

# CS 500

---

CS 500 Fundamentals of Programming is taken by CS MS students during their first term in the MS program. The main outcomes of the course are proficiency in C programming and data structures and algorithms.

It is assumed that students starting the course have passed the CS MS admissions process (see [Graduate Admissions Interview](#), in particular those taking the course should be proficient in some programming language). Those who are looking for an introductory programming course should consider [CS 510](#) instead.

This page contains the syllabus for CS 500 and is used to keep track of assignments, etc. as well for the most recent offering (fall 2022).

## Contents

---

### General Information

#### Announcements/Assignments/Quizzes

- Assignments

- Announcements

#### Course Description and Content

#### Grading and Assignments

#### CS-Specific Items

- CS Course Policies

- Lab Help

- Course Announcements

- Classroom conduct

- Academic Integrity

- Office hours (using Teams)

- Canvas

- Lectures (using Zoom)

- Participating online

#### ISU Required Syllabus Items

- COVID-19 Information

- Special Needs / Disability Services

- Disclosures Regarding Sexual Misconduct

## General Information

**Course website** - [https://cs.indstate.edu/wiki/index.php/CS\\_500](https://cs.indstate.edu/wiki/index.php/CS_500)

**Your Instructor**

Jeff Kinne (<https://kinnejeff.com>), [jkinne@cs.indstate.edu](mailto:jkinne@cs.indstate.edu) (<http://mailto:jkinne@cs.indstate.edu>)

Office: Root Hall A-142 and in Microsoft Teams, phone 812-237-2126

Instructor Office Hours: MWF 10am-1pm, MTWRF 2-3pm

## Lecture, Exam

Lecture: TR 9:30-10:45 in Root Hall A-017, over Zoom (link in Canvas, see below), and recorded

Mid-term exam: TBA

Final exam: Tuesday, Dec 6, 10-11:50am

*Asynchronous students:* For students who will be mostly participating asynchronously even though the course is being offered synchronously, you should pick a regular time each week to check in with the instructor. Make an appointment with the instructor during the first 2 weeks at this time to make sure you are on track. Each week at this time, write an email or Teams message to the instructor to let them know how things are going and if you have any questions.

**Prerequisites** - admission to the CS MS program or approval of instructor, should be a competent programmer in some programming language.

**CRN numbers** - 53200 for the 001 face to face section, 52021 for the 301 online section

## Required text

- We will use the following free online sources.
- For C programming - **Essential C** (<http://cslibrary.stanford.edu/101/EssentialC.pdf>) by Nick Parlante, C Reference (<https://en.cppreference.com/w/c>) at [cppreference.com](http://cppreference.com)
- For data structures and algorithms - **Open Data Structures** (<https://opendatastructures.org/>) by Pat Morin, Algorithms (<http://jeffe.cs.illinois.edu/teaching/algorithms/>) by Jeff Erickson
- For math content - Building Blocks for Theoretical Computer Science (<https://mfleck.cs.illinois.edu/building-blocks/index-sp2020.html>) by Margaret M. Fleck, Mathematics for Computer Science (<https://courses.csail.mit.edu/6.042/spring18/mcs.pdf>) by Eric Lehman, F Thomson Leighton, and Albert R Meyer
- Additional sources - as needed.

**Class notes** - Notes during class will mostly be kept in the **CS 500 OneNote notebook** ([https://sycamoresindstate-my.sharepoint.com/:o:/g/personal/jeffrey\\_kinne\\_indstate\\_edu/EgVH\\_IYlo3pMqngNbvzxjBUBgjEgrbOPFUK7Moza5vLybA?e=Da3CKu](https://sycamoresindstate-my.sharepoint.com/:o:/g/personal/jeffrey_kinne_indstate_edu/EgVH_IYlo3pMqngNbvzxjBUBgjEgrbOPFUK7Moza5vLybA?e=Da3CKu)) and might be made available later as a PDF. Note that you will need to authenticate with your ISU account to view the notebook. Programs that we develop during lecture will be available on the CS server here: <https://cs.indstate.edu/~cs500/>. If you are logged into one of the CS systems in a terminal, you can get to these files by doing: `cd ~cs500/public_html/`.

# Announcements/Assignments/Quizzes

This section will be kept up to date with announcements of assigned reading, assignments, quizzes, etc. This will be kept as a "stack" with the most recent at the top of the list.

