

Syllabus

CSC315 Programming Language Concepts

Professor Leon Tabak

Block 2

September 25, 2023 to October 18, 2023

Contents

1	Our meeting times and places	2
2	Attendance	3
3	Our Plan	3
4	Textbooks	3
5	Other Resources	4
6	Etiquette for the Classroom	5
7	Goals	6
8	Policies	7
8.1	Learning Disabilities/Accommodations	7
8.2	Academic Honesty	7
8.3	Dropping Courses	7
8.4	Mandatory Reporting	7
9	Grades	8
10	Calendar	9
11	Schedule	9

11.1	Week 0	9
11.1.1	Monday, 25 September 2023	9
11.1.2	Tuesday, 26 September 2023	10
11.1.3	Wednesday, 27 September 2023	11
11.1.4	Thursday, 28 September 2023	12
11.1.5	Friday, 29 September 2023	12
11.2	Week 1	13
11.2.1	Monday, 02 October 2023	13
11.2.2	Tuesday, 03 October 2023	14
11.2.3	Wednesday, 04 October 2023	14
11.2.4	Thursday, 05 October 2023	15
11.2.5	Friday, 06 October 2023	16
11.3	Week 2	16
11.3.1	Monday, 09 October 2023	16
11.3.2	Tuesday, 10 October 2023	17
11.3.3	Wednesday, 11 October 2023	18
11.3.4	Thursday, 12 October 2023	18
11.3.5	Friday, 13 October 2023	19
11.4	Week 3	19
11.4.1	Monday, 16 October 2023	19
11.4.2	Tuesday, 17 October 2023	20
11.4.3	Wednesday, 18 October 2023	20
11.4.4	Thursday, 19 October 2023	20
11.4.5	Friday, 20 October 2023	21

1 Our meeting times and places

- My office is in West Science 211.
- You may call me in my office at (319) 895-4294.
- You may send me electronic mail at ltabak@cornellcollege.edu.

- I will be in my office and available to meet with you in person Monday through Friday from 3:00 p.m. until 3:30 p.m.

	Where	When
Classroom/Laboratory	West Science Hall 200	9:00 a.m.—11:00 a.m.
Classroom/Laboratory	West Science Hall 200	1:00 p.m.—3:00 p.m.

2 Attendance

I will record your attendance each day. Cornell College now requires all instructors to record students' attendance. The law requires colleges to keep these records.

Software that we are using for our collaboration gives me an easier way of reviewing your work than I have had in the past. Because the software places timestamps on your work, I will be able to not only review the quantity and quality of your contributions, but also the frequency and regularity of your contributions.

3 Our Plan

- We will use the Scala programming language as an example of a modern programming language. We will learn Scala by writing programs with Scala.
- We will learn how to use regular expressions and context-free grammars to describe languages.
- We will take a brief look at how compilers work.
- We will survey the evolution of programming languages over the last 70 years.

We will use [Piazza](#) to share questions, answers, recommendations, and notes.

4 Textbooks

- [Effective Programming in Scala](#), Julien Richard-Foy and Noel Welsh, École Polytechnique Fédérale de Lausanne—a course from Coursera (free to audit)
- [Scala for the Impatient \(third edition\)](#), Cay S. Horstmann, Pearson Education 2023, ISBN 978-0-13-803365-1

Note: The author's website (see above) includes links to download source code for the examples in the book.

In the past, I taught this course with Robert Sebesta's book [Concepts of Programming Languages](#). **I am not asking you to get this book.** However, I am inviting you to look online at the book's table of contents. We will discuss many of the topics that you see listed there.

