Fall 2023

COMPUTER SCIENCE 126

Fall 2023

https://www.princeton.edu/~cos126

computer-generated music “Stories from the Silk Road”
I will be recording class meetings to make them available to students in Covid isolation.

Because of privacy, compliance, and legal considerations, you may not record or redistribute recordings of this class.
Questions welcome at any time

raise your hand and ask

ask on Ed
(use ❤ to upvote)
R O B E R T  S E D G E W I C K
KEVIN WAYNE

[Sedgewick and Flajolet] are not only worldwide leaders of the field, they also are masters of exposition. I am sure that every serious computer scientist will find this book rewarding in many ways.

—From the Foreword by Donald E. Knuth

Despite growing interest, basic information on methods and models for mathematically analyzing algorithms has rarely been directly accessible to practitioners, researchers, or students. An Introduction to the Analysis of Algorithms, Second Edition, organizes and presents that knowledge, fully introducing primary techniques and results in the field.

Robert Sedgewick and the late Philippe Flajolet have drawn from both classical mathematics and computer science, integrating discrete mathematics, elementary real analysis, combinatorics, algorithms, and data structures. They emphasize the mathematics needed to support scientific studies that can serve as the basis for predicting algorithm performance and for comparing different algorithms on the basis of performance.

Techniques covered in the first half of the book include recurrences, generating functions, asymptotics, and analytic combinatorics. Structures studied in the second half of the book include permutations, trees, strings, tries, and mappings. Numerous examples are included throughout to illustrate applications to the analysis of algorithms that are playing a critical role in the evolution of our modern computational infrastructure.

Improvements and additions in this new edition include

- Upgraded figures and code
- An all-new chapter introducing analytic combinatorics
- Simplified derivations via analytic combinatorics throughout

The book's thorough, self-contained coverage will help readers appreciate the field's challenges, prepare them for advanced results—covered in their monograph Analytic Combinatorics and in Donald Knuth's Art of Computer Programming books—and provide the background they need to keep abreast of new research.

ROBERT SEDGEWICK is the William O. Baker Professor of Computer Science at Princeton University, where was founding chair of the computer science department and has been a member of the faculty since 1985. He is a Director of Adobe Systems and has served on the research staffs at Xerox PARC, IDA, and INRIA. He is the coauthor of the landmark introductory book, Algorithms, Fourth Edition. Professor Sedgewick earned his Ph.D from Stanford University under Donald E. Knuth.

The late PHILIPPE FLAJOLET was a Senior Research Director at INRIA, Rocquencourt, where he created and led the ALGO research group. He is celebrated for having opened new lines of research in the analysis of algorithms; having systematized and developed powerful new methods in the field of analytic combinatorics; having solved numerous difficult, open problems; and having lectured on the analysis of algorithms all over the world. Dr. Flajolet was a member of the French Academy of Sciences.
Goal 1. Read, write, and reason about computer programs.
Goal 2. Apply concepts to science, engineering, and beyond.
Goal 3. Build awareness of substantial intellectual underpinnings.
Goal 4. Demystify computer systems.

<table>
<thead>
<tr>
<th>topic</th>
<th>examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>elements of programming</td>
<td>variables, loops, conditionals, arrays, I/O</td>
</tr>
<tr>
<td>functions</td>
<td>user-defined functions, modularity, recursion</td>
</tr>
<tr>
<td>object-oriented programming</td>
<td>user-defined data types, encapsulation, immutability</td>
</tr>
<tr>
<td>algorithms</td>
<td>sorting, binary search, stacks, queues, BSTs</td>
</tr>
<tr>
<td>theory of computing</td>
<td>universality, computability, intractability</td>
</tr>
<tr>
<td>design of computers</td>
<td>machine language, boolean logic, circuits</td>
</tr>
</tbody>
</table>
COS 126, Spring 2023

- digital revolution
- course mechanics
- course resources

https://introcs.cs.princeton.edu
Key idea. “Everything” can be encoded as a sequence of **bits** (0s and 1s).
• Key idea. “Everything” can be encoded as a sequence of bits (0s and 1s).
  – Numbers and text.
  – Pictures, songs, and movies.
  – Your DNA.
  – 3D objects.
  – Computer programs.
  – ...

• Innovation 1. You can program computers to process bits.
• Innovation 2. Devices use the Internet to send and receive bits.
• Innovation 3. Bits $\leftarrow\rightarrow$ (sensors, displays, actuators, etc.) $\leftarrow\rightarrow$ the world.
Transforming the way we work and live...

