



Course Syllabus for CS 235 Java Programming Fall 2015

Basic Course Information:

| | | | |
|--|--|---------------------|---------|
| <i>Instructor:</i> | Sharon Tuttle | | |
| <i>Lecture time and location:</i> | Monday | 5:00 pm - 6:50 pm | FR 107 |
| <i>Lab time and location:</i> | Wednesday | 5:00 pm - 6:50 pm | BSS 313 |
| <i>Instructor's office:</i> | BSS 322 | | |
| <i>Instructor's e-mail:</i> | st10@humboldt.edu or sharon.tuttle@humboldt.edu or smtuttle@humboldt.edu | | |
| <i>Instructor's office phone:</i> | (707) 826-3381 | | |
| <i>Instructor's office hours:</i> | Monday | 3:00 - 4:30 pm | |
| | Wednesday | 11:00 am - 12:00 pm | |
| | Thursday | 1:00 - 2:00 pm | |
| | Friday | 3:30 - 5:00 pm | |
| | or by appointment | | |
| <i>Course public web page:</i> | follow link from: http://users.humboldt.edu/smtuttle/ or follow link from course Moodle site | | |

Course Description:

[from the HSU catalog]: Object-oriented programming; event handling; abstract windowing toolkit applets, applications; Java database connectivity; applications programming interface and javadoc.

An introduction to Java, a highly-portable object-oriented programming language particularly suited to programming for a variety of platforms. This is not an introductory programming course -- it is really a course in Java as a "second" language, an introduction to Java in particular for those already familiar with object-oriented programming using C++. (That said, if you are familiar with object-oriented programming in some other language, this course should also be suitable for you, although you should be aware that there will be frequent references to/comparisons with C++.)

Some of the Java-centric topics to be covered include layout managers, graphical components, the Java event model, threads, exception handling, IO streams, and of course a useful subset of Java classes. Swing will be used to build Java applications and applets, although some aspects of the AWT (abstract windowing toolkit) will also be covered. In addition, we will be discussing software testing in this course.

Students are expected to come into this course already comfortable with object-oriented programming and C++. **No** prior knowledge of Java is assumed.

Course Prerequisite:

CS 112, or instructor's consent.

Course Objectives:

After successfully completing this course, students should:

- be knowledgeable enough about basic Java language syntax and semantics to be able to successfully read and write Java computer programs;
- be comfortable using the Java Application Programming Interface (Java API) to learn about and then to successfully and appropriately use an already-written Java class;
- have obtained experience designing, implementing, testing, and debugging graphical user interfaces that respond to user events using Java;
- have obtained experience designing, implementing, testing, and debugging Java classes and applications.

CS Program Learning Outcomes that this course addresses:

This course addresses departmental learning outcomes of:

- Computational Thinking
- Technical Writing
- Communicating and Collaborating

This course addresses computational thinking by providing experience writing Java programs. It addresses technical writing via program documentation and coding standards that stress reusable code, and it addresses communicating and collaborating via experience pair-programming in course lab sessions.

HSU Learning Outcomes that this course addresses:

This course explicitly contributes to students' acquisition of skills and knowledge relevant to HSU Learning Outcomes:

HSU graduates will have demonstrated:

- Effective communication through written and oral modes.
- Critical and creative thinking skills in acquiring a broad base of knowledge and applying it to complex issues.
- Competence in a major area of study.

HSU graduates will be prepared to:

- Succeed in their chosen careers.

Required Course Materials:

- Turning Point RF Response Clicker, available at the campus bookstore
 - BEWARE!! My understanding is that HSU does NOT support Responseware, EVEN THOUGH the TurningTechnologies module in Moodle may MENTION it!
- SO -- I would recommend that you DO NOT PURCHASE Responseware for this course, since students have

told me they do NOT give refunds if you are unable to use it!

- "Core Java™, Volume I - Fundamentals", 9th Edition, Cay Horstmann, Gary Cornell, Pearson Education, 2011, ISBN 978-0-13-708189-9
 - NOTE: while this is currently NOT available in electronic form from the HSU Library's version of Safari, it IS available in electronic form from the Association for Computing Machinery's (ACM's) version of Safari (!?). So, if you happen to be a student member of the ACM, you can read this electronically in that way.
- Any additional required readings will be made available either via on-line links or via resources available through the HSU Library such as the ACM Digital Library and Safari TechBooks Online.

Course Software:

We will use Java 8 this semester. The course text uses Java 7, but, happily, there should not be significant differences between the source code for Java 7 and Java 8 for the Java features we will be using in this course.