5 Other Resources

- [Writing READMEs](#)
- [Udacity's *A Beginner's Git and GitHub Tutorial*](#)
- ---
- [The Top Programming Languages \(IEEE Spectrum\)](#)
- [TIOBE index](#)
- [Top Programming Languages to Learn Today](#)
- ---
- [The Scala Programming Language \(project site\)](#)
- [Scala Programming Language \(Geeks for Geeks\)](#)
- [Learn Scala Programming \(tutorialspoint\)](#)
- [Scala solved examples](#)
- [Uses of Scala \(EDUCBA\)](#)
- [Why you must learn Scala programming to boost your career \(Emeritus\)](#)
- [Scala programming language: an introduction](#)
- [Scala tutorial \(W3Schools\)](#)
- [Scala style guide](#)
- [Scala interview questions](#)
- ---
- [ACM Turing Award winners](#)
- ---
- [Regular Expressions](#)

- [Context Free Grammars](#) (from Clemson University's Professor Wayne Goddard)
- [Introduction to Compilers](#) (from Loyola Marymount University's Professor Ray Toal)

6 Etiquette for the Classroom

Please show respect to your classmates, to your instructor, and to the seriousness of our enterprise by behaving courteously.

Courtesy is an important aspect of professionalism. Courteous and professional conduct will open for you opportunities to work with greater autonomy, to take on greater responsibility, to exercise greater creativity and leadership, and to solve more challenging problems and thereby serve your clients in more significant ways.

While we are studying online, many software engineers are working online. Let us develop the discipline and habits that will help us succeed in the work that will follow our studies at Cornell College.

How might we adapt rules for etiquette in a classroom and office for our online collaboration?

How can each of us be...

- a member of the team who delivers work in full and on time?
- a member of the team who delivers products of the highest quality?
- a member of the team to whom others turn for help?
- a member of the team whose critiques and suggestions others welcome?

Here are guidelines that I shared with my students in past years. I would like to hear your opinion: does this advice still apply in our new circumstance? Is there anything here that we should add or subtract?

- Please give your attention to whomever is speaking. You cannot view unrelated pages on the Web and be part of our class' discussion at the same time.
- You learn from your classmates. Be generous in offering help to classmates in the laboratory. Take interest in your classmates' work. Encourage them. Compliment them for work that is well done. Give them a good audience when they stand at the front of the room to present their work. Show these courtesies to all of your classmates.

- Please do not interrupt the class by late entries or early departures. If you anticipate a need to be absent from all or part of one of our meetings, please notify me in advance of your anticipated absence.
- You may listen to music while working in the laboratory so long as you are still able to hear your name when called and you do not disturb neighbors.
- Please refrain from bringing food or drink into the classroom or laboratory. We can make reasonable exceptions for eating that is not noisy and foods that do not have strong smells. Acceptable beverages and foods include water, tea, and granola bars. Bringing breakfast to class is not courteous. Please clean up crumbs and spills. Please dispose of empty containers and leftovers.
- Please dress as you might for an employer in the software engineering industry. This does not mean fancy dress—you do not need to buy new clothes. The dress in most workplaces is casual. Just be neat. Please keep your shoes on. Wearing hoods, hats, or sunglasses (except when there is a medical reason for shielding the eyes) that hide your face is not courteous.
- Imagine that you are seeking employment. How will you present yourself to your prospective employer? Imagine that you are now employed in a software engineering firm. How will you speak to your teammates, the head of your team, and your company's clients? Imagine that your grandmother has purchased the company for which you work. She has joined you in the company's conference room to hear and see you walk through the code that you have written for the company (her company). Are there some words that you will keep out of your vocabulary during this hour?

7 Goals

We will give special attention to three of Cornell College's **Educational Priorities and Outcomes**:

Knowledge You will learn how programming languages have evolved. You will learn more about what to look for in a new language, how to compare languages, and how to learn new languages. You will learn how to describe the grammars of languages. The course will introduce you to the parts of a compiler and algorithms that are at the heart of algorithms.

Communication You will present your work to your peers and you will learn from your peers. This practice will serve you well wherever you go.

Vocation In this project-oriented course you will have opportunities to create work that you can include in applications for internships, employment, and admission to programs of advanced study. You will learn gain familiarity with online courses that software engineers use to continue their education throughout their careers.