## Assignments

---

- File transfer between personal computer and server. Instructions a bit later.
- Text editors on your personal computer. Instructions coming a bit later.
- Install C on your personal computer. Instructions coming a bit later.
- Quiz over Linux and CS Systems - Getting Started - commonly used commands, operators when running commands, etc. When I get a chance to put it in.

- Using the GNU debugger. When I get a chance...
- Quiz over all C operators and keywords. When I get a chance to put it in.
- HW where you fix common bugs in programs. When I get a chance to put it in.
- Classic C programs - overflow/max values, command-line arguments, calculator with all possible operations, swap function, print data as char/int/float/address/C-string, different types of memory, read into array / print out in reverse order, read from file / sort / write out. Coming soon.
- **C Basics Quizzes** - you will have the quizzes listed under assignments at **C Basics**. You can take the practice quizzes as many times as you like, but the quizzes in our canvas course just once. Note that the "fix errors" quiz involves a fair amount of debugging work, so get started on that one soon. Due Sept 12.
- **h1\_C** ([https://cs.indstate.edu/~cs500/assign/h1\\_C/](https://cs.indstate.edu/~cs500/assign/h1_C/)). Login to your cs500 account in the terminal, and use `handin --checkout h1_C` to checkout the assignment. When complete, run `submit` from within your `h1_C` directory to submit. See [Handin](#) and [C Starting](#). Due Sep 6.
- **Text Editors Terminal** assignment. Due Aug 26.
- **Hello unix lab**. Follow instructions in [Hello Unix Lab](#). Due Aug 24.
- Make sure your personal computer can connect to the CS server. Follow the directions in [CS Server - Terminal](#) and see Canvas for the submitting the assignment. Due Aug 19.
- Login to the CS systems. See Canvas for the details, and [CS Accounts and CS Lab Computers](#) for more info. Due Aug 19.

## Announcements

---

- 2022-09-15 - In `h1_C`, I have noticed some programs that either (a) don't give a citation to online resources being used, or (b) are clearly being worked on by multiple people in the class together (again, without giving a citation). Either is plagiarism. Anyone who used any online sources for `h1_C`, worked with someone else in the class, or got significant help from anyone (even if it was one of the GAs) should edit their files to include citations and explanation of anyone they got help from or worked with, and resubmit your files. Do this today. I will then check files again tomorrow. Citation/explanation should go at the top of each file. In terms of needing to cite internet sources, you do not need to if it is general information about how to use a given C function; you do need to cite anything that talks about solving the particular problem.
- 2022-09-13 - no lecture today.
- 2022-09-08 - aside back into some more examples in C.
- 2022-09-08 - ODS sections we will be doing in lecture (for now) - chapter 1 (Intro), 2.1-2.4 (array stack and queue), 3.1-3.2 (linked list stack and queue), 5.1-5.3 (hash table), 6.1-6.2 (binary tree), 7.1-7.2 (randomized binary tree). After that, TBD which parts of ODS we'll continue with.
- 2022-09-08 - `h1_C` - programs are looking better, but let's look again at whether your programs are exactly matching mine. First, run yours and mine on a few different inputs to see if it looks like they are matching. Second, run your program through the `handin hwcheck`:

```

cd ~/h1_C
make # programs need to be compiled before running hwcheck
check
# or
cd ~
handin --hwcheck h1_C
# if you want to save the result into a file,
handin --hwcheck h1_C > hwcheck.txt

```

And then a printout shows if there were any test inputs where your program had different output than mine.

- 2022-09-07 - an easy C assignment as well to practice?
- 2022-09-06 - Jeff - recreate `bigger_pal.txt`.
- 2022-09-06 - see C Basics Quizzes above. I'll put them in our Canvas course at some point, but you can check out the practice quizzes already. For the fixing errors quiz, you can also check out the files with `handin (handin --checkout C_fix_errors)`, but will need to do the Canvas quiz as well when it's released.
- 2022-09-06 - `h1_C` grading - your program should have the exact same output as mine on all inputs, and it needs to be handed in on the system.
- 2022-09-01 - `h1_C reverse.c` - there was an issue with running: `echo -e -n "1\n2\n3" | ./reverse.o` that would happen sometimes and other times not. A `-e` was showing up in the printout. It depends on which shell you are using. If using

tcsh, it has a builtin echo command that doesn't have the -e option, so that just gets printed (what echo does in general). If using bash, it uses a different version of echo that takes -e as an argument (to tell it to interpret \n as newline). Just need to keep in mind when testing the program.

