CSCI 3155 Principles of Programming Languages

D Term Summer 2018

Table of Contents

CSCI 3155 Principles of Programming Languages 1

D Term Summer 2018 1

Staff 2

Instructor: 2

Office Hours: ..... **Error! Bookmark not defined.**

Grader: 2

Course Assistant: 2

Times 2

Course Overview and Goals 2

Prerequisites 3

Schedule 3

Requirements 3

Evaluation 5

Textbook and Resources 5

Tools 6

Integrity of the Course Materials 7

Collaboration Policy 7

Classroom Behavior 7

Discrimination and Harassment 8

Disability 8

Religious Observances 8

Acknowledgments 8

Staff

Instructor:

- Spencer Wilson
- Office : ECST 121
- Always contact via Piazza
- If absolutely necessary call or text me at: 317-363-4190
- Office Hours: See Google Calendar for all office hours

Grader:

- Surya Kancharla

Course Assistant:

- Hartland Brown
- Office Hours: See Google Calendar for all office hours

Times

Lectures: T-R 5:00pm - 6:20pm in KOBL 375.

Recitation: M 5:00pm - 6:20pm in KOBL 375.

Course Overview and Goals

This course is about principles, concepts, and ideas that underlie programming languages. But what does this statement mean?

As a student of computer science, it is completely reasonable to think and ask, "Why bother? I'm proficient and like programming in Ruby. Isn't that enough? Isn't language choice just a matter of taste? If not, should I be using another language?"

Certainly, there are social factors and an aspect of personal preference that affect the programming languages that we use. But there is also a body of principles and mathematical theories that allow us to discuss and think about languages in a rigorous manner. We study these underpinnings because a language affects the way one approaches problems working in that language and affects the way one implements that language. At the end of this course, we hope that you will have grown in the following ways:

- You will be able to learn new languages quickly and select a suitable one for your task.
- You will gain new ways of viewing computation and approaching algorithmic problems.
- You will gain new ways of viewing programs.
- You will gain insight into avoiding mistakes for when you design languages.

We will *dissect* programming languages by constructing interpreters. The semester project is to construct an interpreter for JavaScript (incrementally). We will see that interpreters are the basis for realizing computation, and we will study the programming language theory that enable us to reason carefully about a language's design and implementation. Our approach will be gradual in that we will initially consider a small subset of JavaScript and then slowly grow the aspects of the language that we consider.

Incoming students often expect this course to be what I will call a trip to the Zoo of Programming Languages. While it is certainly interesting to go to the zoo, we seek a more informative and scientific study of the underlying principles. A more apt analogy is an **anatomy** course where we will study the "guts" and inner-workings of programming languages. After this course, such an anatomological study will enable us to compare and contrast programming languages in a substantive manner and address the learning goals outlined above.

The course covers many aspects of using, understanding, and reasoning about programming languages (e.g., syntax, scoping, induction, data types, and typing). We will build up a set of mathematical tools for careful discourse. A significant part is devoted to abstraction, that is, how languages help programming in the large (e.g., subtyping polymorphism, parametric polymorphism, modules, and objects).

This course prepares you for introductory courses on both programming language implementation (e.g., CSCI 4555) and programming language semantics (e.g., CSCI 5535). This course is also a prerequisite for OOAD, TOC, and the senior experience courses in computer science.

Prerequisites

The prerequisites for this course are CSCI 2270 (Computer Science 2: Data Structures) and CSCI 2824 (Discrete Mathematics). It is also suggested that one of CSCI 2400 (Computer Systems) or ECEN 2120 (Computers as Components) be either previously taken or be taken in the same semester.

If you have not already taken these courses or if you have any concerns, please talk with the instructor. Proficiency in programming is absolutely expected, though there is no specific language required. This means that you should be able to pick up a new programming language with relative ease. You will be expected to learn new programming languages with guidance but relatively independently in this course.

Schedule

Important University dates: http://www.colorado.edu/summer/calendars/session-d

Wrought Course Schedule

Week ID	Event
Week 1	Lab 1
Week 2	Lab 2
Week 3	Lab 3
Week 4	Midterm & Lab 3
Week 5	Lab 4
Week 6	Lab 4
Week 7	Lab 5
Week 8	Lab 5
Week 9	Lab 6
Week 10	Lab 6 and Final

Requirements

- **Study Materials**. There will be required articles, notes, or book chapters to read and occasionally videos to watch. This course will emphasize learning from the resources out of class and actively discussing problems and questions on these resources in class.