(Why not use Java 7 in this course, then? Because while the source code differences will likely not be significant, Java source code that is compiled using a Java 8 `javac` program cannot be run on a Java 7 Java Virtual Machine (JVM). And, at this point, it is easier to find and install Java 8 than Java 7.)

Note that some use of the UNIX operating system will be required for this course, as will some use of such programs as `ssh`, `sftp`, e-mail, and a web browser, along with the Java JDK (Java Development Kit) and at least one Java IDE (integrated development environment).

Java JDK

To download a version of the Java JDK for Java 8 (for free) for your personal computer running Windows, Mac OS X 10.7.3 or later, or Linux, go to:

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

...and download JDK 8. (I'm hoping that, for our purposes, any Java JDK version 8 should be fine -- you should let me know if there are any issues that arise over the course of the semester.)

(Note that the Java you can download at <http://www.java.com> is **not suitable** for this course -- that is the JRE (Java Runtime Environment), which is for end-users who want to run already-written Java programs, rather than for those who would like to program in Java. **Programmers need the JDK.** Java is rather cursed with a myriad of acronyms!)

Java IDEs

We will also be making use of at least one Java integrated development environment (IDE) during the semester; these "sit atop" the Java JDK, and make Java program development more convenient. There are a number of high-quality free Java IDEs available; all those used in this class will be free Java IDEs with versions for Windows, Mac OS X, and Linux.

In class, I will frequently use the DrJava IDE for projected in class examples; this can be downloaded from:

<http://www.drjava.org>

(There is a kluge for persuading DrJava being run on Mac OS X to use Java 8 -- I will provide instructions for this for Mac users.)

I may also occasionally use the BlueJ IDE in class; this can be downloaded from:

<http://www.bluej.org>

(and follow the **download** link)

There is also a chance that we may experiment with other Java IDE's, most probably NetBeans and/or Eclipse. These are still free, but are also more-professional-level IDEs, with a definite learning curve and more sophisticated features. NetBeans can be downloaded from:

<http://netbeans.org/downloads>

...and Eclipse can be downloaded from:

<http://www.eclipse.org/downloads>

Other course software

We may use some additional Java libraries; if we do, you will be provided with instructions for how to access these.

We may experiment with submitting homework programs to Web-CAT at Virginia Tech, but I will give you instructions for that if it comes to pass. In the meantime, submitting course work is probably going to be a **three-step process**.

- Step 1 is to transfer your files to `nrs-projects.humboldt.edu` using `sftp` (secure file transfer).
- Step 2 is making sure that the way your Java IDE handled tabs did not make your code unreadable (and fixing the indentation if there are any issues!). You will need to use `ssh` (secure shell) to do this.
- Step 3 is to use the course submission tool (which will be described in-class). You will need to use `ssh` (secure shell) to run the course submission tool, also.

`ssh` and `sftp` are already available in on-campus labs, and you should be able to download one of several versions for free for your outside-lab use.

Campus labs run by Academic Computing that contain computers running Windows (such as BSS 317) have PuTTY installed to provide a GUI implementation of `ssh` and WinSCP installed to provide a GUI implementation of `sftp`. Also, command-line versions of `ssh` and `sftp` (usable from the Mac OS X Terminal command-line) are installed already as part of Mac OS X.

This class's lab sessions are held in a lab, BSS 313, that has different versions of these than campus academic computing labs. An introductory tutorial, including some screenshots, for the graphical Windows version of `ssh` and `sftp` installed in BSS 313 can be found at:

<http://sils.unc.edu/it-services/servers/using-ssh>

Clickers:

We will be using Turning Technologies student response clickers in class. There is significant literature indicating that using clickers may increase student engagement and success in learning.

Students purchase this clicker (they are available from the campus bookstore); purchased clickers can be returned at the end of the semester for a partial refund of the purchase price or they can be kept for use in future HSU courses. You register your clicker at the beginning of each semester by logging into Moodle and going to the course Moodle site; in the lower left or right columns, you should find a block labeled "Turning Technologies" -- if you click on the link in that block, you should be able to enter your clicker code (consisting of 6 characters/digits) from the back of your clicker. You then bring your clicker to every class meeting (lectures and labs).

These clickers will be used for in-class questions, which will be interspersed within class meetings. These will usually be given in a **think-pair-share** fashion, in which you answer a question first individually, and then discuss your answer with another student, discussing why you think your answer is correct; if they gave a different answer, you try to persuade them that yours is the correct answer, and then either of you can change your answer if you wish. The response system will record the overall class response percentages as well as keep track of individual answers.

