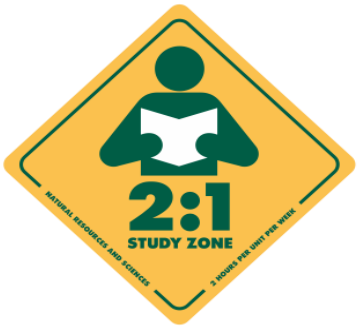

Humboldt State University
Course Syllabus for CS 235 - Section 10
Java Programming
CRN 43406 - Fall 2021

Lecture meets:	Mondays, 3:00 pm - 4:50 pm	1st 2 weeks: Zoom URL: see course Canvas site Starting Sept.13: BSS 408
Lab Section 11 meets:	Fridays, 3:00 pm - 4:50 pm	1st 2 weeks: Zoom URL: see course Canvas site Starting Sept. 10: BSS 317

Instructor:	Sharon Tuttle	
Instructor's e-mail:	st10@humboldt.edu or sharon.tuttle@humboldt.edu or smtuttle@humboldt.edu	(note: these are all ALIASES to the SAME mailbox)
Student hours: (Zoom URL: see course Canvas site)	Monday, Wednesday 12:00 pm - 1:00 pm Tuesday, Thursday 2:00 pm - 4:00 pm or by appointment	
Course public web site:	follow CS 235 link from: http://nrs-projects.humboldt.edu/~st10/ OR follow link from course Canvas 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++. **No** prior knowledge of Java is assumed.

Some of the Java-centric topics to be covered include an introduction to the Java Application Programming Interface (Java API), layout managers, graphical components, the Java event model, threads, exception handling, IO streams,

and (of course) a useful subset of Java classes from the Java API. Swing will be used to build Java applications, although some aspects of the AWT (abstract windowing toolkit) will also be covered. In addition, we will be discussing software testing in this course.

(The catalog description includes the term "applets", but those have fallen pretty heavily out of favor and so will not be covered in this Fall 2021 offering of CS 235.)

Course Prerequisite

CS 112, or instructor's consent.

No prior knowledge of Java is assumed. This course is an introduction to Java for those already familiar with object-oriented programming using C++.

(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 and comparisons with C++.)

Student Learning Outcomes

After successfully completing this course, students should be able to:

- Understand basic Java language syntax and semantics sufficiently to successfully read and write Java code.
- Use the Java Application Programming Interface (Java API) to learn about and to then successfully and appropriately use an already-written Java class and its methods.
- Design, implement, test, and debug Java classes and applications.
- Design, implement, test, and debug Java applications featuring graphical user interfaces that respond to user events.

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

- "Core Java - Volume I - Fundamentals", **11th Edition**, Cay Horstmann, Prentice Hall, 2019, ISBN 978-0-13-516630-7

  - Note: the **11th Edition** includes some coverage of new features added in Java Versions 9, 10, and 11 -- previous editions will not include such coverage.
  - When I checked in July 2021, the e-book of the 11th Edition WAS available on-line from the HSU Library's version of Safari!

    Instructions for reaching this: http://nrs-projects.humboldt.edu/~st10/f21cs235/235text-safari-instructions.pdf
- Turning Account License used with TurningPoint app (see "Clicker Questions" section below)
  - The TurningPoint app is free, but you do need to purchase a Turning Account License and register it from the CS 235 course Canvas site, or I will not be able to "see" your answers.
  - Note: I am told that the best price for the Turning Account License is available when you follow the "Turning Account Registration" link in Canvas -- this link is on the left-hand-side of the course Canvas site.
- Additional required readings will be made available either on-line, or via resources available through the HSU Library such as the ACM Digital Library and Safari TechBooks Online.

Course Software

IMPORTANT: Note that the Java you can download at https://www.java.com is **NOT SUITABLE** for this course -- that is the Java Runtime Environment (JRE), which is for end-users who want to run already-compiled Java programs, rather than for those who would like to program in Java. **Programmers need a version of Java Development Kit (JDK).** (Java is rather cursed with a myriad of acronyms!)

The question of precisely which version(s) of Java we will be using this semester is unexpectedly complicated...!

We will be using an **OpenJDK** (Open Java Development Kit) version of Java -- a free and open-source implementation of the Java Platform, Standard Edition (Java SE). [https://en.wikipedia.org/wiki/OpenJDK].