- **Moodle Exercises** (20%) To aide in your learning we have created daily exercises on Moodle based on reading materials. It is vital that you complete these to be properly prepared for lecture. There will be at least 37 Moodle exercises in this course. Each is graded out of 10 points. Grading information for each exercise will be posted with the exercise. The lowest 4 grades will be dropped. Each exercise and its associated reading should take approximately 45 minutes to complete on average. No late assignments will be accepted except in the case of documented personal, family, or medical emergency.
- **In class exercises** (10%) There will be in-class exercises during each lecture and recitation session. There will be no make-up opportunities for in class exercises. These exercises are graded for 1 point each and are wholly participatory. These will be turned in to the instructor at the end of each class session. The lowest 4 grades in this section will be dropped. No late assignments will be accepted except in the case of documented personal, family, or medical emergency.
- **Lab Assignments** (30%). This course is a project-based course, so most of the understanding will come from attacking the lab assignments. Lab assignments will be variable in length. Because iteration on lab assignments is the basis for learning in this course it is important that you do not fall behind. This semester we will **not** offer a "redo" policy for the labs. If you fail to complete a lab it is your responsibility to use the resources available to you to understand the information covered. Each lab is graded out of 100 points. All labs are weighted equally. Every lab has a code to be submitted to COG and an interview with a member of the course staff. Most (perhaps all) labs will also have a write-up section. No late assignments will be accepted except in the case of documented personal, family, or medical emergency.
    - If the lab has a write-up
        - **10 :** COG auto grader report scaled to 10 points from 60 unless designated otherwise
        - **20 :** Write-up
        - **70 :** Grading Interview
    - If the lab does NOT have a write-up
        - **10 :** COG auto grader report scaled to 10 points from 60 unless designated otherwise
        - **90 :** Grading Interview
- **Exams** (40%) This course has 2 exams to help gauge your understanding of the course materials. Each exam is worth 20% of your total course grade. The final exam is "comprehensive" i.e. it is not "cumulative" but all materials learned in the first half of the course are building blocks for the second half and thus it is expected that you know this material during the final exam. **You must obtain a 60% average on your exams to pass the class, regardless of your grades on your assignments.** There will be no special or make-up examinations for any student except in the case of documented personal, family, or medical emergency.
    - **Midterm Exam** : Tuesday, June 26, 2018 5:00pm – 6:15pm
    - **Final Exam** : Thursday, August 9, 2018 5:00pm – 6:15pm

Grading. Your overall grade will be determined using the ratio for Moodle exercises, in class exercises, lab assignments, and exams shown above. There is no predetermined curve (i.e., I hope everyone gets an A based on the level of mastery demonstrated). Cutoffs will be announced after the midterm exam to give you an idea where you stand.

Regrades. Mistakes happen. Any concern about an error in grading must be submitted for appeal on Piazza in a private post to the "Instructors" within **two days** of when the graded item is returned. Any coursework submitted for reconsideration may be regraded in its entirety, which could result in a lower score if warranted. To request a regrade, please go to the instructor's office hours with your coursework and an explanation of what you believe the grading error to be. If the office hours do not work for you, inform the instructor via a private post on Piazza so that you can set up a meeting time that works for both parties.

Make-Up Exam Policy. There will be no make-up examinations for any student except in the case of documented personal, family, or medical emergency. These rare exceptions must be handled promptly via private discussions on Piazza with the Instructor.

Extra Credit and Participation. Extra credit opportunities will be offered during the course semester.

Teams and Pair Programming. You are asked to work on lab assignments in **pairs** from your team, enabling pair programming. You are also asked to work with a different partner for each lab assignment, enabling the benefit from the experience of your team. Your goal as a team is to help each other successfully complete the course project and the course together. Lab assignments are the main opportunity to learn material in this course and thus they count for a relatively small portion of your final course grade. It is strongly advised that you work on all the problems in a lab assignment together so that you understand all of the material and are prepared for the exam. Everyone will submit assignments, and you must cite your partner explicitly. You are responsible for all assignments individually (e.g., if your partner drops the course midway though an assignment, you still need to submit on time).

Interview Grading. We will use interview grading for the largest portion of your lab assignment. Interviews will be conducted individually with a member of the course staff. **All interviews must be scheduled by Monday at 6am the week of interviews.** Failure to schedule an interview on time will result in a 0 for the interview portion of the lab except in the case of documented personal, family, or medical emergency. Interview scheduler will be posted at least 72 hours in advance of the sign-up deadline. If no time slots will work with your schedule it is your responsibility to perform due diligence in scheduling a time that will work for you and the interviewer – via Piazza. **Interviews will start and end on time.** Please be on-time to your interview slot. We will not offer extra time if you are late. Except in the case of documented personal, family, or medical emergency, if you show up late for the interview you will be graded on what you are able to complete in the time remaining. Except in the case of documented personal, family, or medical emergency, failure to show up for the interview will result in a 0 for the interview portion of the lab and we will not offer make up interviews. In the event of documented emergencies – when possible - please contact the interviewer to let them know your status.

