

# A Beginner's Guide to Surviving and Conquering CS 121 Spring 2016

## Instructor Info

Name: Nathan Schmidt

(Please call me "Nathan")

Email: [nathanschmidt@boisestate.edu](mailto:nathanschmidt@boisestate.edu)

Office: MB 103

Office Hours:

- 4:00-5:00pm on Mondays and Wednesdays

Degrees:

- Minor in Mathematics
- B.S. in Computer Science
- M.S. in Computer Science
- 3/4 of a M.S. in Mathematics (In Progress!)

## When and where do we meet?

- 1:30-2:45pm in MEC 309 on Tuesdays and Thursdays

## Why should you care about computer science?

Which essential tools of computer science will be useful to you? What about your discipline of interest, future career, and beyond? How can you learn, develop, acquire, and apply these tools to help you solve problems, make decisions, conquer mighty challenges, and achieve your goals? These are examples of how earning the tools of computer science can be advantageous. For this, computer science is a powerful vehicle for training your brain to solve problems and puzzles. In this course, you'll learn to train your "puzzle solving brain" to attack and defeat both theoretical and practical problems in a wide variety of contexts and situations, many of which have real-world application. Yes, computer science is everywhere, and we're always solving problems---even when we're not aware of it! So let's get to it and train that brain!

## How will this course help you succeed?

Admittedly, years after you've completed this course, there is a good chance that you won't remember every single topic that we covered... but this isn't the point! Learning computer science activates and strengthens the regions of the brain involved with decisive mental attributes, such as focus, memory, spatial awareness, critical thinking, and puzzle solving skills. Such earned characteristics for your brain can be useful for effectively combating a broad spectrum of great challenges to obtain solutions; it may be that conquering such challenges will be required for you to advance along your desired career path and succeed in the achievement of your goals. If it was easy, then everybody would do it.

The in-class and out-of-class exercises of this course are interactive and challenging, and are continually proven to be effective tools for training the puzzle solving brain. This course will help you acquire a conceptual and practical framework that you can apply to solve puzzles and problems in your current and future work, and beyond. By the end of this course, you'll be able to answer the following questions:

- How can I use computer science to effectively train my "puzzle solving brain"?
- How can I use the basic tools of computer science to design and create computer programs to solve problems and puzzles?
- How can I "learn how to learn" challenging topics of computer science and develop a deeper insight into basic concepts such as algorithms, abstraction, and encapsulation?
- How might all of the above help me reach my goals?

## What can you do to survive and conquer this course?

What is success to you? Every individual has their own goals, so take a moment and think about how you will succeed in this course. Here are some helpful tips for success:

- **Participate!** Each week, show up to class and lab, where you'll train to tackle challenging problems.
- **Run Solo!** Work in computer labs and perhaps on your own computer (if you choose to set this up) to interactively solve puzzles day or night, at light speed!
- **Collaborate!** Work interactively with your peers and teams to openly discuss, analyze, and solve in-class puzzles. In the workforce and in countless other "real-life" situations, the confidence and ability to collaborate with others in order to attack challenging problems is imperative.
- **Make Mistakes! Learning Is A Process!** Everybody makes mistakes, especially when it comes to disciplines such as computer science or electrical engineering with lots of puzzles! Your instructors will make mistakes, your tutors will make mistakes, your classmates will make mistakes, and even big shots like Elon Musk and Steve Wozniak have made countless mistakes. In fact, it's impossible to learn computer science without making mistakes! You've just got to get out there and give it your best shot. Making mistakes, and learning from those mistakes, is a fundamental component of the training process.
- **Communicate!** Keep the communication lines open with your instructor, your classmates, and your friendly tutors. Don't be afraid to ask questions and openly discuss a topic that has you bogged down. Not only can you receive help from others (who are often struggling just as much as you are), but often just talking through the problem can help organize it in your brain and lead to a strategy or solution that you may not have thought of before. Being able to verbalize and write down your thought process is absolutely essential.
- **Find a Plan and Discipline Yourself!** Computer science is arguably one of the most challenging subjects to learn, but it is also often considered to be one of the respected and rewarding. Come up with a training plan that works for you and stick to it. Each week, be consistent and don't be afraid to work hard and push yourself on a regular basis in order to obtain your goals.
- **Pace Yourself and Take Breaks!** Don't try to solve every problem in a single, epic marathon (for example, avoid being a "weekend warrior"). Eventually, your puzzle solving brain will get exhausted and it'll be *significantly* more difficult to process, absorb, and retain the new material. Break up your training workload and spread it over multiple days throughout the week. Just take it one step at a time. If you get frustrated or confused, or find yourself repeatedly making silly mistakes, then it might be good time to take a break and do something completely different.
- **Attitude!** When faced with a tough problem under dire circumstances, a positive, persevering attitude can help you maintain focus, stay relaxed, and build confidence so you can get the job done and move on.
- **Don't Procrastinate!** In order to conquer this class and reach your goals you've got a certain amount of work to accomplish. You can start early or wait until the last minute---it's up to you. But regardless of your choice, you'll still need to get it done eventually, so why not start early and make life easier later on?
- **... Find Something that Motivates YOU to Succeed!**

