

Syllabus in Fall 2019 for CS 202 Computer Science II

Contact Your Instructor

Name: *Jeff Kinne, Gage Golish*

Email: jkinne@cs.indstate.edu, ggolish@sycamores.indstate.edu

Office: Root Hall A-140D (Kinne), A-015 (Golish)

Lecture, Exam, Office Hours

Lecture: TWRf 11-11:50am in Root Hall A-017

Section and CRN: 50801

Credit Hours: 4

Exam: Dec 12 10-11:50am **and** Dec 13 10-11:50am,
Also check the [Office of the Registrar's exam schedule](#)

Instructor Office Hours:

Jeff Kinne - MW noon-3pm, TR 9-11am

Gage Golish - see hours at cs.indstate.edu

Tutoring: See <http://cs.indstate.edu/info/labs.html>

Website: <http://cs.indstate.edu/~jkinne/cs202-f2019>

Instructor/class Directories (cs.indstate.edu):

`/u1/h0/jkinne/public_html/cs202-f2019/code`

Prerequisites

C or better in CS 201

Recommended (not required)

- [Introduction to Algorithms](#) by Cormen, Leiserson, Rivest, and Stein
- [The C Programming Language](#) by Kernighan and Ritchie
- [GNU Make](#)
- <http://www.cplusplus.com>
- And will supplement with other online sources

Course Announcements

Announcements regarding the course will be made both during class and via email to your @sycamores.indstate.edu email address. You should regularly check this email account or have it forwarded to an account that you check regularly.

Classroom conduct

You may not use cell phones, iPods/music players, etc. during class. You should be civil and respectful to both the instructor and your classmates, and you should arrive to class a few minutes before the scheduled lecture so you are ready for lecture to begin on time. You may use your computer during class if you are using it to follow along with the examples that are being discussed. You may not check email, facebook, work on other courses, etc. during class.

Course Description

The official description of this course from the catalog is

“This course is a continuation of CS 201. It involves a deeper study of programming languages, but emphasizes programming in a particular language. Topics include algorithm design and analysis, data structures, recursion, threads, network programming, graphics, security, and ethics.”

Most important prerequisite skills/knowledge: basic programming concepts, ability to use unix.

Most important skills/knowledge gained: C programming and debugging, data structures understanding (can explain lookup/insert/delete for each DS, work small examples on paper, know running times) and coding (can finish partially complete DS code, write new functions to traverse DS).

Course Outline

Note - the precise order of topics is subject to change, but this is what we plan to complete by the end of the course.

- Review and New (3 weeks)
 - C Programming - all keywords and operations, libraries and header files
 - Data formats and number systems (binary, decimal, octal, hex), signed/unsigned, two's complement, floating point, long, int, short char, C strings
 - I/O with different data formats and number systems
 - Basic data structures - unsorted array, sorted array, linked list, heap, stack, queue - code for *all* in class and as assignments
 - Search and sorting - linear and binary search, selection and insertion sort, mergesort, heapsort, quicksort - code for *all* in class and as assignments
- Asymptotic analysis and running time (1 week) - big O/Omega/Theta, little o/omega, definitions, basic proofs, polynomials / exponentials / logarithms, recursion trees, measuring running time
- Memory organization (1 week) - text, data, heap, stack, malloc, free, pointer arithmetic
- Unix and debugging (1 week) - gdb, grep, sed, awk, find, compiler options, makefiles, unix regular expressions, reading manual and manual organization, pipes, I/O redirection
- Disk organization (1 week) - inodes, superblocks, etc.
- New data structures (4 weeks) - binary search tree (including balanced), hash table, skip list, trie, B tree - code for *some* in class and as assignments

- Graphs (1 week) - terminology, adjacency matrix, adjacency list, basic algorithms (BFS, DFS, shortest path, minimum spanning tree) - do *all* algorithms on the board, code for *some* in class, running time for *all*
- Classes and objects in C++ (2 weeks) - data abstraction and encapsulation, defining classes, creating objects, inheritance, protection, virtual functions and polymorphism
- The Standard Template Library (1 week)
- Comparison of C++ and other OO Languages (Java, Python, etc.) (1 week)