- 2022-09-01 - h1\_C updates. Note that strncpy\_ call in strings.c should have been with max\_length 99. And swap.c copying strings should behave like strncpy in terms of the length. reverse.c - don't worry about the \n or not at the end of the file. Jeff - fix "./jeff\_calc.o 3 """" and \n on the reverse.c. In general - our programs should not crash.
- 2022-09-01 - for Open Data Structures, the html version is missing some of the figures, so check the pdf version if something seems to be missing or improperly formatted.
- 2022-08-30 - h1\_C updates. reverse - what to do if input does not end in a newline - see lecture and reverse.c strcmp - what to return (just +/- 1 or 0, or other +/- values based on the difference between characters) - just +/- 1 or 0, like jeff\_strings.o does. time\_scale.c - change the (if argc < 2) to (if argc < 3).
- 2022-08-30 - reading assignment - Open Data Structures chapters 1 and 2 (introduction, array-based lists).
- 2022-08-30 - reading assignment - the rest of Essential C.
- 2022-08-28 - for homework programs, if you can't figure it out and want to ask me, then hand it in so I can look at the program as well, and send me a screenshot of what you are seeing. FYI in terms of when I tend to reply to messages - starting some time around 7am depending on the day, up until about 4pm (with gaps for classes, meetings, etc.), then again roughly 8-10pm. Weekends I will often have longer periods of not checking messages but try to at least a few times per day.
- 2022-08-26 - reading assignment - Linux and CS Systems - Getting Started and wiki pages linked from there (in particular the ones about using the terminal).
- 2022-08-18 - reading/viewing assignment - 2's complement.
- 2022-08-17 - reading assignment - Essential C introduction and sections 1,2,3.
- 2022-08-06 - creation of this site, including the preliminary list of topics, outcomes, achievements, etc.

# Course Description and Content

## Course Description

The catalog description for this course is: "Review of undergraduate topics in Computer Science including Data Structures, Computer Architecture, and Computer Organization. Review of a computer programming language that students can expect to use in advanced courses." The two main goals are that you leave the course as proficient C programmers and proficient at using and analyzing the most important data structures and algorithms.

## Course Outline

- Getting started - system setup, git, linux, bash, make, apache, math background, development on your personal computer.
- C programming basics - operators, reserved words, data types, base systems, overflow.
- C programming strings - manipulation of C strings.
- C programming memory management - different types of memory in C, how to use them, pros and cons.
- C programming style - good programming style for reliability, readability, extensibility, security.
- Data structures - understanding/use of most important data structures - arrays, linked lists, skip list, binary search trees, red black trees, hash tables, heaps, B tree. Implementation of some of these in C.
- Algorithms - understanding/use of basic algorithms - sorting (various), binary/linear search (and uses), graph algorithms (basic properties, BFS, DFS, MST, shortest path), strings (edit distance) - including some algorithms that are each of - greedy, dynamic programming, heuristic, randomized, brute force / backtracking.
- Vocab - additional terms, algorithms, concepts at a shallow level.

## Learning Outcomes

- System setup - personal computer setup for both remote (connecting to CS server with terminal, sftp, X windows) and local development (editor, compiler).
- Git - basic use of git for source code management.
- Linux - proficient using the Linux terminal for development.
- Bash - basic use of bash scripts for testing code.

- Make - basic use of GNU make for compiling code.
- Apache - able to post content to your CS accounts to be viewable over the web.
- Math background - proficient in math background needed for data structures and algorithms.
- Personal computer - is setup for development so you can do coursework from your home computer as well.
- C programming - understanding of all language features, proficient in writing code using the most common, write code using good programming style.
- Data structures - understanding of operations, efficiency, use cases, can write code in C for data structure operations.
- Algorithms - understanding of basic algorithms, arguments for correctness and efficiency, can use the write algorithms to solve problems efficiently.

# Grading and Assignments

We will be trying out what I am calling "achievements-based" grading. There are a series of skills, knowledge, and experiences that I want you to achieve. Your final letter grades will be based strictly on which of these you have completed. For each achievement, you can achieve the rating of incomplete, pass-, pass, pass+. The following will be our starting point for how letter grades will be assigned. I will reevaluate this throughout the term to make sure we are on track. I will also be setting the standards for pass-, pass, and pass+ for each of the achievements as we get to them in the course.