Typically, you will receive:

- **2 points** for a correct answer,
- **1 point** for an incorrect answer, and
- **0 points** for no answer,
- but with a **maximum** semester clicker-questions grade of **120**.
- (There may be some no-point questions from time-to-time as well -- such questions will be noted if/when they come up.)

Thus you will be rewarded for regular attendance and participation. If you miss a class session, you miss that day's

clicker questions and cannot make them up (except for extraordinary circumstances). However, there will be a sufficient number of questions asked to allow for the possibility of extra credit (up to a **maximum** clicker grade of **120**) or to make up for a day that you are out due to illness (although note that you are still responsible for finding out what you missed on such days).

If you forget your clicker for a class meeting, then **up to 5 times** you may still receive some clicker credit, **usually minus a 2-point penalty**, by e-mailing me your clicker answers for that day, **by midnight on that day**, using a Subject: line of: Subject: CS 235 Clicker Answers for <date>. Later e-mails, or e-mails without the proper Subject: line, might not be accepted for credit.

The idea is that the clicker questions will help you to see if you are starting to understand concepts being discussed; sometimes they will also provide review of concepts discussed previously. Clicker questions are typically quite different from exam questions (since clicker questions are typically multiple-choice questions, while exam questions will rarely be multiple-choice). They still enable you to get some immediate feedback regarding whether you are grasping course concepts, whether you need to pay more attention to course discussions and/or readings, etc. They may even help me to know what concepts might need more explanation in-class.

I hope to run tests of the system during the first week's class meetings, and hope to begin asking questions that "count" during the second week's class meetings. Therefore, you must purchase your clicker and register it as soon as possible. If there is an issue with this (for example, if the bookstore runs out of clickers), contact me immediately.

Finally, please note that use of another CS 235 student's clicker, or having someone else use your clicker in a CS 235 class session -- that is, pretending that someone is in class who actually is not -- is considered to be **cheating**, with the same policies applying as would be the case if you turned in someone else's work as your own or permitted someone else to copy your work. Please **ASK ME** if you are not sure what I mean by this.

Grading Breakdown:

If you are a Computer Science (CS) major, note that you must earn at least a **C-** in CS 235 for this course to count as a major elective.

Your semester grade will be determined by the percentage of points that you earn, **subject to some minimum requirements**. Here are the grade percentages, followed by those minimum requirements:

| | |
|------------------------------|---|
| Homework assignments: | 25.0% |
| Lab exercises: | 12.5% |
| Clicker questions: | 12.5% |
| Exams: | |
| Exam 1: | 15.0% |
| Exam 2: | 15.0% |
| Final Exam: | 20.0% Monday, December 14, 5:00 - 6:50 , FR 107 |

(CS 235 is using the MW 1800 Final Exam time, while REC 210, which meets in FR 107 on Wednesdays, is using the MW 1700 Final Exam time)

Grade Requirements:

- To earn a grade of **C- or better** in this course, the following three requirements must **all** be met:
 - your overall semester average must **equal or exceed 70%** - this is to show a reasonable level of overall mastery of the course material.
 - the **average** of your Exam 1, Exam 2, and Final Exam grades must **equal or exceed 60%** - this is to show that you understand at least a minimal reasonable level of the most important course concepts.
 - the **average** of your Homework assignments must **equal or exceed 60%** - because this is a programming course, but programming acumen is not tested as effectively on exams, this is to show at least a minimal level of Java programming competence and experience in addition to Java concept mastery. Also, past experience has

shown that students who do not put a solid effort into the course homework assignments do not do well on the course exams.

- If **all three** requirements above are **not** met, then your semester grade will be **either D+ or the letter grade computed according to the mapping given below, whichever is lower**.
 - (That is, if a student had an overall semester average of 74% but a Homeworks average of 55%, that student would receive a **D+** for his/her semester grade; if a student had a Homeworks average of 61% and an Exams average of 71%, but an overall semester average of 65%, then that student would receive a **D** for his/her semester grade. You are expected to ASK ME if this aspect of the grading policy is not clear to you.)
- Including the three requirements noted above, your semester grade will be computed according to the mapping given below:

| Overall Percentage (based on the given weights) | Exams Average | Homework Average | Letter Grade |
|---|---------------|------------------|--------------|
| ≥ 93 | ≥ 60 | ≥ 60 | A |
| ≥ 90 and < 93 | ≥ 60 | ≥ 60 | A- |
| ≥ 87 and < 90 | ≥ 60 | ≥ 60 | B+ |
| ≥ 83 and < 87 | ≥ 60 | ≥ 60 | B |
| ≥ 80 and < 83 | ≥ 60 | ≥ 60 | B- |
| ≥ 77 and < 80 | ≥ 60 | ≥ 60 | C+ |
| ≥ 73 and < 77 | ≥ 60 | ≥ 60 | C |
| ≥ 70 and < 73 | ≥ 60 | ≥ 60 | C- |
| ≥ 70 | < 60 | any | D+ |
| ≥ 70 | any | < 60 | D+ |
| ≥ 67 and < 70 | any | any | D+ |
| ≥ 60 and < 67 | any | any | D |
| < 60 | any | any | F |

Final Exam:

Again, the Final Exam for this course is scheduled for **Monday, December 14, 5:00 - 6:50 pm**, in **FR 107** (unless I announce otherwise). Note this time and date **BEFORE** making your end-of-semester travel plans.

(CS 235 is using the MW 1800 Final Exam time, while REC 210, which meets in FR 107 on Wednesdays, is using the MW 1700 Final Exam time.)

Additional Grading-Related Policies:

- Clicker questions will be given during most lectures and labs; graded lab exercises will be given during most lab sessions.
 - The **two lowest lab exercise grades** will be dropped from your grade.
 - Between the ample quantity of clicker questions and the dropped lab exercise grades, then, you can be absent several times from non-exam lecture or lab sessions without direct penalty, for whatever reason (although you are, of course, still responsible for the material covered on those days, and it is **your responsibility** to determine what that material is).
- Note: **NO** homework grades are dropped; **ALL** homework grades count toward your homework average. Every homework includes important practice of course fundamentals.

Course Expectations:

First: remember the general rule of thumb for college-level courses:

*To be successful in a course, you should plan to spend at least 3 hours outside of class for each 1 hour of college course credit. That implies an estimate of **at least 9 hours a week spent outside of class for this 3-credit course.***

However, you should be aware that:

- This is a programming course; you cannot learn Java programming without practicing doing it!
- Programming courses can be notorious time eaters. Occasionally, a problem with code will take large amounts of time to locate and fix.
 - Starting **early** enough so that you have time to ask me questions when you run into problems can help with this!
 - Why spend 4 hours struggling with a frustrating roadblock the night before the assignment is due, when you can spend 10 minutes composing an e-mail early in the week, work on other problems while waiting for the answer, and then get a reply that makes everything clearer as soon as you read it?
- You can only get comfortable with a new programming language by writing code using it, and it takes some much longer than others to master a language's basics. Practicing as much as possible helps.
 - (This means playing around with in-class examples, experimenting to see if something you are curious about really works like you think, doing further research on topics of interest, and so on.)
- Later concepts are built upon earlier concepts as the course progresses -- if you ask me as soon as you realize that some concept is not clear to you, that can help keep you from falling behind.
- Homework deadlines will **not** be extended because you waited too late to start or because you did not allocate enough time before the deadline to work on it; likewise, they will typically **not** be extended because of hardware or network failure. (Admittedly, campus failures might affect deadlines. But don't assume so until you have heard from me definitively.) You need to keep backups of your files at all times, and need to plan your schedule to be able to work on on-campus computers as necessary.
- If you have not completed an assignment by the deadline, **your best choice is to submit whatever you have managed to do by then**, as partial credit is your friend, to carefully study the posted example solution as soon as it is available, to ask me about anything there that is still unclear, and to get a good **early** start on the next homework.

A successful student in this class will:

- Attend every lecture and lab, clicker at the ready.
- Participate in class (discussing clicker answers with other students, asking questions, paying attention, taking notes, being an attentive partner when pair-programming in lab).
- Complete reading assignments in a timely fashion.
- Practice and "play around" with posted examples.
- Ask specific questions -- in class, in lab, in office-hours, and in e-mail.
- Read through each homework assignment as soon as it is posted.
- Start working on each homework assignment early in the week.
- E-mail the instructor with specific homework-related questions starting early in the week both to clarify what a question is asking for and when hitting roadblocks (being sure to include **BOTH** the code involved **AND** any error messages or descriptions of bizarre behavior).
- Always submit SOMETHING for a homework assignment, even if it is not complete. (I believe in partial credit on homeworks, believing that if you have at least started working on a problem, the posted example solution will be more helpful/understandable than if you have not.)
- Compare his/her homework solutions to posted example solutions when they become available.

