# CS 212 Course Syllabus

|  |  |
| --- | --- |
| **Prefix:** | CS 211 |
| **Title:** | Algorithms |
| **# Credits:** | 4 |
| **Semester/Year:** | Fall 2017 |
| **Pre-requisites:** | CS 211, STAT 108 (C), and MATH 105 or MATH 109 |
| **Meeting Times:** | **Lecture:**  Tu, Th 3:00 PM - 4:50 PM |
| **Course website:** | Canvas (canvas.humboldt.edu)  Github (see Canvas for link) |

## Instructor Information

|  |  |
| --- | --- |
| **Name:** | Adam Carter |
| **Office location:** | BSS 324 (Next to Sharon) |
| **Office hours:** | Tuesdays & Thursdays 10:30-12:00; 1:00-3:40; Also by appointment |
| **Contact information:** | email: adam.carter@humboldt.edu |

## TA

|  |  |
| --- | --- |
| **Name:** | None! |
| **Office location:** |  |
| **Office hours:** |  |
| **Contact information:** |  |

## Textbooks

Cormen, Leiserson, Rivest, & Stein – Introduction to Algorithms (3rd edition) [FREE through HSU](https://humboldt-primo.hosted.exlibrisgroup.com/primo-explore/fulldisplay?docid=01CALS_ALMA51438951350002901&context=L&vid=01CALS_HUL&lang=en_US&search_scope=01CALS&adaptor=Local%20Search%20Engine&tab=books_csu&query=any,contains,cormen&sortby=rank&mode=Basic&offset=0)

## Other Materials

Turning point clickers may be required.

## Course Outline

Introduction to algo­rithmic thinking. Recurrences and solution tech­niques, fundamental algorithms including graph algorithms, algorithm design techniques, balanced trees, performance trade-offs.

Note that MATH 253 is a prerequisite to CS 211! By the time you take this class, you should have been exposed to the following: inductive proofs, proof by contradiction, linked lists, stacks, queues, recursion, binary trees and binary tree traversals (inorder, postorder, preorder), hash tables, binary search, and a couple simple sorting algorithms. You should know a good deal about logarithms and exponentials, and the graphs of functions. *If you are missing something from this background, you will need to study up on these items right away.*

This course builds upon prior coursework by rigorously exploring algorithmic analysis and design.

## Student Learning Outcomes and Assessment

This course addresses departmental learning outcomes of:

• Computational Thinking

• Technical Writing

• Communicating and Collaborating

This course addresses computational thinking at a moderate level, adding the next level of computational maturity through its coverage of classic data structures and algorithmic analysis, design, and critique. It addresses technical writing and communicating at a moderate level via program documentation, coding standards that stress reusable code, and reflection essays. By the end of the course, students should be able to:

* Discuss common algorithms for sorting, order statistics, matrix operations, elementary graphs (including shortest path and minimum spanning tree algorithms), and number theoretic problems.
* Apply design techniques such as divide-and-conquer, chip-and-conquer, dynamic programming and greedy method to new problems in order to generate new algorithms.
* Analyze the performance of algorithms using amortized, recurrence based, and experimental methods and explain the advantages and disadvantages of each approach.
* Decide which of many advanced data structures are appropriate for solving particular problems.
* Give examples of NP-Complete problems and discuss trade-offs to solving them with parallel and probabilistic algorithms.

## 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.

## Grades

Grades will be based on the following (weights are in parenthesis):

* Participation (5%)
* Github (5%)
* Programming assignments, written homework, and micro assignments (45%)
* Midterm exams (20%)
* Final exam (25%)

Letter grades will be assigned according to the following scale:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 93.00 – 100.00 | A |  | 71.00 – 75.99 | C |
| 90.00 – 92.99 | A- |  | 69.00 – 70.99 | C- |
| 87.00 – 89.99 | B+ |  | 60.00 – 68.99 | D |
| 82.00 – 86.99 | B |  | 00.00 – 59.99 | F |
| 78.00 – 81.99 | B- |  |  |  |
| 76.00 – 77.99 | C+ |  |  |  |

### Homework Policy

All homework is to be turned in on the due date. No late submissions will be accepted. However, each student is given a pool of 40 bonus homework points that can be spent to raise low grades.

### Participation

Attendance in class is mandatory and will be collected throughout the semester. Periodically throughout the semester, I will require you to answer questions posed on the course website. Additionally, certain in-class activities may be awarded participation points.

### Github

You will be expected to use a private github repository to house your solutions. Regular updates to your github will get earn you full credit in this category.