## C - lowest passing grade in a grad course

- Pass or higher achievement for *all* of the following
- System setup, git, linux, bash, make, apache, math background, personal computer
- C programming basics
- Data structures basic understanding - arrays (stack and queue), linked lists (stack and queue), binary search trees, hash tables
- Data structures programming - can write the code for methods for *some* of the above
- Data structures programming - can use *some* of the above data structures to solve problems
- Sorting algorithms - basic understanding of at least one slow and at least one efficient sorting algorithm
- Sorting algorithms - can write the code for at least one sorting algorithm
- Algorithm techniques - basic understanding of *at least one* greedy, divide and conquer, brute force, and dynamic programming algorithm.
- Asymptotic resource usage - basic understanding, can use Jeff's rules of thumb in simplifying expressions

## B - satisfactory

- In addition to the above...
- Data structures programming - can write the code for methods for *most* of the "C grade" basic data structures
- Data structures programming - can use *most* of the "C grade" basic data structures to solve problems
- Data structures basic understanding - skip list, red black trees, heap, B tree
- Sorting algorithms - can write the code for at least one slow and at least one efficient sorting algorithm
- Sorting algorithms - basic understanding of multiple slow and multiple efficient sorting algorithms
- Graph algorithms - basic understanding of the adjacency lists, adjacency matrices, graph algorithms DFS and BFS
- Asymptotic resource usage - can apply the precise definitions of big-O, little-o, big-Omega, little-omega, big-Theta to simplify and compare expressions; can apply the master theorem *or* recursion tree methods to recursive resource formulae

## A - good/excellent

- In addition to the above...
- Pass+ rating on most of the above
- Pass or higher achievement for *all* of the following
- Graph algorithms - can write the code for some of the "B level" graph algorithms
- Graph algorithms - basic understanding of an MST algorithm and shortest path algorithm

- Asymptotic resource usage - can apply the master theorem *and* recursion tree methods to recursive resource formulae
- Integer algorithms - basic understanding of faster-than-obvious multiplication and exponentiation algorithms
- Algorithm techniques - can write code for at least one algorithm for each of the "C level" algorithm techniques
- Algorithm techniques - basic understanding of *multiple* algorithms for each of the "C level" algorithm techniques
- Complexity theory - basic understanding of the definitions of L, P, NP, PSPACE, EXP and a few canonical problems in each

Achievements can be earned based on quizzes, assignments, in-class work, and exams. Rather than having numerical scores for these, I will use them to mark off your achievements. Note that achievements can be "lost" if you demonstrate a skill early in the term and then demonstrate a lack of the skill later in the term. I expect this will not normally be the case, but I will continue to evaluate you based on all of the skills throughout the term.

**Late Work** - Assignments will generally be available to still handin for around a week after their due date. Once the solutions are posted and discussed, late submissions will no longer be graded. Quizzes will normally need to be taken on the day they are due, or perhaps within a few days of when they are due. Solutions will normally be discussed or posted within a week of their due date. Not accepting late work that is more than about a week old is in part because it takes much longer to grade quizzes/assignments that are no longer super fresh in the instructor's head, and in part to try to keep everyone in the class working on the same material.

**Start Assignments and Quiz Studying Early** - I suggest attempting an 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 assignments require thought and problem solving, which takes "time on the calendar" not just "time on the clock". By that I mean that spending an hour on 3 consecutive days is likely to be more productive than trying to spend 3 hours at once on the assignment.

**Expected Amount of Work** - My expectation is that an average student will spend about 5-10 hours OUTSIDE of class each week (that is in addition to class time or viewing lecture videos) WORKING PRODUCTIVELY/EFFICIENTLY (not just staring at the computer) to complete their coursework for this class. Some students may spend less time than this, and some students will spend more.

This is the foundation for the rest of CS, so it definitely pays off to do your best here.

Note - please find a way to spend enough time on this class (the investment will pay off in terms of skills, being able to get a job, etc.).

**Grade Meanings** - The letter grades are intended to have the following rough meaning. The list of achievements needed for each was chosen with this in mind.

- A+/A: You understand everything and probably could teach the course yourself.
- B+/A-: You understand nearly everything, and should be all set to use this knowledge in other courses or in a job.
- C/C+/B-/B: Some things you understand very well and others you don't (more towards the former for a B and more towards the latter for a C).
- D-/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. **Note that the lowest grade for grad courses is a C, so if you fall in the range below C then your letter grade will be an F.**
- F: Normally, students that get an F simply stopped doing the required work at some point.

## CS-Specific Items

This section contains items that are generally the same for all CS courses (and in particular those taught by this instructor).

# CS Course Policies

---

Note that this course follows all standard CS course policies. In particular, (a) cheating/plagiarism by graduate students results in an F in the course, (b) and there will be no makeup exams. See <http://cs.indstate.edu/info/policies.html> for details.