- Study with others for exams, and practice explaining concepts to one another.
- Attempt every exam problem, and carefully study over exams when they are returned.
- Practice Java programming as much as possible.

Academic Honesty:

Students are responsible for knowing policy regarding academic honesty. For more information, visit:

<http://www2.humboldt.edu/studentrights/academic-honesty>

Observe that among the actions that are unacceptable are submitting another's program, code, or file as your own and failing to quote material (that includes algorithms, project, code, and comments, too!) taken from another person's work. (Note that copying another student's comments is also unacceptable.)

All course work is to be the work of each student, **individually**, **unless** it is **explicitly** stated otherwise at the beginning of that course work's description. Except for explicit exceptions, this is **not** a group or team programming course. When group work is explicitly permitted, the names of all students involved must be included on the work submitted. (For example, when you use **pair programming** in lab, the lab exercise will specify that, and then each pair-programmed file turned in will include both of the names of the students who worked on it as a pair.)

(**Important aside:** pair programming specifically means that two people sit at one computer, with one typing while the other says what to type. Both people are actively involved in the programming process. Pair-programming is **not** two people working at two computers, each doing different parts of the work individually. Pair-programming is also not one person doing all the work while the other does nothing or does something else. If pair-programming is ever explicitly permitted, then you are expected to actually pair-program any files you do not complete on your own.)

(If an assignment does explicitly specify that it is acceptable to pair program or work in groups, make sure that you don't get into the situation where you are merely watching someone else learn.)

For homework assignments (that are not explicitly specified as permitting pair-programming), students may discuss general approaches **as long as no one involved in the discussion is writing anything down or typing anything during such discussions**. Students may also help one another in determining causes of program bugs, or in determining the meaning of compiler error messages. However, in general, students may not work together to complete homework assignments, one student should not instruct another in how to write the code for a homework assignment, and **any type of copying or modifying of another person's computer files, OR of providing computer files to another, related to homework assignments is definitely over the line, and never justified**. This applies to copying of documentation and comments as well as to copying of program code.

Note that it is **your** responsibility to ensure that course assignment files are read-protected. If you are careless about this, and someone else copies your work, you will share the penalty. (In particular, be very careful about leaving work on shared network drives in campus labs, or in UNIX/Linux directories that are not read-protected.)

Learning takes hard work; when students turn in others' work as their own, it is a slap in the face to those seriously interested in learning. Not turning in an assignment results in no credit for that assignment, of course, but that is an honest grade. Work that violates the course honesty policy deserves a lower grade than that, and therefore the course policy is that work violating this policy will receive **negative** credit. A person providing a file for copying receives the same **negative** credit as the copier. Repeat offenses will be handled according to University policies.

Asking Questions/Getting Help:

- Sending questions by e-mail can be a very effective way to ask for help.
 - Include CS 235 along with the subject of your e-mail in the **Subject :** line of any class-related e-mail that you send me. This will help your e-mail be more recognizable as a class-related message, and will make it less likely that I will accidentally overlook it.
 - ALSO include a descriptive subject along with the CS 235 in that **Subject :** line -- this also increases the chances that I will notice and reply to your question more promptly. (In particular, do not just reply to a class e-mail message I have sent previously, and do not simply leave the **Subject :** line blank!)

- That said, if I have not replied to your e-mail within 24 hours, please re-send it, just in case I did overlook it somehow.
- You are expected to sign each e-mail you send me with your name -- sometimes the sender's identity is not obvious from one's e-mail address, especially for an off-campus e-mail address.
- Also, DON'T INCLUDE the word "password" in your e-mails to me -- pwd is a handy abbreviation to use instead -- because, due to phishing scams, HSU's spam filtering does not seem to like e-mails with that word in it! (Odd, but this was definitely the case in Spring 2010...)
- I try to check my e-mail (`st10@humboldt.edu` or `sharon.tuttle@humboldt.edu` or `smtuttle@humboldt.edu`) about once a day on weekdays, and about once over each weekend. This is another reason to start assignments early, so that you have time to receive a reply to any questions that might arise.
- You are encouraged to ask me questions in class, in office hours, and by e-mail. The most successful students are those who are not afraid to ask questions early and often (I will gently let you know if you are overdoing it), who do the assigned reading, who attend lecture and lab regularly, who start homeworks promptly after they are made available from the course web page, and who practice course concepts as much as possible.
 - It is better to ask a question sooner than later -- for example, it is better to send an e-mail with a specific question as soon as you think of it than it is to wait a day or two until the next class meeting or office hour. If you wait to ask such questions, you may not have time to complete the assignment.
 - It is perfectly reasonable if you send me a question and then end up finding out the answer yourself before you receive my answer; likewise, it is not a problem if you end up sending me several questions in separate e-mails (as you work on different parts of a homework while awaiting earlier answers).
- That said, I am expecting that you will ask **specific** questions – overly vague or broad questions are problematic.
 - (For example, an example of a specific question is, "When I try to run the method: (paste in the method), I receive the following error message: (paste in the error message) Can you point me in the right direction about what is wrong?" An example of an overly vague or broad question is: "Here's my Java class. Is it right?")