The currently-available (as of July 2021) OpenJDK version is **OpenJDK 16.0.2**, available for download from:

https://jdk.java.net/16/

When your Java program does not involve a graphical user interface (GUI), you can also use the version of OpenJDK on the CS50 Integrated Development Environment (IDE). As of July 2021, this is running **OpenJDK 15.0.1**. You can reach this free browser-based IDE at:

https://ide.cs50.io/

- (note that you initially log in to the CS50 IDE using a free GitHub account -- see instructions at:

  http://nrs-projects.humboldt.edu/~st10/f21cs235/235cs50-ide-setup.pdf

)

Our course text says that it "...has been updated to reflect the features of Java Standard Edition (SE) 9, 10, and 11." If I am understanding correctly, then, you should be OK for CS 235 as long as you are running an **OpenJDK that is version 11 or later**.

Can you use different Integrated Development Environment (IDE) software for this course? **IF** that IDE software can use a recent-enough version of OpenJDK, I *think* so. **I will announce in class if it turns out that some course topic** (such as JUnit for unit testing) **might require use of a particular IDE.**

Note that some use of a Linux shell will be required for this course, along with a recent-enough version of the Java OpenJDK as noted above.

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 one of your approved CS major electives.

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:		30%	Note: **NO** homework grades are dropped
Lab exercises:		15%	Note: Lowest two lab exercise grades are dropped
Clicker questions:		15%	**Sum** of points earned from answering clicker questions, up to a **maximum** of 120 points
Exams:	**Exam 1:**	10%	**Friday, October 15**
	Exam 2:	10%	**Friday, November 12**
	Final Exam:	20%	**Wednesday, December 15, 3:00 - 4:50 pm**

Grade Requirements

- To earn a grade of **C- or better** in this course, the following three requirements must **ALL** be met:
  1. your overall semester average must **equal or exceed 70%** - this is to show a reasonable level of overall mastery of the course material.
  2. 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.
  3. 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 minimum level of Java programming experience in addition to Java concept mastery. Also, part experience has shown that, in general, students who do not put a solid effort into course homework assignments do not do well on course exams.
- If **any** of the **three** requirements above is **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 their 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 their semester grade. You are expected to ASK ME if this aspect of the grading policy is not clear to you.)
- So, your semester grade will be computed as shown in this table:

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+

Overall Percentage (based on the given weights)	Exams Average	Homework Average	Letter Grade
>= 67 and < 70	any	any	D+
>= 60 and < 67	any	any	D
< 60	any	any	F

More Coursework-related Policies

- It is nearly impossible to write unambiguous specifications. If you have questions about what is being asked for -- whether on a homework problem, in a lab exercise, on an exam question, or even for a clicker question -- you are expected to **ask** me.
  - Being able to ask such questions is a necessary and important real-world skill in computer science!
- There is more to a computer command, expression, statement, function, file, or program than simply whether it "runs".
  - Part of your grade may be determined by how well your work **meets the stated requirements**.

    Your work is expected to meet stated requirements precisely. When working as part of a team on larger software projects, following specifications precisely is vital, and can mean the difference between a working product and one that just sits there.
  - Work may be graded on **style** as well -- following style and coding standards likewise helps to result in programs that are more readable, understandable, and maintainable over time. Discussions on style will be ongoing throughout the semester.

Homework Assignments

- Note that **no homework assignment grades are dropped**; *every* homework assignment grade is included in determining the homework portion of your semester grade. Every homework includes important practice of Java features.
- Homework problems are to be completed individually (although *discussing* homework problems with other students without copying their comments or code is fine!).
- Each homework assignment must be submitted as specified on its handout to be accepted for credit. This may vary for different homework assignments.
- Each homework assignment will be clearly marked with one or more due dates/deadlines (a single homework assignment could have multiple parts with multiple due dates/deadlines).
  - To best benefit from this class, it is important to **practice Java programming regularly** and to **attempt homework problems before the homework deadlines**.
  - However, you can still submit homework problems (and requested revisions, if any) up until example solutions are posted, before each Exam.
  - Once a homework's example solutions are posted, no more submissions or revisions will be accepted for that homework (*unless* you have discussed your unusual situation with me and we have set up a different arrangement).
- You may submit **multiple versions** of homework files and problems; I will grade the **most recent** submission unless you inform me otherwise. (Homework short-answer questions answered on Canvas are handled differently, though -- see the section below.)

  - One reason for encouraging multiple submissions is to encourage you to **turn work in early and regularly**, even perhaps while it is still in-progress, since you can always turn in an improved version later, or if further inspiration strikes, etc.
  - Another benefit of early and regular submissions as you work through homework problems: you don't have to worry about forgetting to submit something that has already been submitted!

Homework Short-Answer Questions

- Most homework assignments will start with one or more problems that are short-answer questions answered on Canvas -- these are meant to give you a chance to see if you are on the right track on new syntax, new terms, or just important concepts.
- These short-answer questions are automatically graded -- after you have attempted all of a problem's questions, you will not be shown the correct answers, but you will be shown if your answers were correct or not, often including some additional explanation.
- **You can attempt these short-answer questions as many times as you would like -- your score for these will be the highest score from all of your attempts.**
  - Because students in the past mentioned that these were useful for exam review, these will be left available/open through the Final Exam.
  - HOWEVER, you will receive the **MOST** benefit from these if you start attempting them well **before** that homework's deadline, as a warm-up to the programming problems.