Baron, A better pencil. When WordStar was king, making PCs altogether, selling the rights to its ThinkPad laptop to the Chinese manufacturer Lenovo. Perhaps one reason why IBM failed to foresee that the PC was destined for a place on every home or office desk in America was the company's commitment to its own typewriter line, together with a conviction that writers didn't want—or need—to change technologies. What most writers do at their desks is create or copy documents, and so far as IBM was concerned, between the pencil and the typewriter, the writers' market was all sewn up. People used computers for numbers, and numbers meant bigger and bigger mainframes, not trim, under-powered desk units. The entertainment potential of the personal computer, particularly in the area of electronic games, boosted its popularity, but factoring in such post-1982 developments as email, instant messaging, and the web, what most people now do with their PCs is process words, and while a lot of word processing still involves copying or manipulating text created by others, more and more writers are taking advantage of the new digital genres to create and publish texts of their own.

Mainframes and dedicated word processors showed those writers and computer manufacturers who cared to look a glimpse of what their future would be. Two things happened that paved the way for the personal computer to become the word processor of choice: PCs became affordable, and they became not just user-friendly, but writer-friendly as well.

The Machine of the Year

Image 37. Time Magazine named the computer the “machine of the year” for 1982 [image courtesy of Time Magazine; used by permission]
Transforming science and engineering...
The digital revolution has only just begun

In 2020. 50 billion+ smart connected devices, all developed to collect, analyze, and share data.
What will the future be?

‣ Language Processing

‣ Computer Vision/Image Processing

‣ Entertainment and Games

‣ Robotics

‣ Healthcare

‣ Communication

‣ Finance

‣ ...
The digital revolution has only just begun

You are already a consumer (and producer and product). Now, become a creator!

[99% of politicians agree]
[computer science education—a rare nonpartisan issue!]

Programming:
Break a big problem into smaller pieces and identify the right steps to solve it.

Computer Science is a basic skill
Use your new superpower responsibly!

Consider The Therac-25 Medical Linear Accelerator
- Designed to destroy tumors with minimal damage to surrounding tissue
- Relied on software for safe operation
- Six known massive radiation overdoses 1985-1987
- Responsible for several deaths

An Investigation of the Therac-25 Accidents
Nancy G. Leveson*
Clark S. Turner†

Abstract
Risk in any complex technology is unavoidable. However, important lessons can be learned from accidents which can be used to design procedures for reducing risk in the future. Although descriptions of the Therac-25 medical electron accelerator accidents have been published previously, they are incomplete and often misleading. This paper contains a detailed account of these accidents, along with some lessons that can be learned from them in terms of system engineering, software engineering, and government regulation of safety-critical systems involving software.
Use your new superpower ethically!

The Poison on Facebook and Twitter Is Still Spreading

Social platforms have a responsibility to address misinformation as a systemic problem, instead of reacting to each case.

Oct. 19, 2020

Hospital Algorithms Are Biased Against Black Patients, New Research Shows

RESEARCH SAYS FACEBOOK'S AD ALGORITHM PERPETUATES GENDER BIAS

A University of Southern California study provides more evidence that the company's ad targeting illegally discriminates.

April 2, 2020, 5:44 a.m.

Twitter algorithm prefers slimmer, younger, light-skinned faces

An Algorithm That Grants Freedom, or Takes It Away

Across the United States and Europe, software is making probation decisions and predicting whether teens will commit crime. Opponents want more human oversight.
Understanding Computer Science can help get you elected!
COS 126, Spring 2023

- digital revolution
- course mechanics
- course resources

https://introcs.cs.princeton.edu
Course website

https://www.princeton.edu/~cos126

COS 126 Fall'23  Syllabus  Schedule  Assignments  Project  Help  Resources  Exams  People

Syllabus

Course Description
This course is an introduction to computer science in the context of scientific, engineering, and commercial applications. The goal of the course is to teach basic principles and practical issues, while at the same time preparing students to use computers effectively for applications in computer science, physics, biology, chemistry, engineering, and other disciplines. Topics include: programming in Java; hardware and software systems; algorithms and data structures; fundamental principles of computation; and scientific computing, including simulation, optimization, and data analysis.

Textbook
R. Sedgewick and K. Wayne, Computer Science: An Interdisciplinary Approach, Addison–Wesley Professional, 2016. ISBN 978-0134076423. We will be referencing this text all semester. The lectures are based on its contents. The hardcopy version of the textbook is available from Labyrinth Books.

Course Organization
This course includes lectures, class meetings, precepts, programming assignments, a final project, and two (2) two-part exams, as outlined below.

Lectures. Lectures are provided as studio-produced videos. You are expected to watch lecture before precept so that you are well-prepared. See the Schedule page.