## Lab Help

---

We have a few lab assistants who are available to help students in beginning computer science courses. Please see [https://cs.indstate.edu/wiki/index.php/Unix\\_Lab\\_and\\_Help](https://cs.indstate.edu/wiki/index.php/Unix_Lab_and_Help) for details. The lab hours are in a calendar on the CS homepage, at [http://cs.indstate.edu/info/index.php#lab\\_hours](http://cs.indstate.edu/info/index.php#lab_hours). You can join the lab when working on your programs. You can ask the lab assistants to look at your programs, and you can work with any other CS students that are there (you could use the lab as a regular meeting place to work with your classmates).

## 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. You can set the account to forward by logging into your indstate.edu email online (if you aren't able to find the option, try a different browser or search online for things like - outlook online forward email setting).

## 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 should avoid spending time on email, Facebook, work on other courses, etc. during the lecture for this class (be fully present wherever you are, make the most of each experience).

## Academic Integrity

---

Please follow these guidelines to avoid problems with academic misconduct in this course:

*Homework:* You may discuss the homework assignments, but should solve and finish them on your own. To make sure you are not violating this, if you discuss with someone, you should DESTROY any work or evidence of the discussion, go your separate ways, SPEND at least an hour doing something completely unrelated to the assignment, and then you should be able to RECREATE the program/solution on your own, then turn that in. If you cannot recreate the solution on your own, then it is not your work, and you should not turn it in.

*Note on sources:* if you use some other source, the web or whatever, you better cite it! Not doing so is plagiarism.

*Exams:* This should be clear no cheating during exams. Each instructor has different rules for what is allowed on exams in terms of notes, etc. If not noted otherwise, you should assume that a quiz or exam is closed notes, no computer, no calculator.

*Projects:* You should not copy from the Internet or anywhere else. The project should be your own work. It will be fairly obvious to me if you do copy code from the Internet, and the consequences will be at the least a 0 on the project. If cheating is observed, you will at the least receive a 0 for the assignment (and may receive an F for the course), and I will

file a Notification of Academic Integrity Violation Report with Student Judicial Programs, as required by the university's policy on Academic Integrity. A student who is caught cheating twice (whether in a single course or different courses) is likely to be brought before the All University Court hearing panel, which can impose sanctions up to and including suspension/expulsion. See <http://www.indstate.edu/sjp/docs/code.pdf> and <http://www.indstate.edu/academicintegrity/> for more information.

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

## **Office hours (using Teams)**

---

Office hours will be through Microsoft Teams by default. If you would like to meet in person you should reserve an appointment using <http://cs.indstate.edu/scheduler> to reserve an in person meeting with Jeff Kinne. I am normally in my office during my listed office hours, but by making an appointment you can be more certain. For meeting through Teams, you should start Teams in your browser or start the application. You should be logged in using your ISU credentials. Once you have Teams open you can message me to ask me questions or to ask to talk. We can use Teams to message (better than emailing back and forth repeatedly if you have questions about something that you just want to write about) or to talk and share screens (e.g., to take a look at your code). I normally have Teams open on my computer all of the time, including during my office hours. During my office hours I will normally reply right away; at other times I will reply when I get a chance.

## **Canvas**

---

The course has a canvas site. Click <https://indstate.instructure.com/> to go to canvas. You should see this course listed under your courses for the current term. If you don't you may need to click on the Courses icon and then click the "All courses" link. The canvas site is used for giving you your grades, for quizzes/exams, and for getting to online lectures (which are done using Zoom). Announcements will be sent through canvas and to your university email. Links and such will be kept on this website.

## **Lectures (using Zoom)**

---

Here at ISU section numbers starting with the number 3 (e.g. 3xx: 301, 302, etc.) are generally online sections. There are 2 types of online sections, synchronous online and asynchronous online. Sections that are synchronous should be joined at the regularly scheduled time of the course, whereas sections that are asynchronous generally keep up with the material independently without regularly scheduled meetings. In general async sections are more difficult to stay on top of, and require a great deal of self-discipline (it is much easier to think "I can watch the videos tomorrow" and just get behind). So if you are in one of these sections make sure you get off to a strong start, and ask for help sooner rather than later. If you are in an online section, check your course schedule for course meeting times; if you have a meeting time, then your section is synchronous, otherwise it is asynchronous (or there is an error in the system).

This course has a 301 section (synchronous online) and 001 section (face to face). Students in either section can participate in whatever way you need to.