8 Policies

8.1 Learning Disabilities/Accommodations

Cornell College is committed to providing equal educational opportunities to all students. If you have a documented learning disability and will need any accommodation in this course, you *must* request the accommodation(s) from the instructor of the course no later than the third day of the term. Additional information about the policies and procedures for accommodation of learning disabilities is available on [Cornell College's Web site](#).

8.2 Academic Honesty

Please also familiarize yourself with the college's statement on [academic honesty](#)

8.3 Dropping Courses

Please also familiarize yourself with the college's [policies for dropping courses](#).

8.4 Mandatory Reporting

It is my goal that you feel supported and able to share information related to your life experiences during classroom discussions, in your written work, and in any one-on-one meetings with me. You should also know that all Cornell College faculty and staff are mandatory reporters. This means that I will keep information you share with me private to the greatest extent possible. However, I am required to share information regarding sexual assault, abuse, criminal behavior, or about a student who may be a danger to themselves or to others. If you wish to speak to someone confidentially who is not a mandatory reporter, you can [schedule an appointment with one of the counselors](#) in the Ebersole Health and Wellbeing Center or contact the College Chaplain, Rev. Melea White, at mwhite@cornellcollege.edu

9 Grades

You can earn all the points for “Daily Work” by arriving on time for each of our meetings, arriving prepared, and keeping your focus on the work before us. Of course, there might sometimes be good reasons for an absence (for example, illness)—in those cases, just notify me of your need to be absent. There is no competition for these points—I would like to award every student 20 points for daily work!

Experience presenting work to peers will be a central part of the course. Practice asking your teammates questions during their presentations, critiquing their decisions, and suggesting improvements to their code will also be an important part of your education during this term.

You will learn more if you draw upon the knowledge of classmates and generously share what you know with your classmates. Ask questions. Offer suggestions and help. Work together.

There will be three Graded Exercises. For each Graded Exercise, ...

- you will produce a lesson on a topic that is relevant to our course. Your instructor will suggest topics and invite you to propose others if you wish. You will create 4–6 slides and a 1–2 page handout. You will present your lesson to your instructor in a talk that lasts no more than 5 minutes.
- you will produce a program that illustrates the use of some feature (or features) of the Scala language
- you will answer 6–12 short-answer questions on an oral or written quiz

At the end of the course, you will share with your instructor and your classmates a programming project that you have created for your portfolio. Include documentation and tests. Be prepared to explain what you learned (and what your classmates might learn) from your project. Your instructor will suggest programming projects and show examples.

Effort counts. Each student will be starting from a different place. How far you go from your starting point will determine your score for projects. The quality of your presentation, as much as the quality and sophistication of your software, will determine your score. Earn points by explaining clearly what you have learned. Earn points by explaining what we can learn by studying your work.

Activity	Points
Daily work	20
Graded exercise 0 (Friday, 29 September 2023)	20
Graded exercise 1 (Friday, 06 October 2023)	20
Graded exercise 2 (Friday, 13 October 2023)	20
+ Portfolio contribution (Wednesday, 18 October 2023)	20
	100

10 Calendar

	Mon	Tue	Wed	Thu	Fri
Week 0	25	26	27	28	29
Week 1	02	03	04	05	06
Week 2	09	10	11	12	13
Week 3	16	17	18	19	20

11 Schedule

11.1 Week 0

11.1.1 Monday, 25 September 2023

Read: Begin viewing the lectures and answering the questions in the quizzes in the Week 1 section of the *Effective Programming in Scala* course on Coursera.

Read Chapters 1–2 (pages 1–37) of *Scala for the Impatient*.

Discuss:

- goals of course
- why Scala?
- organization and schedule
- how to get started with Scala

- how to declare values and variables
- common types
- operators and methods in Scala
- how to use the documentation
- conditional expressions
- blocks
- loops
- functions and arguments
- exceptions

Write: Which programming languages are most popular today?

Return to calendar.

11.1.1.2 Tuesday, 26 September 2023

Read: Read Chapters 3–4 (pages 39–60) of *Scala for the Impatient*.