Additional Coursework-Related Policies:

- You should not expect to be able to finish course assignments during the lab sessions -- although you may occasionally get some lab time to work on course assignments, typical lab sessions will include a lab exercise that is to be completed in lab. Even when you finish the lab exercise early, it will still be the case that, like any college-level course, you should expect to put in a significant amount of time outside of scheduled class meetings (lectures and labs) doing the assigned reading, working on course assignments, and practicing concepts discussed.
- Each assignment must be submitted as is specified on its handout to be accepted for credit. This may vary for different assignments. Often, parts of assignments will be submitted using a special tool on nrs-projects.
- Each assignment will be clearly marked with one or more due dates (a single assignment could have multiple parts with multiple due dates).
 - **No assignments will be accepted late. If you wish to receive any credit for an assignment, then you must turn in whatever you have done, even if it is incomplete, by the deadline. Partial credit is usually preferable to no credit.** Note that "the computer/network/etc. going down" is no excuse --- if you leave an assignment for the last minute and there are technical problems, you still must turn in whatever you have by the deadline. As with any work done on computer, make frequent back-ups of your files!
 - You may submit **multiple versions** of assignment files before the deadline; I will grade the latest pre-deadline submission unless you inform me otherwise. This is to encourage you to turn assignment parts in early (since you will know that you can always turn in an improved version if further inspiration strikes). You also don't have to worry about forgetting to submit something that has already been submitted!
 - If for any reason you cannot submit course work using the submission tool on nrs-projects (or as specified by the assignment), e-mail me your homework files as attachments by the deadline, and then submit the files using the submission tool (or other specified means) as soon as you are able. The e-mailed files will establish that these

files were completed by the deadline.

- The nrs-projects tool that you will be using to submit some assignments results in a file that serves as your "receipt" for having submitted items. You are expected to retain these "receipt" files at least until a grade has been posted to the course Moodle site for that assignment. If there is a system glitch or other hardware/software/network problem, you may be asked to make me a copy of one or more receipt files; if you do not have them, then you will not receive credit for the files involved. These receipt files are for your protection!
- It is nearly impossible to write unambiguous specifications. If you have questions about "what she means", get them resolved very early in the development cycle by **asking**.
- There is more to computer code than simply whether it runs or not...
 - Part of your grade will be determined by how well your work meets the written requirements. Work that you turn in is expected to meet handout specifications precisely; when one eventually works within a team on large projects, following the specifications precisely is vital, and can mean the difference between a working product and one that just sits there.
 - Note that work may be graded on **style** as well as on whether it runs properly and whether it precisely meets the homework specifications and requirements. Discussions on style will be ongoing throughout the semester.
- Some course work may be graded simply based on whether it has been attempted (the instructor's decision is final as to whether this is the case) -- other course work may be graded for correctness, style, and whether it meets specifications. You will not know in advance which will be the case.

Incompletes:

Incompletes are rarely given and only in the case of a true emergency. They certainly are not appropriate for students who find they have fallen behind on assignments, missed a test, or taken on too much academic, work, or family responsibilities. For these situations, dropping the course would be appropriate (**if** that is still possible according to the University policies for dropping courses).