For ISU's links to information on getting started with Zoom, see <https://indstate.teamdynamix.com/TDClient/1851/Portal/KB/ArticleDet?ID=107534>. You can also see the information linked at <https://www.indstate.edu/services/student-success/cfss>. You will get to the lectures for this course by going to Canvas, select this course, click Modules on the menu on the left, and click on the Zoom module. Once there you should see a schedule of lectures and be able to view recorded lectures. Note that you should install the Zoom application for your



computer, and you will need to be logged into to Zoom with your ISU credentials to be able to connect. Also note that the lectures are recorded and only available to those in our class. Recorded lectures normally appear later the same day as the lecture.

Note that if you have not used Zoom with your ISU account previously, you need to go to <https://indstate-edu.zoom.us> and login with your ISU email address and password to get it setup.

## **Participating online**

---

If you are participating online, please see the information at <https://www.indstate.edu/services/student-success/cfss> about participating in online courses. You are expected to either join lectures live through Zoom or watch the recordings once they are available. You will complete assignments, quizzes, and exams on the same schedule as the rest of the class. For quizzes and exams you will normally have a 24 hour period during which to take the quiz/exam (note that different students will have slightly different questions and any communication between students about quiz/exam content is academic misconduct).

So also the General Information section at the top of this page for setting up a normal check-in time with the instructor.

## **ISU Required Syllabus Items**

The items in this section are required and are the same for every ISU course.

## **COVID-19 Information**

---

Information specific to CS courses - [Start of Term Announcements](#)

*Standard ISU language required in all syllabi (read this all once, then skim for your other courses)...*

Students are expected to adhere to course attendance policies, as stated in the course syllabus. Documented COVID-related absences will be treated like any other serious medical issue. Following University policy, students with a documented, serious medical issue must contact the Office of the Dean of Students for assistance. The Office of the Dean of Students will supply documentation for faculty. Students with a documented serious medical issue should not be penalized and will be given a reasonable chance to complete exams or assignments. Once notification is made, faculty will make reasonable efforts to accommodate the student's absence and will communicate that accommodation directly to the student. Please note that faculty are not required to accommodate a serious medical issue with virtual content options, like streaming or recorded lectures. To avoid the potential of missing significant class time, students are strongly encouraged to receive the COVID vaccination that has been made available on campus. For more information about the vaccines or to find a vaccination site, go to: <https://ourshot.in.gov>. The ISU Health Center also administers COVID-19 vaccines by appointment.

Students should contact the Office of the Dean of Students with questions by calling 812-237-3829.

The information provided in this section of the syllabus is subject to modification based on guidance by public health authorities. Changes to Covid-related policies or updated information will, as always, be posted on the ISU website and communicated in multiple ways.

## **Special Needs / Disability Services**

---

*Standard ISU language required in all syllabi...*

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 ISU language required in all syllabi...*

Indiana State University Policy 923 strictly prohibits discrimination on the basis of: age, disability, genetic information, national origin, pregnancy, race/color, religion, sex, gender identity or expression, sexual orientation, veteran status, or any other class protected by federal and state statutes in ISU programs and activities or that interferes with the educational or workplace environment.

Title IX of the Educational Amendments of 1972 prohibits discrimination based on sex, including sexual harassment. Sexual harassment includes quid pro quo harassment, unwelcome verbal or physical conduct, sexual assault, dating violence, domestic violence, and stalking.

If you witness or experience any forms of the above discrimination, you may report to:

*Office:* Equal Opportunity & Title IX; (812) 237-8954; Rankin Hall, Room 426

*Email:* ISU-equalopportunity-titleix@mail.indstate.edu

*Online:* [https://cm.maxient.com/reportingform.php?IndianaStateUniv&layout\\_id=10](https://cm.maxient.com/reportingform.php?IndianaStateUniv&layout_id=10)

Disclosures made to the following confidential campus resources will not be reported to the Office of Equal Opportunity and Title IX:

*ISU Student Counseling Center:* (812) 237-3939; Gillum Hall, 2nd Floor

*Victim Advocate:* (812) 237-3829; HMSU 7th Floor

*UAP Clinic/ISU Health Center:* (812) 237-3883; 567 N. 5th Street

---

Retrieved from "[https://cs.indstate.edu/wiki/index.php?title=CS\\_500&oldid=3186](https://cs.indstate.edu/wiki/index.php?title=CS_500&oldid=3186)"

---

**This page was last edited on 14 September 2022, at 14:27.**

Content is available under [GNU Free Documentation License 1.3 or later](#) unless otherwise noted.