtrytten@ou.edu  
Teleconference: My Skype ID is deborah.trytten. You may Skype to office hours if you wish.  
My office hours sometimes have to be changed during the semester.

| My Office Hours |
|-----------------|-----------------|
| Day             | Time            | Location       |
| Wednesday       | 7-8 p.m. (starts August 29) | Devon 145 |
| Tuesday         | 11 a.m.- 12 p.m. | Devon 252 |
| Friday          | 1-2 p.m.        | Devon 252 |

Appointments for additional office hours are scheduled through email, preferably 24 hours in advance.

Required Materials: You must purchase the following items immediately (all contain graded assignments that cannot be completed any other way):

- TuringsCraft Java CodeLab package (http://www.tcgo1.com or http://www.tcgo2.com and click Register for CodeLab). The access code is OKLA-25802-VJGW-34. This software purchased over the Internet directly from the company. If you do not have a credit or debit card available, call Turingscraft and they will turn on your account while you mail a check.
- iClicker polling software on a device. If you change devices during the semester, remember to register your new device. People with unregistered remotes will not get credit for their in class work.
- Working laptop computer with 2 hours of battery life, and wireless network access. We will use computers during class time every day. You are expected to bring your laptop to class.

Topics: Programs, Java, input and output, identifiers, variables, assignment statements, constants, memory diagrams, primitive data types, operations on primitive data, conditional statements, repetition, methods, parameters, arguments, return values, nested control statements, one dimensional arrays, objects, introduction to user defined classes, and classes from the Java Application Programmers Interface (API) (including Arrays, ArrayList, Character, Collections, Double, Integer, Float, Math, Scanner, String, StringBuffer, and StringBuilder).
In this class, students will increase their ability to meet the following ABET outcomes:

- **Outcome B**: Analyze a problem, and identify and define the computing requirements appropriate to its solution.
- **Outcome C**: Design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs,
- **Outcome K**: Apply design and development principles in the construction of software systems of varying complexity.

**Teaching Assistants and Tutors**
All teaching assistant office hours will be held in Devon 115. All TA office hours are open to students from all sections.

Venala Anugu ([vanugu@ou.edu](mailto:vanugu@ou.edu))
Khaled Jabr ([khaled.jabr-1@ou.edu](mailto:khaled.jabr-1@ou.edu))
JoAnn Le ([joann.h.lee@ou.edu](mailto:joann.h.lee@ou.edu))
Lauren Wells ([laurenbrookewells@ou.edu](mailto:laurenbrookewells@ou.edu))
University College Action Center Tutor: Noah Zemlin ([noah.zemlin@ou.edu](mailto:noah.zemlin@ou.edu))

<table>
<thead>
<tr>
<th>Time</th>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
</tr>
</thead>
<tbody>
<tr>
<td>9 to 10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10 to 11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11 to 12</td>
<td>Lauren</td>
<td>Tryptten</td>
<td>Khaled</td>
<td>Vennala</td>
<td></td>
</tr>
<tr>
<td>12 to 1</td>
<td>Vennala</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 to 2</td>
<td></td>
<td></td>
<td>Lauren</td>
<td>Tryptten</td>
<td></td>
</tr>
<tr>
<td>2 to 3</td>
<td></td>
<td></td>
<td></td>
<td>JoAnn</td>
<td></td>
</tr>
<tr>
<td>3:30 to 4:30</td>
<td></td>
<td></td>
<td></td>
<td>Noah</td>
<td></td>
</tr>
<tr>
<td>4:30 to 5:30</td>
<td></td>
<td></td>
<td></td>
<td>Noah</td>
<td></td>
</tr>
<tr>
<td>6-8 starting August 29</td>
<td>Noah</td>
<td>Noah</td>
<td>7-8 with Tryptten</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Locations**
- Devon 115
- Devon 252
- Devon 145
The School of Computer Science also provides William Kerber Teaching Scholars to support learning in CS classes. The hours are below. All of these sessions are in Devon 115.

Sunday: 12:30 – 7  
Monday and Wednesday: 9:00-1:00, 3:00-7:30  
Tuesday and Thursday: 10:30-12:30, 3:30-7:30  
Fridays: 9:00-1:00

Useful Links  
This course is run from http://canvas.ou.edu. Login with your OU 4x4.

Free Software Tutorial Products  
We will use a number of software products to help you learn to program. You need to login to all products in order for grades to be recorded. When asked to enter your name, please use the name that appears in official University records and fill in all fields.

Coding Bat http://codingbat.com/ requires you to create an account of your own. Please give your name in the account and not just your email. Share your responses with cs1323-fall2017@cs.ou.edu so we can get your grades recorded properly. Also, remember to login when you work exercises—it isn’t required and if you are not logged in then your grades will not be recorded.