Discuss:

- elements of programs
- types
- methods and parameters
- conditions
- evaluating definitions

-
- arrays
 - more about Scaladoc
 - interoperating with Java
 - maps (constructing, accessing, updating, iterating)

- the Option type
- tuples
- zipping

Write: What is a machine language? an assembly language? a high level language?

Return to calendar.

11.1.3 Wednesday, 27 September 2023

Our class will not meet this afternoon.

Read: Finish viewing the lectures and answering the questions in the quizzes in the Week 1 section of the *Effective Programming in Scala* course on Coursera.

Read Chapters 5–6 (pages 63–83) of *Scala for the Impatient*.

Discuss:

- domain modeling
- case classes
- sealed traits
- enumerations
- business logic

-
- classes
 - properties, getters, setters
 - private fields
 - constructors
 - singletons
 - companion objects
 - more about the apply method
 - enumerations

Write: What are some programming language paradigms?

Return to calendar.

11.1.4 Thursday, 28 September 2023

Read: Begin viewing the lectures and answering the questions in the quizzes in the Week 2 section of the *Effective Programming in Scala* course on Coursera.

Read Chapter 7 (pages 85–94) of *Scala for the Impatient*.

Discuss:

- first steps with lists
- functions
- more operations on lists
- introducing collections

-
- packages
 - nesting
 - visibility
 - imports and exports

Write: Tell us about the life and work of someone who won the Turing Award for advancing the state of the art for programming languages.

Return to calendar.

11.1.5 Friday, 29 September 2023

Read:

Discuss: Share the work you did for Graded Exercise 0.

Write: Graded Exercise 0 is due today.

Return to calendar.

11.2 Week 1

11.2.1 Monday, 02 October 2023

Read: Read Chapter 8 (pages 97–114) of *Scala for the Impatient*.

Discuss:

- querying collections
- transforming collections
- sequences and maps
- Option
- collections extra

-
- extending a class
 - overriding methods
 - determining the class to which an object belongs
 - testing the equality of two objects
 - primary and auxillary constructors and the superclass' constructor
 - overriding methods and fields
 - Scala's inheritance hierarchy

Write: Tell us a little about the origins of a handful of programming languages.

Return to calendar.

11.2.2 Tuesday, 03 October 2023

Read: Finish viewing the lectures and answering the questions in the quizzes in the Week 2 section of the *Effective Programming in Scala* course on Coursera.

Read Chapter 10 (pages 129–147) of *Scala for the Impatient*.

Discuss:

- introducing loops
 - more loop examples
 - tail recursion
 - “for” syntax
-

- why the language does not allow multiple inheritance
- Scala traits as a generalization of Java’s interfaces
- concrete and abstract methods
- how to use several traits together

Write: How are finite state machines, regular languages, and regular expressions related?

Return to calendar.

11.2.3 Wednesday, 04 October 2023

Read: Begin viewing the lectures and answering the questions in the quizzes in the Week 3 section of the *Effective Programming in Scala* course on Coursera.

Read Chapter 11 (pages 149–166) of *Scala for the Impatient*.

Discuss:

- organize code
- build tools
- introduction to the sbt build tool

- sbt, keys, and scopes
 - program entry point
-

- rules for selecting names of objects
- how to create our own operators
- precedence and associativity of operators
- infix (binary) operators, unary operators, assignment operators
- the apply, update, unapply, and unapplySeq methods

Write: What kind of machine can we use to recognize a context-free language?

Return to calendar.

11.2.4 Thursday, 05 October 2023

Read: Finish viewing the lectures and answering the questions in the quizzes in the Week 3 section of the *Effective Programming in Scala* course on Coursera.

Read Chapter 12 (pages 169–181) of *Scala for the Impatient*.

Discuss:

- encapsulation
 - extending and refining classes
 - case classes vs. simple classes
 - opaque types
 - extension methods
-

- functions as values (for example, a function can return a function)
- parameters that are functions (we can pass a function to a function)
- introduction to higher order functions with examples
- closures
- currying

Write: What is a Turing complete language?

Return to calendar.

11.2.5 Friday, 06 October 2023

Read:

Discuss: Share the work you did for Graded Exercise 0.