Workload. CSCI 3155 is a 4-credit course over 10 weeks. This translates to a standard expectation of 12 hours of outside of class time per week. Moreover, it is expected that you work 3 hours per class meeting to properly prepare for class, follow up on materials covered, and complete required assignments. For best results, it is recommended that you aim spread your workload over the week and not take on all 12 hours in one day. Please note the 12 hour mark is a rough average of hours per week you should expect to work on the material outside of class. Some weeks will contain material that you find significantly more challenging. It is your responsibility to use as much time as you need to master the course material. For best results in this course it is recommended that you work on the course material 6 days per week for about two hours each day.

Evaluation

Both your ideas and also the clarity with which they are expressed matter—both in your English prose and your code!

We will consider the following criteria in our grading:

- *How well does your submission answer the questions*? For example, a common mistake is to give an example when a question asks for an explanation. An example may be useful in your explanation, but it should not take the place of the explanation.
- *How clear is your submission*? If we cannot understand what you are trying to say, then we cannot give you points for it. Try reading your answer aloud to yourself or a friend; this technique is often a great way to identify holes in your reasoning. For code, not every program that "works" deserves full credit. We must be able to read and understand your intent. Make sure you state any preconditions or invariants for your functions (either in comments, as assertions, or as require clauses as appropriate).

Textbook and Resources

Textbook. The primary study materials for the course are the course notes: Principles and Practice in Programming Languages (found at the top of Moodle), and working title ~ The Gist of PPL (still in progress, fragmented throughout the Moodle exercises).

We also strongly recommend that you get access to Programming in Scala, 2nd edition by Martin Odersky, Lex Spoon, and Bill Venners. This book is an extended tutorial for learning Scala by those directly involved in the language's development. (Link on Moodle)

The course follows many ideas from a supplemental text: Essentials of Programming Languages, 3rd edition by Daniel P. Friedman and Mitchell Wand. We will take liberty to deviate this text where appropriate, as we will use a different implementation language (Scala instead of Scheme). This book is available as an e-book to all University students. (links on Moodle)

Two other useful Scala texts: Atomic Scala by Bruce Eckel and Dianne Marsh; Functional Programming in Scala by Paul Chiusano and Runar Bjarnason. (Link on Moodle)

JavaScript.

- node.js. A JavaScript platform for server applications based on Google Chrome's V8 JavaScript engine. We recommend installing node.js to try out small JavaScript examples on the command-line. Node.js 0.10.24 will serve as our reference JavaScript implementation.
- JavaScript: The Good Parts by Douglas Crockford.
- Mozilla JavaScript overview and reference.
- ECMA-262 standard specification.
- js2-mode for Emacs.

Scala.

- Language Website. The central point for Scala information. The reference implementation for this course is Scala 2.10.5.
- IntelliJ IDEA. You are welcome to use any code development environment, but the course staff will only support IntelliJ. There are many options, including support for Emacs, Eclipse, and NetBeans.
- Scala Style Guide. We will in general follow the Scala style guide.
- An Overview of the Scala Programming Language (2. Edition). This document is a technical overview that assumes background in object-oriented programming.
- The Scala Language Specification. This document is the most detailed specification of the language.

Tools

Moodle. We will use Moodle for grades and feedback. If you do not already have an account, please create one and join the course moodle. The enrollment key is in the welcome e-mail.

Piazza. We will be using Piazza for online, outside-of-class discussion. Rather than emailing questions to the teaching staff, questions should be posted on Piazza. I encourage you to make class-wide posts whenever possible, but there is an option to send an instructor-private message. You also have the option of posting anonymously.

Computing. We strongly suggest you use the standardized development environment on the CU CS Virtual Machine. After obtaining the Virtual Machine, you will need to install the following separately:

- The CSCI 3155 package via

```
$ sudo apt-get update
$ sudo apt-get install cu-cs-csci-3155
```

- IntelliJ IDEA from https://www.jetbrains.com/idea/download/

Integrity of the Course Materials

The development effort in the course materials, including the lab assignments, the exercises, and the exams, is significant. You agree that you will not share any course materials publicly or privately outside of your teams. The course materials, include your or anyone else's solutions to the lab assignments, exercises, and exams. In particular, you agree not to post your solutions to the lab assignments in a public source code repository, such as public Github repositories. Please use private source code repositories for your work.