How I *hope* to handle grading homework assignments in Fall 2021:

- At a homework's deadline, I hope to start grading submitted homework files. (As noted above, the homework short-answer questions graded on Canvas are automatically graded.)
  - I *hope* to grade homework problems submitted by that homework assignment's deadline first -- you may have to wait longer for feedback on homework problems submitted after its deadline.
  - I am going to see if Canvas will allow me to give you grading comments while grading is still "in progress" (before the grading of a homework assignment is necessarily finished).

Lab Exercises

- Graded lab exercises will be given during most lab sessions.
- If you miss a lab session, typically its graded lab exercise cannot be made up later (except for extenuating circumstances - please let me know!). However, the **two lowest lab exercise grades** will be dropped from the lab exercise portion of your semester grade.
- You will typically be using **pair programming** for lab exercises -- in pair programming, two programmers work on and view the same file at the same time, one typing and the other saying what to type, but also discussing along the way.
  - **Both** are **actively** involved in the programming process **together**.
  - This software engineering practice can result in programs with fewer errors, amongst other potential benefits.
  - While learning to program, this practice can also give you more chances to discuss course concepts with other students, (along with the practical benefit of reducing the total number of questions the instructor has to try to answer during lab sessions, hopefully also reducing your wait time for those answers).
- Note: if, for example, there is an odd number of students at a particular lab, or there are technical difficulties, we'll also sometimes have trios -- in that case, **all three** are still working on and viewing the same file at the same time, one programmer types, and the other two alternate saying what to type, and of course all are still also discussing along the way.

- It is **not acceptable** to simply sit back during a lab exercise and have your partner(s) do all the typing and saying what to type and discussing -- you are expected to **actively participate** in your pair/trio.
- Please let me know of any issues that come up related to pair programming.
- Once you have completed a lab session's lab exercise, made sure that all of your pair's/trio's members have a copy of its files, and submitted your copy of those files, it is acceptable to leave the lab session.
  - After completing and submitting the lab exercise, it is also fine to use the remaining lab time to work on the current course homework assignment, to practice course concepts, and/or to ask questions about course-related topics.
  - However, note that questions from those still working on the lab exercise will be prioritized!

Clicker Questions

We will be using the Turning Technologies student response software in class. There is significant literature indicating that using such so-called "clicker questions" may increase student engagement and success in learning.

Students purchase a Turning Account license/subscription and register it from the CS 235 course Canvas site, and they use this license with the TurningPoint application on a mobile device or from a web browser. You then will answer questions using this during **every** class meeting (lectures **AND** labs). (Part of the idea here is to stress that **every** class meeting is important, and that participating during **every** class meeting is important.)

Follow the **"Turning Account Registration"** link on the course Canvas site for registering so that your answers receive credit. (You can also purchase the Turning Account license via this link, and I am told they offer the best price for this.)

This software will be used for in-class questions, which might be asked at any time within class meetings. These will usually be given in a **think-pair-share** fashion, in which you answer a question first on your own, and then discuss your answer with other students, 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.

Note that a large part of the benefit of this is from these discussions with other class members -- research suggests both that putting concepts into your own words helps you to learn them better and that the other class member's explanations may also help you to learn them better.

Typically, you will receive:

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

Thus you will be rewarded for regular attendance and participation.

I hope to run tests of the system during the first week's class meetings, and to begin asking questions that "count" during the second week's class meetings. So, you need to purchase and register your license as soon as possible. If there is an issue with this, please let me know as soon as possible.

Finally, **NOTE** that use of another CS 235 student's account, or having someone else use your TurningPoint account in a CS 235 class session, or otherwise having anyone but yourself answering a clicker question on your behalf -- 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.

Can clicker questions be "made up"? [revised 2021-09-29, since allowing Zoom attendance after the 1st 2 weeks due to the Delta variant surge]

Given that:

- due to the Delta variant surge, you may choose to attend CS 235 class sessions either face-to-face or via Zoom (that is, given that we are attempting a hyflex approach to CS 235, after all), and
- so, even after the first two weeks, class sessions will typically be recorded using Zoom, with links posted to the course Canvas site,

...it seems reasonable to shift CS 235 to use the same policy as I have been trying in my Zoom-only courses with regard to making up missed clicker questions.