# CS 121: Computer Science I

## Spring 2016

### Class Home Page

<http://cs.boisestate.edu/~cs121>

### Catalog Description

**CS 121 COMPUTER SCIENCE I (3-0-3)(F,S)**. Introduction to object oriented problem solving and programming. Software development process. Data and expressions, conditionals and loops, arrays and lists, and classes and interfaces. Introduction to graphical user interfaces and UML diagrams. *PREREQ: MATH 170. COREQ: CS 121L.*

**CS 121L COMPUTER SCIENCE I LAB (0-3-1)(F,S)**. Lab work to accompany CS 121 Computer Science I. *COREQ: CS 121.*

### Learning Objectives

At the end of this course, the student is expected to be able to:

- design object-oriented solutions to programming problems,
- implement working solutions to programming problems using good coding and documentation styles,
- explain basic concepts of computer science such as algorithms, abstraction, and encapsulation, and
- use an integrated development environment that is specialized for program development with reasonable proficiency.

### Text

*Java Foundations: Introduction to Program Design and Data Structures, 3rd edition* by John Lewis, Peter DePasquale, and Joseph Chase.

\* The same text will be used in CS121 and CS221, so you may want to keep it at the end of the semester.

# CS Tutoring Center

Tutors for this course will be available in the computer science labs at scheduled times. Check tutoring center website for CS 121 tutors and their hours.

<http://coen.boisestate.edu/cs/computer-science-tutoring-center-cstc/>

## Piazza

This term we will be using Piazza for class discussion. The system is designed for getting you help quickly and efficiently from classmates, the tutors, and instructors. Rather than emailing questions to the teaching staff, we encourage you to post your questions on Piazza.

Find our class page at: <https://piazza.com/boisestate/spring2016/cs121/home>

Piazza will also be used for disseminating information. Subscription is **required**. Students are responsible for knowing anything posted via Piazza.

## Attendance

Students are expected to attend all classes. Missing classes without explanation may result in a grade penalty.

## Programming Projects

There will be several programming projects throughout the semester. Written communication skills are assessed in documentation for programming projects.

- Programming projects require the implementation of working programs using the language constructs and techniques introduced in class.
- *Programs must be written individually.* Students who copy programs or sections of programs from each other or from any other source will be considered to be cheating as will students who allow their programs to be copied. See Academic Honesty section below for more information.
- Programs must run on the lab server **onyx**. *Any programming project that does not compile and run on onyx will be awarded a score of 0 points.* In order to improve that score, students must spend enough time with the instructor or a tutor to get the program running.

- Programs must be submitted by midnight the day they are due. Late programs are subject to a deduction of 10% every 2 days from the maximum possible score (e.g. a perfect program is worth 100 points if submitted *before midnight* and is only worth 90 points at 12:00 AM the second day). Programs will not be accepted more than 4 days late.

## Exams and Quizzes

### In-Class Quizzes

Quizzes will be given in class over assigned reading and/or material covered in class.