Course Policies  
Platform: You are responsible for things posted on Canvas or sent by email with a 24 hour delay.

Examinations: There will be three midterm examinations and a final examination. Missing an examination without a previously approved excuse will result in a grade of zero for that examination. Makeup examinations are only available when required by University policy, in other words, almost never. Midterms are currently scheduled on Monday, September 18, Monday, October 23, and Monday, November 20. Midterm examinations are 50 minutes long.

Final Examination: The final examination date is 4:30-6:30 Wednesday, December 13. The final is comprehensive, as required by College of Engineering policy. No final examinations can be given early, except as required by University policy. The final examination is two hours long.

iClickers: I will use iClickers in class every day to gauge student learning and track attendance and participation. Bringing someone else’s device to class and entering answers for them is academic misconduct. I will do integrity checks during the semester to ensure that people are using only their own clickers. iClickers use geolocation to assure that devices are in the classroom when clicking.

Laptop Computers: It is the responsibility of each student in this class to have a working laptop computer available for every class. If your computer requires repair during the semester, it is your responsibility to make arrangements to have another computer available and get the necessary software installed before the class time. A student without a fully usable laptop computer will be at a severe disadvantage in this class.

Adjustments for Pregnancy/Childbirth Related Issues: Should you need modifications or adjustments to your course requirements because of documented pregnancy-related or childbirth-related issues, please contact me as possible to discuss. Generally, modifications will be made where medically necessary and
similar in scope to accommodations based on temporary disability. Please see
http://www.ou.edu/eoo/faqs/pregnancy-faqs.html for answers to commonly asked questions.

**Religious Observances**: It is the policy of the University to excuse absences of students that result from
religious observances and to provide without penalty for the rescheduling of examinations and additional
required class work that may fall on religious holidays. Please check the schedule and inform me of
conflicts as soon as possible.

**Title IX Resources**: For any concerns regarding gender-based discrimination, sexual harassment, sexual
misconduct, stalking, or intimate partner violence, the University offers a variety of resources, including
advocate on call 24/7, counseling services, and mutual no contact orders, scheduling adjustments and
sanctions against the perpetrator. Please contact the Sexual Misconduct Office 405-325-2215 (8-5) or
the Sexual Assault Response Team 405-615-0013 (24/7) to learn more or to report an incident.

**Academic Integrity Violations**: The University of Oklahoma defines an integrity violation to be any act
that improperly affects the evaluation of a student's academic performance or achievement. The
Student's Guide to Academic Integrity (http://integrity.ou.edu/students.html) gives examples: “cheating
on examinations with cellphones, notes, or neighbors; plagiarism, improper collaboration on
assignments intended for individual completion.”

The most common violations in this course are plagiarism, usually on laboratory assignments and
homework. Plagiarism is defined (http://dictionary.reference.com/browse/plagiarism?s=t) as “an act or
instance of using or closely imitating the language and thoughts of that author’s work as one’s own, as
by not crediting the original author.” I avoided committing plagiarism by putting words in quotes and
citing the source in the previous sentence. However, this mechanism does not work on homework and
laboratory assignments since it only changes one form of academic misconduct (plagiarism) into another
(improper collaboration on assignments intended for individual completion).

When you pass this class with a grade of C or better, I am certifying to the world that you are a
competent Java programmer. I cannot make this certification without seeing work that you did on your
own. Interactive programming tutors, homework and examinations should be the work of a single
individual, not their friends and not their tutor. Although I can’t really believe I have to say this, the
solutions to assigned work should not be copied from internet sources, including cheat sites and paid
professional programmers. Remember, I can do internet searches too.

1. Do not show, give, or email another student a copy of your work before the submission
deadline. Every semester I have multiple students submit another student’s work as their own
with the other student’s name still on it. Do not trust other students to not do stupid things that
will get you in trouble.

2. The penalties for permitting your work to be copied are usually the same as the penalties for
copying someone else’s work because it is not always possible for me to distinguish the person
who copied from the person who allowed his or her work to be copied. In cases where I can
make the distinction, the person who copied the work will have a more severe sanction.