So: if you miss a class session, or something happens that keeps you from answering the clicker questions during that session, I want to give you a chance to still answer them, but somehow also give you an opportunity to interact with them more actively than just hearing their answers (since you won't have the opportunity to discuss them with other class members). The goal is to help you still benefit from the clicker questions in this situation.

Here, then, is the **experiment** I am continuing in Fall 2021 for providing a bit of this:

- **Watch the recording of the Zoom session** you missed or during which you were unable to answer clicker questions. Start an e-mail to me (that you will fill in as described below).
  - **When a clicker question is asked** during the recording, **pause** the recording and try to answer it yourself before going on (and fast-forwarding through the breakout room part, etc.)
  - **Add to your e-mail-in-progress** your **answer** to that clicker question, **ALONG with** at least **ONE** of the following:
    - a **question** you have about that clicker question and/or any of its answers, OR
    - an **explanation** of why one of the **incorrect** answers is **wrong**, OR
    - an **explanation** of why the **correct** answer is **right**
- After you are done watching the Zoom recording, **send me the resulting e-mail with your answers plus questions/explanations**, using the **Subject:** line:

  `Subject: CS 235 Clicker Answers for <date>`
- And of course, please be sure to ask me if you have any questions about material covered in that Zoom recording.

Exams

For Fall 2021, CS 235 will have two exams during the semester and a Final Exam, at the dates given below.

Make-up exams are only possible by special prior arrangement or because of extenuating circumstances. You are expected to **contact me as soon as reasonably possible** in such circumstances.

There will be a review session before each of these exams as noted in the Tentative Course Schedule section.

Exam 1:

Exam 1 will be given during class on **Friday, October 15**.

Exam 2:

Exam 2 will be given during class on **Friday, November 12**.

Final Exam:

The Final Exam will be given from **3:00 - 4:50 pm** on **Wednesday, December 15** in **BSS 317** (unless I announce otherwise).

- (This is the required date and time specified in the campus Final Exam schedule for a course that meets at 3:00 pm on MWF-- a class with one lecture a week at 3:00 pm on Mondays is also expected to use this Final Exam time.)

NOTE: You can also find the schedule for ALL of YOUR Final Exams in your Student Center! See:
https://studentcenterhelp.humboldt.edu/final-exam-schedule

But, just in case: HSU Fall 2021 Final Exams schedule matrix is also posted at:
https://registrar.humboldt.edu/sites/default/files/final-exam-matrix-fall-2021.pdf

Expectations of the Student

- Read this syllabus, and be prepared to verify in a required Canvas activity that you have received it, have read it, and understand its contents.
- **Attend all class sessions, and participate!** Participating includes:
  - paying attention
  - discussing clicker question answers and class concepts with other students
  - being an attentive partner when pair-programming in lab
  - asking questions
- There is a general rule-of-thumb for college-level courses:

  To be successful in a course, you should plan to spend at least 2 hours outside of class for each 1 hour of college course credit. That implies an estimate of **at least 6 hours a week spent outside of class for this 3-credit course.**
  - You can only learn Java by **practicing** it. Practicing Java programming as much as possible helps!
  - This can include **typing in and playing around with in-class examples**, experimenting to see if something you are curious about really works like you think, and so on.
  - Think of a musical instrument -- you have to practice to master playing a guitar, violin, trumpet, drums, etc. You can't master it by just reading about the instrument. Think, also, of sports skills such as pitching, archery, etc. -- again, repetition and practice is required to hone such skills.
- Complete reading assignments in a timely fashion. Ask me if you have any questions about them.
- Check the CS 235 public course website and Canvas course site **frequently** for homework and other assignments, postings of course handouts and in-class examples, announcements, and updates.
- Check your HSU e-mail **daily**.
  - All e-mails that I send for this course will include `CS 235` in their `Subject:` line.
  - Likewise, include `CS 235` along with a description of your e-mail in the `Subject:` line of all class-related e-mails that you send to me.
- Start working on homework assignments as soon as they are posted, submitting frequently. This gives you time to ask questions if you run into problems.
  - Why spend 4 hours struggling with a frustrating roadblock the night before the homework 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?
- Ask questions when you are having difficulty understanding a class concept or not making progress on a homework problem.
  - Ask questions early and often (I will gently let you know if you are overdoing it.)
  - Writing programs can be a notorious time-eater. Sometimes a very small issue can take a long time to locate and fix, especially if you do not ask for help.
  - Later concepts are built upon earlier concepts as the course progresses -- if you ask as soon as you realize that

some concept is not clear to you, that can help keep you from falling behind.