### A Note on Cheating

The digital nature of computer science can make cheating a very tempting option. Remember that cheating can result in a failing grade, and perhaps more importantly, ***result in academic***. In order to make sure that this doesn’t happen, I've included some potential scenarios to help guide your behavior.

1. You and some friends are working together on the next homework and you just finished the assignment. Unfortunately, you have to head off to work. A friend who isn't finished asks to get a copy of the assignment from you so that he can use it as a guide. Wanting to be a good friend, you email him a copy of your assignment on the guarantee that he won't copy. ***This is cheating***. You are not allowed to give copies (analog or digital) of your work. More often than not, your friend will copy and paste some of your work into his assignment. This will likely show up when I run my cheat detection software. ***This is a very common way to get in trouble. Don’t do it!***
2. You and some friends are getting together after class to discuss the assignment due tomorrow. You're done and want to help your friends out. To do so, you discuss the assignment using high-level descriptions (e.g., "I have class X that does Y, which helps me get task Z done") and even draw some nice pictures and/or flowcharts for them to follow. ***This*** ***is not*** cheating and is actually encouraged.
3. As part of a lab, you're required to implement a poker game. A quick Google search reveals several existing implementations. You download the project and modify it to suit the requirements of the lab. ***This is cheating.***
4. Your friend is almost done with his homework assignment but has run into a nasty debugging issue. You take a look at his program and quickly realize that he isn't correctly deleting a dynamically created pointer. You show him how to fix is error. ***This is not*** cheating and is actually encouraged!
5. A student has asked for code help on the course's online discussion forum. You respond with a partial code snippet. ***This is not*** cheating and is actually encouraged!
6. During an exam or other testing environment, you use your phone to ask someone else a question. Alternatively, you receive an email or text message from someone in class asking for help and you respond. ***Both cases are considered to be cheating.*** Communicating in any manner during a testing environment is not allowed.
7. You and a friend are working together on an assignment using the same computer. As a result, both of you turn in the same or nearly identical assignments. As I cannot determine who actually did the work, I must consider this ***to be cheating***. Remember, you must turn in original work typed out entirely by yourself.
8. A friend of yours has taken the class in a prior semester. You ask him for all of prior homeworks and labs, which you use to aid you in completing your assignments. ***This is cheating***.

If you find yourself in a situation that is not listed above and are worried about whether or not it might be considered cheating, error on the side of caution and don't do it. Instead, email me (I usually respond fairly quickly). I'll give you a definitive answer and add it to the list above.

# Students with Disabilities:

If you have any type of disability that may hamper your full participation in course activities, it is your responsibility to inform me of your need for accommodations as soon as possible. I expect to hear from you **within the first two weeks of the semester** so that appropriate accommodations can be arranged.

Complete information on the services available at HSU can be found at the **Student Disability Resource Center** in the basement of the Library, Library 055, **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 qualify for extra time on exams or need other accommodations, it is your responsibility to obtain and provide me with the Exam Accommodation Request form from the SDRC. The form must be presented to me in a timely manner so appropriate arrangements can be made in advance for all exams. I strongly recommend submitting the form at the beginning of the semester or at least one week before the first exam that you wish to use accommodations. It is also strongly recommended that you communicate eligible accommodations and scheduling arrangements with me one week prior to each exam. This will ensure your test accommodation arrangements are completed in a timely manner.

# 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 that the CSU (and thus HSU) policies on withdrawing from and repeating courses changed as of Fall 2009:

* 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.
* Students may repeat courses only if they earned grades lower than a C.
* Students may repeat up to 16 semester-units with grade forgiveness.
* Students may repeat up to an additional 12 semester-units with grades averaged.

**Be careful** – as of Fall 2009, HSU is being much more strict about what constitutes a “serious and compelling reason”.

If you do drop the course, note that it is **your responsibility** to complete and submit the appropriate paperwork. As noted in the University policies for dropping courses, "As a matter of university policy, **the instructor in the course may not drop on behalf of the student**."

# 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).

# Time Expectations:

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 12 hours a week spent outside of class for this 4-credit course.

# Academic Honesty:

Students are responsible for knowing policy regarding academic honesty:

http://www.humboldt.edu/studentrights/academic\_honesty.php

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

# Additional University Information:

* **Campus Policies:** HSU maintains a complete list of campus policies at the following website:  
  http://www2.humboldt.edu/academicprograms/syllabus-addendum-campus-resources-policies
* **Attendance and disruptive behavior:** Students are responsible for knowing policy regarding attendance and disruptive behavior:   
  http://www.humboldt.edu/studentrights/attendance\_behavior.php
* **Late arrival to class:** Please attempt to come to class on time, with your headphones put away and your cell phones 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