Write: Graded Exercise 1 is due today.

Return to calendar.

11.3 Week 2

11.3.1 Monday, 09 October 2023

The afternoon meeting will end at 1:30 p.m. (Professor Tabak has a medical appointment today.)

Read: Begin viewing the lectures and answering the questions in the quizzes in the Week 4 section of the *Effective Programming in Scala* course on Coursera.

Read Chapter 13 (pages 183–207) of *Scala for the Impatient*.

Discuss:

- reasoning about code
- refactoring-proof programs
- a case for side-effects
- mutable objects

-
- most important traits in Scala’s collections hierarchy
 - sequences, lists, sets

- adding and removing elements
- a catalogue of important methods for working with collections (for example, methods for transforming and filtering)
- lazy lists

Write: Is SQL a programming language?

Return to calendar.

11.3.2 Tuesday, 10 October 2023

Read: Read Chapter 14 (pages 209–228) of *Scala for the Impatient*.

Discuss:

- unit testing
 - property-based testing
 - mocking
 - integration testing
 - testing the tests
 - debugging programs
-
- several uses of pattern matching
 - improvement of the switch statement of C and Java
 - guards—a Boolean expression
 - variables in patterns
 - matching on types
 - looking for patterns in the contents of arrays, lists, tuples
 - matching with case classes
 - matching with sealed classes

Write: What is a “build tool”?

Return to calendar.

11.3.3 Wednesday, 11 October 2023

Read: Read Chapter 16 (pages 245–260) of *Scala for the Impatient*.

Discuss:

- running tasks in the future (asynchronous methods)
- waiting for results
- the Try class
- callback functions
- Promises

Write: What are the principal parts of a compiler?

Return to calendar.

11.3.4 Thursday, 12 October 2023

Read: Finish viewing the lectures and answering the questions in the quizzes in the Week 4 section of the *Effective Programming in Scala* course on Coursera.

Read Chapter 17 (pages 263–275) of *Scala for the Impatient*.

Discuss:

- generic classes—classes parameterized by type
- specifying restrictions on the types allowed
- defining relations between

Write: Use examples to help us understand the difference between parse trees and expression trees.

Return to calendar.

11.3.5 Friday, 13 October 2023

Read: Together, and as time allows, we will view lectures and answer questions in remaining parts of the *Effective Programming in Scala* course on Coursera.

- type-directed programming
- error handling
- concurrent programming

-
- contextual abstractions (Chapter 19 in *Scala for the Impatient*)
 - type-level programming (Chapter 20 in *Scala for the Impatient*)

Discuss: Share the work you did for Graded Exercise 2.

Write: Graded Exercise 2 is due today.

Return to calendar.

11.4 Week 3

11.4.1 Monday, 16 October 2023

Professor Tabak will be absent today. He will be visiting Morningside University in Sioux City, Iowa.

Professor Tabak will invite one of his colleagues to work with you during his time away from campus. Your instructor might choose to adjust the meeting times.

Read:

Discuss: Rehearse your final presentations. Review and suggest improvements to the portfolio contributions that your classmates are writing.

Write: There is no written work due today.

Return to calendar.

11.4.2 Tuesday, 17 October 2023

Professor Tabak will be absent today. He will be visiting Morningside University in Sioux City, Iowa.

Professor Tabak will invite one of his colleagues to work with you during his time away from campus. Your instructor might choose to adjust the meeting times.

Read:

Discuss: Rehearse your final presentations. Review and suggest improvements to the portfolio contributions that your classmates are writing.

Write: There is no written work due today.

Return to calendar.

11.4.3 Wednesday, 18 October 2023

Read:

Discuss: Share our portfolio contributions.

Write: Your portfolio contribution (your final project) is due today.

Return to calendar.

11.4.4 Thursday, 19 October 2023

Block Break We will not meet today.

There is no work due today.

Return to calendar.

11.4.5 Friday, 20 October 2023

Block Break We will not meet today.

There is no work due today.

Return to calendar.