Additional Course Policies:

- You are expected to read this syllabus and be prepared to verify in a required Moodle activity that you have received it, have read it, and understand its contents.
- Exam dates are given in the course schedule below. Make-up exams are only possible by special prior arrangement or because of a valid medical excuse.
- You should monitor your e-mail for course-related messages. The University provides a means for you to specify your preferred e-mail address, so if you wish to receive e-mail into an account other than the one HSU provides, change your preferred e-mail address in both Account Center and Moodle accordingly. Course-related messages from me will include CS 235 in the Subject: line.
- You are expected to check the public course web page and the course Moodle site regularly -- course handouts, homework assignments, examples from lectures and labs, and possibly more will be posted to the public course web page, and grades will be posted to the course Moodle site. You are expected to monitor your posted grades and let me know about any discrepancies.
- When reading assignments are given, you are expected to prepare (read and study) assigned readings before class and to participate in class discussions. Projected examples will be utilized frequently during discussion. You should understand that there may be material in the reading that will not be discussed in lecture/lab, and material in the lectures/labs that may not be found in the reading. You are responsible for both.
- Regular attendance at lecture and lab sessions is expected. If you should happen to miss a lecture or a lab, then you are responsible for finding out what you missed. "I wasn't there that time" is never an acceptable excuse. Lecture and lab notes are not posted, although many of the projected examples will be made available on the public course web site. Clicker questions and graded lab exercises missed **cannot** be made up later (except for extraordinary circumstances).

- As previously mentioned, during lab sessions, there may be lab exercises due during that lab session. Once a lab's lab exercise is complete, the remaining lab time should be used to continue work on the current course homework assignment, to practice course concepts, and/or to ask questions about course-related topics.

Campus policies:

Students with Disabilities:

Persons who wish to request disability-related accommodations should contact the **Student Disability Resource Center** in the Learning Commons, Lower Library, **826-4678 (voice)** or **826-5392 (TDD)**. You can reach the Student Disability Resource Center's web site at:

<http://www.humboldt.edu/disability/>

Please note that some accommodations may take up to several weeks to arrange. If you are eligible for such accommodations, please contact me as soon as possible to discuss them.

Add/Drop Policy:

Students are responsible for knowing the University policy, procedures, and schedule for dropping or adding classes. You can find these on the web at:

<http://www.humboldt.edu/registrar/students/regulations/schedadjust.html>

You can find the University policies for repeating classes at:

<http://www.humboldt.edu/registrar/students/regulations/repeat.html>

Note the following highlights from the above CSU (and thus HSU) policies on withdrawing from and repeating courses:

- Students may withdraw from no more than 18 semester-units after the first four weeks of instruction; that is, students may withdraw from no more than 18 semester-units between census and the final 20% of instruction, and only then with a serious and compelling reason. (Note that: "Withdrawal from courses for reasons that are catastrophic, such as accident or serious illness do not count toward the 18-unit limit." [from the Registrar's web site])
- Students may repeat courses for grade forgiveness only if they earned grades lower than a C.
- Students may repeat up to 16 semester-units with grade forgiveness.
- Students may only repeat a course for grade forgiveness two times and each of these attempts counts toward the 16-unit maximum for repeats.
- Students may repeat up to an additional 12 semester-units with grades averaged.

Please note - it is the Registrar's Office that determines what constitutes a "serious and compelling reason".

The census date for Fall 2015 (before which you can drop **without** a W, and **without** it counting toward your 18 semester-units drop limit) is: **Monday, September 21st**.

The last date for Spring 2015 to drop with a W on your transcript, with a serious and compelling reason, and subject to the 18 semester-unit drop limit, is: **Monday, November 2nd**.

If you do drop the course, note that it is **your responsibility** to complete and submit the appropriate forms.

Attendance and disruptive behavior:

Students are responsible for knowing policy regarding attendance and disruptive behavior:

<http://www2.humboldt.edu/studentrights/attendance-behavior>

- **Late arrival to class:** Please attempt to come to class on time, with your headphones/earbuds/etc. put away and your cell phones/tablets/pads/gadgets/etc. turned off. If you must arrive late or leave early, please do so with the least possible distraction to other students. If your late/early habits become disruptive, you may be asked to leave the class permanently.

- **Class disruption:** University policy requires that instructors eliminate disruptions to the educational process. Distractions such as excess talking, ringing cell phones, working on assignments for other classes, inappropriate or distracting laptop/tablet/smartphone/gadget use, demonstrations of affection, packing of books early, loud music leaking from headphones, chronic late arrivals or early departures, excessive comings and goings or other behaviors that disrupt the class are not acceptable. Students indulging in such behaviors will first be warned before being required to leave the class permanently.

Emergency Evacuation

Please review the evacuation plan for the classroom (posted on the orange signs), and review the campus Emergency Preparedness web site at:

http://www.humboldt.edu/emergencymgmtprogram/campus_emergency_preparedness.php

...for information on campus Emergency Procedures. During an emergency, information regarding campus conditions can be found at **826-INFO** or:

<http://www.humboldt.edu/emergency>

Tentative Course Schedule: (subject to change!)