- Keep backups of your CS 235 files; if I cannot open one of your submitted lab exercise or homework files, I may need you to re-submit them or to e-mail them to me.
- If you have not completed a lab exercise or homework problem by its deadline, submit whatever you have done up to that point, even if it is not complete.
  - (Even when you can still submit revisions later, submitting what you have by the deadline shows that you have started, and *might* allow me to give you feedback based on what you have done so far.)
  - Also, I believe in partial credit on homeworks, believing that if you have at least started working on a problem, any eventual posted example solution will be more helpful/understandable than if you have not.
- Take the opportunity to learn how to write your own thoughts; don't plagiarize. Be sure to give credit where credit is due and cite your sources.
- If example solutions for selected homework problems are posted, read those over and compare them to how you approached those problems. Be sure to ask me if you have any questions as a result!
- When grades are posted to the course Canvas website, check them and let me know about any discrepancies or issues.

Class Culture*

We will decide on the final expectations together, but some of the guiding principles will involve:

- Respect for each other (what does that mean to you?)
- Come to class sober
- Keep cell phones and other distractions put away
- Be in the meeting or classroom before class starts, so that you're ready when it starts
- Do not leave in the middle of class unless there is a real need (e.g., family emergency, you are too ill to stay in class)
- Stay until class is over
- Be a regular and willing participant
- Follow campus COVID-19 health/safety guidelines (see the "HSU Campus Ready" site, https://campusready.humboldt.edu/ , for more information)

*These principles are written with face-to-face instruction in mind, but also apply to remote learning via Zoom. Contact the instructor if you need special accommodation or exception from these rules.

Expectations of the Instructor

- I will prepare and review course materials to be as current and accurate as possible.
- I will be available to answer questions or issues that may arise for you during this course. **Expect a 24-hour turnaround time for response to e-mails on weekdays and 48 hours on weekends.**
- I will try to the best of my ability to prepare you for the assignments and other assessments in this course.
- I will utilize fair and honest evaluation techniques for each assignment required for this course.
- I will do my best to address the needs of a diverse range of learning styles in this course.
- I will only share your student information per FERPA (federal privacy) guidelines.
- I will follow campus COVID-19 health/safety guidelines.

Other Course Policies

Inclusivity

Students in this class are encouraged to speak up and participate in-class. Each of us must show respect for each other because our class represents a diversity of beliefs, backgrounds, and experiences. I believe that this is what will enrich all of our experiences together. I recognize that our individual differences can deepen our understanding of one another and the world around us, rather than divide us. In this class, people of all ethnicities, genders and gender identities, religions, ages, sexual orientations, disabilities, socioeconomic backgrounds, regions, and nationalities are strongly encouraged to share their rich array of perspectives and experiences.

If you feel your differences may in some way isolate you from our classroom community or if you have a specific need, please speak with me early in the semester so that we can work together to help you become an active and engaged member of our class and community. *(Adapted from HSU Canvas Accessible Syllabus Template, which was in turn adapted from CSU Chico and Winona State University)*

Thus, spoken language and body language should emanate respect for everyone in our classroom community. This includes coming to class on time and being prepared to listen and share. (*Adapted from Jayne McGuire's syllabi language*)

CS 235 E-mail Policies

- **NOTE:** please do **NOT** use Canvas messages to contact me or ask me a question -- send me **actual e-mail messages** instead. Handling Canvas messages is time-consuming and error-prone on my end.
- HSU students are responsible for checking their HSU e-mail account for official communications. You are expected to check for course-related messages as well.
  - While students may elect to redirect messages sent to their official HSU email address to another address, those who redirect their email to another address do so at their own risk.
  - HSU Email Policy: https://policy.humboldt.edu/p16-01-email-policy
- All e-mails that I send for this course will include `CS 235` in their `Subject:` line.
- Likewise, include `CS 235` along with a description of your e-mail in the `Subject:` line of all class-related e-mails that you send to 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!)
- Ask **specific** questions via e-mail -- for less-specific or broader questions, come to student hours or make an appointment to meet with me. Overly-vague or broad questions are problematic to answer by e-mail.
  - For example, an example of a **specific** question is:

    "When I try to run my application myApplic [attach your `myApplic.java` file with class `MyApplic`'s Java source code], I receive the following error message: [paste in the first 4-5 lines of that 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 program: [just pasting in its code]. Is it right?"
- When e-mailing a question about a program,

  - attach a copy of your program file(s)

    and ALSO

  - paste in the first 4-5 lines of the **error messages** you are getting

    and/or descriptions of **bizarre behavior** you are seeing.