It is permissible to talk to other students in the class to get help completing or improving your work as
long as this help does not interfere with my ability to properly evaluate the quality and quantity of your
understanding of computer programming. To understand the distinction, review the examples in the
table below. These are typical examples and are not intended to be a comprehensive list of all of the
ways in which academic integrity can be or not be violated.
<table>
<thead>
<tr>
<th>Situation</th>
<th>Integrity Violation?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Students A and B meet and work on their homework together. Neither student prepared anything in advance and the resulting work is more or less identical.</td>
<td>Yes</td>
</tr>
<tr>
<td>Students A and B create drafts of their homework assignment independently and get together to compare answers and discuss their understanding of the material. Each person decides independently whether to make changes that are discussed.</td>
<td>No</td>
</tr>
<tr>
<td>Students A and B agree to prepare drafts of their homework assignment independently, but only Student A does. Student A shares his draft to Student B who reviews it and offers suggestions for improvement.</td>
<td>Yes</td>
</tr>
<tr>
<td>Students A and B agree that student A will work the even problems and student B will work the odd problems. They share their work.</td>
<td>Yes</td>
</tr>
<tr>
<td>Student A has completed a project and is helping student B complete the same project. Student A explains to student B what student B’s code actually does, which is different than what student B thinks the code does. Student B determines how to modify the code independently.</td>
<td>No</td>
</tr>
<tr>
<td>Student A has completed a project and is helping student B complete the same project. Student B is having trouble getting one part of the program to work, so student A texts student B three lines of their solution.</td>
<td>Yes</td>
</tr>
<tr>
<td>Student A has completed a project and is helping student B complete the same project. Student B is having difficulty getting the program to work, so student A tells student B exactly what to type for several lines.</td>
<td>Yes</td>
</tr>
<tr>
<td>Student A has completed a project and is helping student B complete the same project. Student B is having difficulty getting the program to work, so student A suggests that student B use a specific debugging strategy (e.g. “Print out the contents of the variable”).</td>
<td>No</td>
</tr>
<tr>
<td>Student A has completed a project and is helping student B complete the same project. Student A shows student B an example program in the online textbook that will be helpful in figuring out the solution to the problem.</td>
<td>No</td>
</tr>
<tr>
<td>Students A and B work on a project together. After they have finished it, student A takes the code and modifies it so the programs do not appear to be identical.</td>
<td>Yes</td>
</tr>
</tbody>
</table>

*Please be aware that I have software at my disposal that can detect these kinds of changes, so this strategy is likely to be detected.

**If you work with anyone else in completing an assignment, you must include that person’s name on the submitted work.** Failure to list a student you worked with on the assignment is a violation of academic integrity.

I sometimes use automated software to determine when student work is overly similar. The results of using this software are then evaluated manually by the instructor before any academic integrity violations are filed.

Upon the first documented occurrence of academic misconduct, I will report the academic misconduct to the Campus Judicial Coordinator. If you are found to have committed academic misconduct by this process, the least penalty is usually failing the class and being suspended from college for a semester. If you have committed academic misconduct previously, the sanctions can be more severe. The procedure to be followed is documented in the University of Oklahoma Academic Misconduct Code. In the event that I elect to admonish the student, the appeals process is described here: http://integrity.ou.edu/students.html.
Tutors and Academic Integrity: Private tutors can be a source of support for students who are having difficulty in the class, but only if the tutor is aware of the distinction between teaching you the material so that you ultimately can do your own work, and doing work for you. Tutors who do work for you are not only failing to help you learn, they are committing academic misconduct. All of the situations in the table of academic integrity scenarios above apply equally when student A is a tutor.

Ownership of Course Materials: The instructor retains ownership and all rights to original content. This includes but is not limited to exams, lectures, quizzes, handouts, protocols, electronic documents, syllabi, and all other materials. Original or transcribed course content may not be copied, recorded, retransmitted, posted on-line, or sold without the expressed written consent of the instructor.

Incompletes: The grade of I is intended for the rare circumstance when a student who has been successful in a class has an unexpected event occur shortly before the end of the class. I generally will not consider giving a student a grade of I unless the following three conditions have been met.

1. It is generally within two weeks of the end of the semester.
2. The student has a grade of C or better in the class.
3. The reason that the student cannot complete the class is properly documented and compelling.

Accommodation of Disabilities: The University of Oklahoma and I are committed to providing reasonable accommodation for all students with disabilities. Students with disabilities who require accommodations in this course are requested to speak with me as early in the semester as possible. Students with disabilities must be registered with the Office of Disability Services prior to receiving accommodations in this course. The Office of Disability Services is located in Goddard Health Center, Suite 166, phone 405/325-3852 or TDD only 405/325-4173.

Projects, Homework, Attendance

Late Work: I do not accept late work. Flat tires, parking problems, trips (even if academic), alarm clock failures, personal illness, internet outages, doctor’s appointments, and dependent care are not acceptable excuses for submitting late work.

Excused Absences: The grading policies in the course are designed to allow students to miss one full week of class without a grade penalty. There will be no further excused absences for any event of duration shorter than a week, unless that activity is an official university sanctioned activity (student organization activities are not included). If you have a properly documented absence that is longer than one week, work can be excused with proper documentation.