In a (hopefully-)typical course week:

- Monday's lecture will introduce the week's main topics (or continue those from the previous week)
- Wednesday's lab session will wrap up the in-class topics coverage and then proceed to an in-lab exercise
- A homework assignment related to that week's topics will be posted by sometime Wednesday night, due by 11:59 pm on the following Wednesday
- The reading to be done by the following Monday will also be announced by Wednesday's lab session (if not before)

Most of the main topics for each week are still being determined, as are each week's course readings.

Exams:

As noted in the Grading Breakdown section, there are two exams during the semester (in addition to the Final Exam). These exams change the hopefully-typical routine described above.

Student feedback has typically indicated that students find exam review sessions to be useful, and that having a weekend between such a review and the exam itself is likewise desirable. For this course, I think this can be achieved by reviewing for an exam during a Wednesday lab, and then holding the exam during the following Monday lecture.

To keep this from unduly reducing topics coverage, then:

- On the Monday before an exam, we will discuss "new" topics, that will NOT be included on that exam (since you will not have practiced those topics on a homework yet).
- On the Wednesday before an exam, we will review for that exam. The previous week's homework will still typically be due by 11:59 pm on that Wednesday, BUT a new homework will NOT be posted yet (to allow you more time for exam study).
- Then the exam is held (during the next Monday lecture).
- On the Wednesday after an exam, we will wrap up the in-class topics coverage from the "new" topics covered on the Monday before the exam, and then proceed to an in-lab exercise on those topics.
- And, a homework assignment related to those topics will be posted by sometime that Wednesday night (after the exam), due by 11:59 pm on the following Wednesday, getting us back into the more "normal" cycle.

Course dates:

Week 1: August 24, 26

- Topics: Intro to course, Intro to Java
- **Homework 1 out**

Week 2: August 31, September 2

- Topics: Java basics for C++ programmers
- **Homework 1 due, Homework 2 out**

Week 3: September 9

- **Monday, September 7 - Labor Day - HSU Holiday - NO CLASS**
- (as a result, the lab session for Wednesday, September 9 MAY be unusually-lecture-like!)
- Topics: Java objects for C++ programmers, continued
- **Homework 2 due, Homework 3 out**

Week 4: September 14, 16

- Topics: Intro to graphical user interfaces (GUIs) in Java -- includes intro to Java event handling and intro to Swing
- **Homework 3 due, Homework 4 out**

Week 5: September 21, 23

- **FYI: NOTE: Last day to drop a course without a W, without a serious and compelling reason, and without it counting toward your 18 semester-units drop limit is Monday, September 21.**
- Topics: Intro to GUIs in Java, continued
- **Homework 4 due, Homework 5 out**

Week 6: September 28, 30

- Topics: Intro to layout managers, review for Exam 1
- **Homework 5 due**

Week 7: October 5, 7

- **Monday, October 5: Exam 1**
- Topics: Intro to layout managers, continued
- **Homework 6 out**

Week 8: October 12, 14

- (Topics still to be finalized from here on out)
- **Homework 6 due, Homework 7 out**

Week 9: October 19, 21

- **FYI: NOTE: Last day to change a registered class' grade option to CREDIT/NO CREDIT is Monday, October 19.**
 - (that said, also note that courses applying to your CS degree requirements -- including major electives -- must NOT be taken as credit/no credit -- they must be graded with a letter grade)

- Homework 7 due, Homework 8 out

Week 10: October 26, 28

- Homework 8 due, Homework 9 out

Week 11: November 2, 4

- **FYI: NOTE: Last day to drop a course with a W, with a serious and compelling reason, and subject to your 18 semester-units drop limit is Monday, November 2.**
- Wednesday, November 4: review for Exam 2
- Homework 9 due

Week 12: November 9

- Monday, November 9: Exam 2
- Wednesday, November 11 - Veterans Day - HSU Holiday - NO CLASS
- Homework 10 out

Week 13: November 16, 18

- Homework 10 due, Homework 11 out

Fall Break - November 23 - 27

Week 14: November 30, December 2

- Homework 11 due, Homework 12 out

Week 15: December 7, 9

- Wednesday, December 9: review for Final Exam
- Homework 12 due

Final Exam:

MONDAY, December 14, 5:00 - 6:50 pm, in FR 107 (unless I announce otherwise)

(CS 235 is using the MW 1800 Final Exam time, while REC 210, which meets in FR 107 on Wednesdays, is using the MW 1700 Final Exam time.)