- It is perfectly reasonable if you e-mail me a specific question and then happen to find out the answer yourself before you receive my answer. (Letting me know you've found the answer is fine, too!)
- Likewise, it is not a problem if you happen to send me several specific questions in separate e-mails (for example, as you work on different homework problems while awaiting earlier answers). I can answer shorter e-mails more quickly than longer e-mails.
- Expect a **24-hour turnaround time** for response to e-mails on weekdays and **48 hours** on weekends.
  - So, in general, if I have not replied to your e-mail within 24 hours, please **re-send** it, just in case I have overlooked it or some glitch occurred.
  - (And if there seems to be a chance that your message is getting chomped by a spam filter -- rare, but not unprecedented! -- leave me a message at 707-826-3381 with the `Subject:` line of the email you are trying to send and the e-mail address you are using, and I will see if I have indeed received it!)
- 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.
- Please take a few minutes to ensure that your message reflects a professional tone. I know I have sent an email or two in the heat of the moment that I soon regretted. Take your time and communicate professionally. (*Adapted from Jayne McGuire's syllabi language*)

Course Absences

Between the ample quantity of clicker questions asked during the semester, and the two dropped lab exercise grades, you can be absent several times from non-exam lecture or lab sessions without significant direct penalty, for whatever reason. However, it is **your responsibility** to find out what was announced and covered on those days; "I wasn't there that time" is not an acceptable excuse.

Lecture and lab notes are not posted, although examples projected in class, and any notes that happen to also be projected, will be made available on the public course web site. You should carefully read over these, and be sure to ask me if you have any questions about them.

Please let me know if class or life issues are making it difficult for you to attend class meetings or to keep up with course material and coursework, so we can make arrangements to help you work through those. It helps if you let me know **sooner** rather than later about such issues!

Academic Honesty

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

https://www2.humboldt.edu/studentrights/academic-honesty

Plagiarism is a serious offense. Copying of another person's work and submitting it as your own for individual assignments, or providing your work to others for them to copy and submit as their own for such assignments, is not acceptable.

Notice that this also means that it is NOT okay to copy or post homework answers or code from or to an online discussion or from or to sites such as Chegg.

Learning takes hard work; when students turn in others' work as their own, or provide it to others to copy, it is a slap in the face to those seriously interested in learning who are putting in that effort. Not turning in an assignment results in no credit for that assignment, 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 may receive **negative** credit. Likewise,

a person **providing** a file for copying would receive the same **negative** credit as the copier. Repeat offenses will be handled according to University policies, and may result in appropriate penalties up to and including a failing grade in the course.

When you pair-program in lab, both/all of your names are included in the work files that result. This should mean that you all *participated* in pair-programming for that assignment.

Did you find an interesting inspiration from a Google search or a book for your algorithm or for a part of your code? **Attribute** it -- include a **comment giving its source**! It is also fine and encouraged to make use of functions from posted in-class examples, also -- but it is professional to comment their source, also.

Note that it is **your** responsibility to ensure that your homework 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, or in UNIX/Linux directories that are not read-protected.)

Is is OK to help each other?

On exams, **no**. (That said, studying together for each exam, before taking it, is an excellent idea and encouraged!)

For homework assignments, discussing approaches to homework problems is fine -- a good rule-of-thumb is that you are discussing approaches but not writing down or copying how to complete a particular problem.

Students may also help one another in determining causes of homework problem bugs, or in determining the meaning of error messages.

However -- again -- any copying or modifying of someone else's answers, source code, or files, OR of providing answers, source code, or files to another, related to homework assignments and exams is definitely over the line, and never justified.

More on Asking Questions/Getting Help

- You are encouraged to ask questions in class, in student 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).
- Especially with regard to homework assignments, it is usually better to ask a question **sooner** than later.
  - For example, it is better to send an e-mail with a specific question you have about a problem as soon as you think of it, rather than wait a day or two until the next class meeting or student hour or CS tutoring session.
  - If you wait to ask such questions, you might not have time to complete the assignment.

Incompletes

Incompletes are rarely given and only in the case of a true emergency. They 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).

If you are facing extenuating or emergency circumstances at any time during the semester, please consider contacting the HSU **Campus Assistance, Response, and Engagement (CARE) Services** office:

https://deanofstudents.humboldt.edu/CARE

Campus policies

The following URL leads to useful links regarding HSU policies, procedures, and resources:

https://academicprograms.humboldt.edu/content/syllabus-addendum

All of the policies linked from the above are applicable to this class, and you are expected to be familiar with these policies.

The following are just a FEW highlights from this site, along with a few additional campus-policy-related notes:

Students with Disabilities

Persons who wish to request disability-related accommodations should contact the **Student Disability Resource Center**, **826-4678 (voice)** or **sdrc@humboldt.edu**. Disability accommodations must be pre-approved by the Student Disability Resource Center.

You can reach the Student Disability Resource Center's web site at:

https://disability.humboldt.edu/

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.