Class Meetings. Class meetings are scheduled Thursdays throughout the semester at 3:00pm. These meetings are used for exams, practice exams, and programming assignment overviews.

Precepts. Precepts meet twice per week (either Monday/Wednesday or Tuesday/Thursday). Working within a small group, your preceptor will review key learning objectives for the week's lectures and precept exercises. The precept will involve question and answer, group discussion, etc. Each precept involves working with a partner on interactive exercises and problem sets. Precept attendance counts for 5% of your overall grade.
“Flipped lecture” format

Lecture videos (required). Watch studio-produced videos before precept.

Precepts

**Active learning.** Discussion, problem solving, pair programming, …

- Bridge between lecture/reading and the assignments
- 50-minute precepts.
- 80-minute precepts.
- Ed Lessons
  - Announcements
  - Download the project zip for precept
  - Precept exercises (for the most part)
  - Review before and/or after precept, as needed
- Can I reuse precept code on assignments / programming exams?
  - YES!
- Attendance required!
- If you cannot attend a precept, attend another
  - Email preceptor and sign in
- If you are ill
  - Makeup precept TTh 7:30–8:20pm (Zoom)

---

**Introduction**
At the end of this precept, you should be able to:

1. Understand how to compile and run code from your terminal, and then write code to print a message to a terminal (Activity 1 - `HelloWorld.java`).
2. Write code to access command-line arguments (Activity 2 - `NameAge.java`).
3. Write code to parse the numeric value from a `String` (Activity 3 - `NextYear.java`).
4. Understand how to use TigerFile to submit assignments.

In general, if you are ever having trouble with the weekly assignments, make sure you have gone through all the precept content and understand most of it. Some precepts may also have extra practice assignments that will help you practice the concepts covered in precept.

This precept is based on:

1. Video lecture: Introduction to Java 1
2. Reading: 1.1
NEW! For students who are interested in biology and medicine

- Meets Mon/Wed 11:00am.
- Same lectures and programming assignments.
- Examples will be presented from relevant biomedical applications, scenarios and datasets.
- Familiarity with or curiosity about fundamental concepts from biology is recommended but not required.

Preceptor - Prof. Yuri Pritykin - https://pritykinlab.github.io
Raspberry Pi Precept (P16A)

For students with some Java experience interested in learning topics beyond the scope of the course

- Meets Tue/Thu 11:00am.
- Same lectures and programming assignments.
- About half precept time will explore the use of Java to program on the Raspberry Pi platform.
- If you have some Java experience and have a maker spirit, then you may find this precept appealing
- Third semester
  - Fall’23 - led by Nick Sudarsky’23 - IW and Senior Thesis formed the basis for this precept!
Class meetings

**Purpose.** Assignment overviews, debugging tutorial, exams, …
- We will use McCosh 50 and other rooms for exams

**Participation.** You are encouraged and expected to participate.
- Raise your hand and ask a question.
- Ask a question (anonymously) in Ed Discussion.

*carpe diem!*
Programming assignments and Final Project

... are an essential part of the experience in learning CS.

- Illustrate a programming or general computer science concept.
- Highlight the role of computation in an important application.
- You solve the problem from scratch, on your own computer!
- Ten (10) programming assignments plus a final Dean’s Date project

Partners allowed and encouraged!
Recursive Graphics

Sierpinski Triangles

"Lorax Trees" by Jonathan Zhang (Fall 2014)

"Piet Mondrian Rectangles" by Laura Herman (Fall 2015)
Every semester, Princeton University’s COS 126 invites students to use their newly acquired programming skills to create some amazing pieces of "recursive" art!

Here is what the Fall 2022 class has come up with!

警告：视频包含闪烁的灯光/图像！
Guitar Hero

Simulate plucking a guitar string using the Karplus–Strong algorithm.

[ performed by Kevin Wayne in 2013 on a MacBook Pro ]
N-Body Simulation

Simulate the motion of \( n \) particles, subject to Newton’s laws of gravity.

- our Solar System (5 bodies)
- two colliding galaxies (30M bodies)
Submitting assignments 🐾

**When?** Due Sundays at 11:59pm ET.

**How?** Upload files via TigerFile.

**Where is the TigerFile link for the current assignment?** On the specific assignment page - *Submit to TigerFile*

**What does Download Project mean?** Stay tuned!

---

**0. Hello, World**

*Download Project* | *Submit to TigerFile*

The purpose of this assignment is to introduce you to programming in Java and familiarize you with the mechanics of preparing and submitting assignment solutions. You will learn to use IntelliJ editor for writing, compiling, and executing Java programs, and TigerFile for submitting your work electronically.