Learning Outcomes

- C programming - complete knowledge of the language - can take a specification and produce correct code, can take existing code and make changes/updates, can correctly solve all easy/medium autograded problems and a large number of hard problems, good working knowledge of “functions every citizen should know” and other C standard library functions.
- Memory and disk - complete understanding of how C organizes memory (code, data, stack, heap) and implications for efficiency, program correctness and security. Good working knowledge of ext2 disk organization - inodes, superblocks.
- C++ programming - moderate knowledge of OO programming in C++ and use of STL - can use the proper STL objects for different purposes (including for efficiency), and develop an OO framework to solve a problem including proper relation between classes, can correctly solve all easy/medium autograded OO problems and a large number of hard problems.
- Developing on Linux - good working knowledge of GNU make and gcc/g++ and able to correctly develop makefiles and use compiler options.
- Basic data structures - complete mastery of basic data structures (arrays, linked list, stack, queue, binary tree, hash table, heap / priority queue, set) - can write the code for methods and properly use the appropriate data structure in different situations.
- More data structures - good working knowledge of additional data structures covered in class (e.g., skip list, trie, AVL tree, red black tree, B tree), can decide on the best for different situations and correct analysis of efficiency.
- Algorithms - good working knowledge of algorithms for data structures and basic algorithms techniques (brute force exhaustive search, greedy, dynamic programming, randomized) and basic graph algorithms (DFS, BFS, shortest path, MST).

Expected Amount of Work

If you take this class seriously and get what you should out of it, some weeks you will likely be spending around **5-20 hours** on the class (a good CS curriculum includes a number of courses where you work on programming projects nearly non-stop at times for a few days). The students who get A's in their CS courses and have an easy time finding jobs do spend this much time on this course. Not everyone would need to spend this much time and not all weeks will be the same, but you should plan on putting in whatever time it takes. Note that the federal government definition of 1 credit hour as requiring 2 hours worth of time on the course for each credit hour of lecture, so you should think of this as the default for all of your courses.

This is the foundation for the rest of CS, so it definitely pays off to do your best here. **Note - this is your most important class, by far (for CS majors).**

Note - your classes should be more important than your part-time job.

Grading and Assignments

The students of this course have the following responsibilities: read assigned readings before lecture, attend lecture, complete homework assignments, take in-class quizzes, take exams, and complete a project.

Your total/final grade will be computed as -

- Exams - 50%
 - If we have two exams plus a final, the weighting will be 10% exam1, 15% exam2, 25% final exam.
 - Earlier exams will be dropped if you do better on later exams - if exam1 is lower than the others it will be dropped, if the final is higher than exams1&2 then only the final will count.
 - No late exams will be given.
- Programs / HWs / quizzes - 30%
 - Program / HW / quiz grade will be calculated proportionally - assignments with more points will count more towards the grade.
- Attendance - 20%
 - Calculated based on the number of times you were present as a fraction of the days that attendance was taken.
 - If you are late to class or leave early, you will miss these points.

Programs/HWs will normally be due by 11:59pm Monday evening. Solutions will often be either discussed in class or posted online on Tuesday.

Expect a quiz roughly once per week. Expect unannounced quizzes on material that was assigned for reading.

CS Course Policies

Note that this course follows all standard CS course policies. In particular check the CS course policies related to - cheating/plagiarism, attendance, missing exams. See <http://cs.indstate.edu/info/policies.html> for details.

Late Homeworks

Late work on programs/HWs will not be accepted.

Start Homeworks Early

We suggest attempting a homework assignment the day it is given, or the day after, so that if you have a problem you can ask early. If you continue to have problems in trying to complete the assignment, you will have time to ask again. Many of the homework assignments require thought and problem solving, which takes “time on the calendar” not just “time on the clock”. By that we mean that spending two hours on 3 consecutive days may be more productive than trying to spend 6 hours at once on the assignment.