Dropping or Adding a Class

- Students are responsible for knowing the University policy, procedures, and schedule for dropping or adding classes.
  - **Note that the drop deadline as well as the deadline for changing a course's grade mode to CR/NC have changed from those in effect during the height of the COVID-19 pandemic.**
  - You can find these deadlines for Fall 2021 in the "Activities and Deadlines" calendar for Fall 2021, available at:

    https://registrar.humboldt.edu/sites/default/files/activites-deadlines-fall2021.pdf

    (There are MANY important deadlines in this calendar -- it is well-worth reading through!)
- **Note that the Add/Drop deadline for Fall 2021 is 11:59 pm on MONDAY, SEPTEMBER 6th.**
  - This is the deadline to add or drop courses through the Student Center.
  - After September 6th, dropping a course requires a "serious and compelling reason", and it is the **Registrar's Office** that determines what constitutes a "serious and compelling reason".
  - Note that it is the student's responsibility to properly drop a course.
- You can also find more information about dropping or adding a class at:

  https://registrar.humboldt.edu/forms - and click on **Add/Drop Date** on the right-hand side (OR toward the BOTTOM if viewing this on a phone or within a narrow browser window!)
- You can find the University policies for repeating classes at:

  https://registrar.humboldt.edu/forms#policies - and click on **Repeating Courses** on the right-hand side (OR toward the BOTTOM if viewing this on a phone or within a narrow browser window!)

Note about Course Grade Modes

During the height of the COVID-19 pandemic, CS majors and CS minors were permitted to take courses that applied to CS major/minor requirements with a grade mode of CR/NC. **This is no longer the case in Fall 2021.**

Starting in Summer 2021, HSU is back to the policy that courses that apply to CS major/minor requirements must be taken with a grade mode of letter grade. This includes courses taken for the CS major's CS electives' requirement!

If you are taking this course as a **free** elective, however (and **NOT** applying it to a CS major *as a CS major elective* or to a CS minor), then note that the limit of **at most one optional CR/NC course per term** is back in effect, and that the deadline to change grade modes has also moved back to its pre-pandemic timing -- for Fall 2021, that date is **Monday, October 18th.**

For more information on optional CR/NC grade mode, see:

https://registrar.humboldt.edu/node/407 - and click on **Credit Limitations** on the right-hand side (OR toward the BOTTOM if viewing this on a phone or within a narrow browser window!), and within that scroll down to the **Credit/No Credit** section within.

Attendance and disruptive behavior:

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

https://www2.humboldt.edu/studentrights/attendance-behavior

- **Class disruption:** University policy requires that instructors eliminate disruptions to the educational process. Distractions such as excess talking or behaviors that disrupt the class are not acceptable.
  - Students indulging in such behaviors will first be warned before any additional measures are taken (although a warning is not required in the case of abusive behavior).

TENTATIVE Course Schedule: (subject to change with fair notice)

Note also that **additional readings may be added** to those given below.

Week 1: August 23, 27

- **Topics:** Intro to course, intro to Java; starting intro to Java basics for C++ programmers
- **Reading:** from "Core Java - Volume I - Fundamentals", 11th Edition:
  - skim Chapters 1, 2, 3
- **Homework 1 out**

Week 2: August 30, September 3

- **Topics:** Continuing intro to Java basics for C++ programmers; Java arrays; Java `ArrayList` class; possibly a simple example of Java file input/output
- **Reading:** from "Core Java - Volume I - Fundamentals", 11th Edition:
  - skim Chapters 1, 2, 3
  - Chapter 3 - Section 3.10 - Arrays
  - Chapter 5 - Section 5.3 - Generic Array Lists
  - Chapter 9 - Subsection 9.3.2 - Array Lists
  - Chapter 3 - Subsection 3.7.3 - File Input and Output
- **Homework 1 due - 11:59 pm Friday, September 3**
- **Homework 2 out**

Week 3: September 10

- **FYI: NOTE: Last day to drop a course through your Student Center (without a W and without a serious and compelling reason) is Monday, September 6.**
- **Monday, September 6 - Labor Day - HSU Holiday - NO CLASS**
  - (as a result, the lab session for Friday, September 10 MAY be unusually-lecture-like -- if that happens, the lab exercise MAY be answering clicker questions during that lab session)
- **Topics:** Starting intro to Java objects for C++ programmers
- **Reading:** from "Core Java - Volume I - Fundamentals", 11th Edition:
  - Chapter 4 - Objects and Classes
  - Chapter 5 - Inheritance
- **Homework 2 due - 11:59 pm Friday, September 10**

Week 4: September 13, 17

- **Topics:** Continuing intro to Java objects for C++ programmers; intro to interfaces; intro to lambda expressions
- **Reading:** from "Core Java - Volume I - Fundamentals", 11th Edition:
  - Chapter 4 - Objects and Classes
  - Chapter 5 - Inheritance
  - Chapter 6 - Section 6.1 - Interfaces
  - Chapter 6 - Section 6.2 - Lambda Expressions
- **Homework 3 out**