Computer Literacy Expectations: A list of specific expectations for computer literacy is posted on Canvas. If you do not meet these expectations, the TAs and I will help you remedy this situation immediately or drop this class.

Projects: Incomplete projects may be turned in for partial credit. Projects which do not compile will generally receive no credit. Projects that do not execute to completion also generally will receive no credit (beyond the first two projects).

Project Submission: The .java file(s) (not the .class files) should be exported from eclipse, uploaded to Canvas. A single folder/file should be submitted.
**Backup Copies of Projects:** It is your responsibility to back up your files appropriately. No extensions to deadlines will be given as a result of lost files, unless there is a massive, network wide problem which affects the entire class. Dropbox or other cloud services are also useful for this if you have reliable internet access. It is particularly important to save a backup copy of any homework or laboratory project that is submitted. This backup version should not be opened or edited after submission in case something goes wrong with the submission (like submitting the .class files instead of the .java files, a very common error).

**Homework Submission:** Homework is due by 11:59 p.m. on the selected due date and submitted on Canvas in a single, well formatted document that contains your name and has the problems in the original order in PDF format. Homework must be word processed. Photographs of hand written or word processed work are not acceptable.

**Evaluation**

**Grade Summary:** Canvas has a grade book that is used to store the raw data that is used to calculate your course grade. It is the responsibility of each student in this class to check their grades periodically and make sure that they are properly recorded. If an error is found, bring the grading document to me, and I will correct it.

**Submission and Formatting Failures:** Submitting files on Canvas is a two-step process. First the file is uploaded, then submitted. Each student will be forgiven for failing to hit submit once during the course of the semester. Each student will also be forgiven for submitting assignments in the wrong format only once (usually submitting a Word document instead of a PDF file, or submitting .class files instead of .java files).

**Grading:** There are 6 components to the course grade. They are weighted as follows. The percentage of the grade that comes from interactive tutors, homework and laboratory activities is designed to be small to allow students to make mistakes and learn from them with only small penalties. However, completing these exercises is how most students develop the conceptual understanding that make it possible to do well on the homework, midterms and final.

<table>
<thead>
<tr>
<th>Item</th>
<th>Weight</th>
<th>Forgiveness Policy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Turingscraft/Javabat</td>
<td>10</td>
<td>100 free points (not to exceed 100%)</td>
</tr>
<tr>
<td>Zyante*</td>
<td>5</td>
<td>100 free points (not to exceed 100%)</td>
</tr>
<tr>
<td>iClicker Questions*</td>
<td>0-5</td>
<td>2 lowest days</td>
</tr>
<tr>
<td>Project assignments</td>
<td>20</td>
<td>1 lowest</td>
</tr>
<tr>
<td>Homework</td>
<td>10</td>
<td>1 lowest</td>
</tr>
<tr>
<td>Midterm Examinations</td>
<td>30</td>
<td>None</td>
</tr>
<tr>
<td>Final Exam</td>
<td>20-25</td>
<td>None</td>
</tr>
</tbody>
</table>

+ Each Turingscraft exercise is worth one point. Each Javabat exercise is worth 2 points.
* iClicker grading is based on the number of days when questions are assigned that you achieve at least 50% success on the clicker questions given in class. Each day one point is given for attendance to students who answer any iClicker question (whether correctly or not). The first two weeks of class will not be counted. If the questions are particularly hard on a given day, a threshold below 50% may be used at the discretion of the instructor.
^Zyante grades are based on the number of problems answered correctly before the deadline for participation activities only. The challenge exercises are not graded. They can be used to get additional practice.
#The grade will be calculated twice: once with iClickers counting and one without. The higher grade will be the grade of record.

The grading scale will be no higher than the following. It may be lower at the discretion of the instructor.

<table>
<thead>
<tr>
<th>Grade</th>
<th>Percentages</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>90+</td>
</tr>
<tr>
<td>B</td>
<td>80-89</td>
</tr>
<tr>
<td>C</td>
<td>70-79</td>
</tr>
<tr>
<td>D</td>
<td>40-69</td>
</tr>
<tr>
<td>F</td>
<td>Otherwise</td>
</tr>
</tbody>
</table>

**Borderline Grade Decisions:** Although it would be preferable that all grades are cleanly decided, it is usually the case that a few final course grades are decided by only a few points. I have an algorithm for determining grades in these difficult cases. A grade is a borderline grade if it is within two points of the next higher grade. Therefore, grades like 69 and 78 are borderline grades, but grades like 81 and 92 are not. The grade on the final examination will be used to determine borderline grades. If the grade on the final is below the threshold for the higher grade, the lower grade will be given. If the grade on the final is above the threshold for the higher grade, the higher grade will be given.