Grade Cutoffs

We try to design homework assignments and exams so that a standard cutoff for grades will be close to what you deserve. After the first exam a grade will be created in Blackboard called “Letter Grade” that is what your

letter grade would be if the semester ended today. Initially, I will likely assign the following grades: 93-100 A, 90-93 A-, 87-90 B+, 83-87 B, 80-83 B-, 77-80 C+, 73-77 C, 70-73 C-, 67-70 D+, 63-67 D, 60-63 D-, 0-60 F

Our goal is that the different grades have the following rough meaning.

A+/A

You can do *all* the assignments *on your own*.

B+/A-

You understand nearly everything, and should be all set to use this knowledge in other courses or in a job.

B-/B

Most things you understand very well and a few you might not (more towards the former for a B and more towards the latter for a C).

C/C+

Learned enough and have the minimum skills to move on in the subject.

D+/C-

You did put some effort in, and understand many things at a high level, but you haven't mastered the details well enough to be able to use this knowledge in the future.

D-

Students will normally *not* get an F if - you attend 80% of the lectures, complete some of the assignments up through the end of the course, and get nearly half of the problems on the final exam correct.

F

Normally, students that get an F simply stopped doing the required work at some point.

Blackboard

The course has a blackboard site. Click [here](#) to go to blackboard. You should see this course listed under your courses for the current term. The blackboard site is only used for giving you your grades (go to the course in blackboard, then click "My Tools", and then "My Grades"). All course content, schedule, etc. is kept in this google doc (which you are currently viewing).

Academic Integrity

Follow the standard CS course policies in terms of what is and is not allowed on assignments:
<http://cs.indstate.edu/info/policies.html>

Please ask the instructor if you have doubts about what is considered cheating in this course.

Special Needs / Student Disabilities

Standard language included in the syllabi for ISU courses.

Indiana State University recognizes that students with disabilities may have special needs that must be met to give them equal access to college programs and facilities. If you need course adaptations or accommodations because of a disability, please contact us as soon as possible in a confidential setting either after class or in my office. All conversations regarding your disability will be kept in strict confidence. Indiana State University's Student Support Services (SSS) office coordinates services for students with disabilities: documentation of a disability needs to be on file in that office before any accommodations can be provided. Student Support Services is located on the lower level of Normal Hall in the [Center for Student Success](#) and can be contacted at 812-237-2700, or you can visit the ISU website under A-Z, [Disability Student Services](#) and submit a Contact Form. Appointments to discuss accommodations with SSS staff members are encouraged.

Once a faculty member is notified by Student Support Services that a student is qualified to receive academic accommodations, a faculty member is obligated to provide or allow a reasonable classroom accommodation under ADA.

Disclosures Regarding Sexual Misconduct

Standard language included in the syllabi for ISU courses.

Indiana State University fosters a campus free of sexual misconduct including sexual harassment, sexual violence, intimate partner violence, and stalking and/or any form of sex or gender discrimination. If you disclose a potential violation of the sexual misconduct policy I will need to notify the Title IX Coordinator. Students who have experienced sexual misconduct are encouraged to contact confidential resources listed below. To make a report or the Title IX Coordinator, visit the Equal Opportunity and Title IX website: <http://www.indstate.edu/equalopportunity-titleix/titleix>.

The ISU Student Counseling Center – HMSU 7th Floor | 812-237-3939 | www.indstate.edu/cns

The ISU Victim Advocate – Trista Gibbons, trista.gibbons@indstate.edu

HMSU 7th Floor | 812-237-3939 (office) | 812-230-3803 (cell)

Campus Ministries - United Campus Ministries | 812-232-0186

<http://www2.indstate.edu/sao/campusministries.htm>

www.unitedcampusministries.org | ucmminister2@gmail.com

321 N 7th St., Terre Haute, IN 47807

For more information on your rights and available resources

<http://www.indstate.edu/equalopportunity-titleix/titleix>