Note that there is no conflict with the Collaboration Policy described below. You are welcome and encouraged to support each other in the learning of the material.

Collaboration Policy

You are welcome and encouraged to work together in learning the material. If you worked with someone on an assignment, or if your submission includes quotes from a book, a paper, or a web site, you should thank the source. Bottom line, feel free to use resources that are available to you as long as the use is **reasonable** and you **cite** them in your submission. However, **copying answers directly or indirectly from solution manuals, web pages, or your peers is certainly unreasonable**. If you have any doubts in this regard, please ask the course staff.

Academic Dishonesty Policy. We will go by the Honor Code set forth by the University:

All students of the University of Colorado at Boulder are responsible for knowing and adhering to the academic integrity policy of this institution. Violations of this policy may include: cheating, plagiarism, aid of academic dishonesty, fabrication, lying, bribery, and threatening behavior. All incidents of academic misconduct shall be reported to the Honor Code Council (honor@colorado.edu; 303-735-2273). Students who are found to be in violation of the academic integrity policy will be subject to both academic sanctions from the faculty member and non-academic sanctions (including but not limited to university probation, suspension, or expulsion). Additional information regarding the Honor Code policy can be found online and at the Honor Code Office.

Academic sanctions may include, but is not limited to, a zero for the assignment or a failing grade for the course.

Classroom Behavior

We trust and expect everyone to behave in a civil and courteous manner.

In class, the course staff promises their undivided attention and reciprocally expects the same from you. Please be thoughtful of those around you, surfing the web and texting can be incredibly distracting. Notebook computers should be used only for purposes directly relevant to the class discussion (e.g., taking notes). Avoid use of cellphones.

Please notify the course staff if you encounter behavior that distracts from your learning. This Classroom Behavior policy will be altered as needed.

We will also go by the policies set forth by the University:

Students and faculty each have responsibility for maintaining an appropriate learning environment. Those who fail to adhere to such behavioral standards may be subject to discipline. Professional courtesy and sensitivity are especially important with respect to individuals and topics dealing with differences of race, color, culture, religion, creed, politics, veteran's status, sexual orientation, gender, gender identity and gender expression, age, disability, and nationalities. Class rosters are provided to the instructor with the student's legal name. I will gladly honor your request to address you by an alternate name or gender pronoun. Please advise me of this preference early in the

semester so that I may make appropriate changes to my records. For more information, see the policies on classroom behavior and the student code.

Discrimination and Harassment

We will go by the policies set forth by the University:

The University of Colorado Boulder (CU-Boulder) is committed to maintaining a positive learning, working, and living environment. CU-Boulder will not tolerate acts of discrimination or harassment based upon Protected Classes or related retaliation against or by any employee or student. For purposes of this CU-Boulder policy, "Protected Classes" refers to race, color, national origin, sex, pregnancy, age, disability, creed, religion, sexual orientation, gender identity, gender expression, veteran status, political affiliation or political philosophy. Individuals who believe they have been discriminated against should contact the Office of Institutional Equity and Compliance (OIEC) at 303-492-2127 or the Office of Student Conduct and Conflict Resolution (OSC) at 303-492-5550. Information about the OIEC, the above referenced policies, and the campus resources available to assist individuals regarding discrimination or harassment can be found at the OIEC website. The full policy on discrimination and harassment contains additional information.

Disability

We will go by the disability guidelines set forth by the University:

If you qualify for accommodations because of a disability, please submit to the course staff a letter from Disability Services within the **first two weeks** of class so that your needs can be reasonably addressed. Disability Services determines accommodations based on documented disabilities.

If you have a temporary medical condition or injury, see Temporary Injuries guidelines under the Quick Links at the Disability Services website and discuss your needs with your professor.

Religious Observances

We will go by the policy for religious observances set forth by the University:

Campus policy regarding religious observances requires that faculty make every effort to deal reasonably and fairly with all students who, because of religious obligations, have conflicts with scheduled exams, assignments or required attendance. In this class, we will try to accommodate religious conflicts in a reasonable manner. Please check the exam dates and submit all requests for adjustments within the **first four weeks** of class.

See www.colorado.edu/policies/fac_relig.html for further details on the policy.

Acknowledgments

This course has benefited from the organization and experience from prior versions of this class taught by Evan Chang, Jim Baker, Thomas Nelson, Amer Diwan, Michael Main, William Waite, Martin Hirzel, and Clayton Lewis.