Week 5: September 20, 24

- **Topics:** Starting intro to graphical user interfaces (GUIs) in Java; intro to Java event handling; intro to Java Swing
- **Reading:** from "Core Java - Volume I - Fundamentals", 11th Edition:
  - Chapter 10 - Graphical User Interface Programming
  - Chapter 11 - Subsection 11.3.2 - Labels and Labeling Components
- **Homework 3 due - 11:59 pm Friday, September 24**
- **Homework 4 out**

Week 6: September 27, October 1

- **Topics:** Continuing intro to graphical user interfaces (GUIs) in Java; intro to Java event handling; intro to Java Swing
- **Reading:** from "Core Java - Volume I - Fundamentals", 11th Edition:
  - Chapter 10 - Graphical User Interface Programming
  - Chapter 11 - Subsection 11.3.2 - Labels and Labeling Components
  - Chapter 11 - Subsection 11.4.3 - Borders
  - Chapter 7 - Sections 7.1, 7.2, 7.3 - related to exceptions
- **Homework 4 due - 11:59 pm Friday, October 1**
- **Homework 5 out**

Week 7: October 4

- **Topics:** Intro to layout managers
- **Reading:** from "Core Java - Volume I - Fundamentals", 11th Edition:
  - Chapter 11 - Section 11.2 - Introduction to Layout Management
- **October 8 - no lab**, because instructor will be traveling to a conference (CCSC-NW 2021)
- **Homework 5 due - 11:59 pm Friday, October 8**

Week 8: October 11, 15

- **Monday, October 11 - REVIEW for Exam 1** (NOTE that this will include several review clicker questions.)
- **Friday, October 15: Exam 1**
- **Homework 6 out**

Week 9: October 18, 22

- **FYI: NOTE: Last day to change a registered class' grade option to CREDIT/NO CREDIT is Monday, October 18.**
  - (now back to limit of at most **one** optional CR/NC course permitted per term)
  - (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**)
- **Topics:** Starting intro to threads
- **Reading:** from "Core Java - Volume I - Fundamentals", 11th Edition:
  - Chapter 12 - Sections 12.1, 12.2, 12.3 - related to basics of threads
  - Chapter 10 - Section 10.3 - Displaying Information in a Component (the parts related to the `Graphics` and `ImageIcon` classes)
- **Homework 6 due - 11:59 pm Friday, October 22**
- **Homework 7 out**

Week 10: October 25, 29

- **Topics:** Continuing intro to threads
- **Reading:** from "Core Java - Volume I - Fundamentals", 11th Edition:
  - Chapter 12 - Sections 12.1, 12.2, 12.3 - related to basics of threads
  - Chapter 10 - Section 10.3 - Displaying Information in a Component (the parts related to the `Graphics` and `ImageIcon` classes)
- **Homework 7 due - 11:59 pm Friday, October 29**
- **Homework 8 out**

Week 11: November 1, 5

- **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 1.**
- **Topics:** Continuing intro to threads
- **Reading:** from "Core Java - Volume I - Fundamentals", 11th Edition:
  - Chapter 12 - Sections 12.1, 12.2, 12.3 - related to basics of threads
- **Homework 8 due - 11:59 pm Friday, November 5**

Week 12: November 8, 12

- **Monday, November 8 - REVIEW for Exam 2** (NOTE that this will include several review clicker questions.)
- **Friday, November 12: Exam 2**
- **Homework 9 out**

Week 13: November 15, 19

- **Topics:** intro to Java Database Connectivity (JDBC)
- **Reading:** to be determined (JDBC is not covered in the course text, "Core Java - Volume I - Fundamentals", 11th Edition)

- **Homework 9 due - 11:59 pm Friday, November 19**
- **Homework 10 out**

FALL BREAK - November 22-26

Week 14: November 29, December 3

- **Topics:** Java packages and JAR files
- **Reading:** from "Core Java - Volume I - Fundamentals", 11th Edition:
  - Chapter 4 - Section 4.7 - Packages
  - Chapter 4 - Section 4.8 - JAR Files
- **Homework 10 due - 11:59 pm Friday, December 3**
- **Homework 11 out**

Week 15: December 6, 10

- **Topics:** To be announced
- **Reading:** to be announced
- **Friday, December 10 - REVIEW for Final Exam** (NOTE that this will include several review clicker questions.)
- **Homework 11 due - 11:59 pm Friday, December 10**

Final Exam:

WEDNESDAY, December 15, 3:00 - 4:50 pm, in BSS 317 (unless I announce otherwise)