- In-class quizzes must be taken in class on the day they are given. Make-up quizzes will not be granted other than for exceptional reasons.
- Credit for the group portion of the quizzes will only be given to group members who participate in the quiz.
- The individual portion is worth 65% and the group portion is worth 35% of the total quiz score.
- If 80% of students complete the end-of-semester course evaluations, then the lowest quiz score will be dropped.

### Final Exam

The final exam date is fixed and shown on the course website. Unless alternate arrangements are made in advance, only officially excused absences will be accepted for missing an exam. Any resources allowed for exams will be at the instructor's discretion.

## Grading Policy

- Programming Projects: 60%
- In-Class Quizzes: 20%
  - Individual Quizzes: 65%
  - Group Quizzes: 35%
- Final Exam: 20%

# Academic Honesty

Students are expected to work on their own on projects and homework assignments unless explicitly instructed otherwise.

Students who copy from each other or from any other source on assignments will be considered to be cheating as will students who allow their work to be copied. This includes trying to find answers to problems or programs from the Internet or other sources (and uploading your completed assignments to Internet sites that are publicly accessible).

## Official University Academic Dishonesty Policy

Overview reproduced below. The full policy and procedures may be found at <http://deanofstudents.boisestate.edu/academic-dishonesty>

*The term “academic dishonesty” may include cheating, plagiarism, or other forms of academic dishonesty. All assignments submitted by a student must represent her/his own ideas, concepts, and current understanding or must cite the original source. Attempts to violate the academic integrity of an assignment do not have to be successful to be considered academic dishonesty. Academic dishonesty may include, but is not limited to:*

1. Stealing and/or Possessing Unauthorized Material – The unauthorized appropriation, possession or use of the property of another; the forgery or misuse of documents;
2. Fabrication and Falsification – The unauthorized alteration or invention of any information or citation;
3. Multiple Submission – The submission of substantial portions of the same assignment for credit more than once without the prior permission of all involved faculty members;
4. Abuse of Academic Material – Destroying, stealing, or making inaccessible library or other academic resource material;
5. Complicity in Academic Dishonesty – Intentionally or knowingly helping or attempting to help another commit an act of academic dishonesty.

## Procedures for Breach of Academic Misconduct

- First offense: Student will receive a 0 on the assignment and an Academic Misconduct Report Form will be submitted to the Office of the Dean of Students.
- Second offense: Student will receive an F in the course and an Academic Misconduct Report Form will be submitted to the Office of the Dean of Students.
- Academic Misconduct Form: [http://deanofstudents.boisestate.edu/wp-content/uploads/2014/10/Academic\\_Dishonesty\\_Report\\_Form.pdf](http://deanofstudents.boisestate.edu/wp-content/uploads/2014/10/Academic_Dishonesty_Report_Form.pdf).

# CS121 Lab

The required lab component of this course is scheduled and graded separately from the lecture. Labs will be led and evaluated by graduate assistants and tutors. For more information on the lab, see the CS 121 lab page: <https://cs.boisestate.edu/cs121/lab>

## Lab Grading Policy

- Lab attendance is **mandatory**. Missing more than 3 labs will result in automatic failure of the CS 121 lab component.
- Each of the 15 labs is graded on a pass/fail basis.
- After you submit a lab, the lab instructor assigns a grade (pass or fail). Typically this will be done in real time in the lab before you leave.
- The final grade for the CS 121 lab component will be based on the ratio of passed labs to total number of labs (which is 15) (with the exception of those who automatically fail by missing more than 3 labs).
- Example grades:
  - 15 labs attended, 15 passed =  $15/15 = 100\%$
  - 15 labs attended, 13 passed =  $13/15 = 86\%$
  - 14 labs attended, 12 passed =  $12/15 = 80\%$
  - 13 labs attended, 11 passed =  $11/15 = 73\%$
  - 13 labs attended, 10 passed =  $10/15 = 66\%$
  - 12 labs attended, 12 passed =  $12/15 = 80\%$
  - 11 labs attended, 11 passed = automatic failure (too many missed labs)