**Requirements**
- Install the COS 126 Java programming environment, called IntelliJ, on your computer. Follow the instructions to download and install IntelliJ.
- Write your code in IntelliJ and submit it electronically through TigerFile.
Preliminary feedback? Click “Check Submitted Files” button.
Resubmit or unsubmit? Yes.
Submitting assignments

DEMO
Submitting assignments

**Preliminary feedback?** Click “Check Submitted Files” button.

**Resubmit or unsubmit?** Yes.

**All done?** Submit acknowledgments.txt file.

**Passing test cases?** This is very good indicator for correctness; however, style, readme.txt, etc. also counts!

**Submitting late?** 10% penalty per (partial) day.

---

we waive first 4 late days automatically (after that, only with recommendation of residential college dean)
What does Download Project mean? Starting with the Loops assignment, you will download the project zip from TigerFile.

When are assignments published? The assignments are available now. However, we reserve the right to change them. So they project zip and official project zip will be published on the Monday morning, before the Sunday due date.
Two Programming exams

**Goal.** Quickly write a short program.

- First programming exam is primarily for practice (2%).
- Second programming exam is for demonstrating your new skills (13%).
- On your laptop
- We provide some previous programming exams. Open course materials!
  - Caveat - not every semester is identical!

**Problem.** Develop a Java program that can draw one of several game boards, shown here:
Two Written exams

Goal. Demonstrate your understanding of programming and non-programming concepts.

- Format:
  - Matching.
  - Short answer.
  - Fill-in-the-blank.
  - Multiple choice.
- On paper!
- WE1 - 15%
- WE2 - 15%

- We provide some previous written exams. Caveat - not every semester is identical
  - WE 1 - single-side of one 8.5x11 sheet - in your own handwriting
  - WE 2 - two-sides of one 8.5x11 sheet - in your own handwriting

2. Java basics. (8 points)

Suppose that the variables \( a \), \( b \), \( c \) are of type \( \text{int} \).

(a) Complete the following code fragment so that, after the last statement, the variable \( \text{max} \) contains the largest of the three values.

*In each blank, write one of \( a \), \( b \), \( c \), or \( \text{max} \). You may use each variable name once, more than once, or not at all.*

\[
\begin{align*}
\text{int max} & \quad = \quad a; \\
\text{if (_____ > _____) _____} & \quad = \quad _____; \\
\text{if (_____ > _____) _____} & \quad = \quad _____; 
\end{align*}
\]
Executive summary.

- Do discuss concepts with others.
- Do partner with a classmate (when permitted).
- Do acknowledge any collaboration with others.
- Do not copy code from anywhere or anyone.
- READ THE COURSE SYLLABUS.
- COLLABORATION POLICY QUIZ.
- ASK QUESTIONS.

<table>
<thead>
<tr>
<th></th>
<th>your partner</th>
<th>course staff</th>
<th>COS 126 alums</th>
<th>classmates</th>
<th>Al Chatbots</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>discuss concepts with ...</strong></td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td><strong>acknowledge collaboration with ...</strong></td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td><strong>show your code/solutions to ...</strong></td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td><strong>view any code/solutions of ...</strong></td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td><strong>copy any code/solutions from ...</strong></td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
</tbody>
</table>
Plagiarism detection.

- We use sophisticated tools to find plagiarism.
- Obfuscating copied code does not work.
- Cite your sources in the acknowledgments.txt file!

RRR warning: Plagiarizing code is treated the same as plagiarizing prose (but is much easier to catch).
Grading  

Programming assignments (40%). Assigned weekly.

Final project (10%). A larger and more challenging programming project.

Exams (45%).
- Two written exams (15% each).
- Two programming exams (2% and 13%).

Participation (5%). Attend and participate in precepts.

Course grades. Uncurved.

<table>
<thead>
<tr>
<th>grade</th>
<th>percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>93.0</td>
</tr>
<tr>
<td>A−</td>
<td>90.0</td>
</tr>
<tr>
<td>B+</td>
<td>87.0</td>
</tr>
</tbody>
</table>

Typical grade distribution:

- A
- B
- C
- D
- F

During class meetings (mark your calendars)
Icebreaker

Activity. Meet a classmate (that you don’t already know!) and discuss the following:

- What you hope to get out of COS 126.
- What you’re most excited about academically.
- What you’re passionate about non-academically.
- Then, keep chatting!
COS 126, SPRING 2023

- digital revolution
- course mechanics
- course resources

https://introcs.cs.princeton.edu
Resources (programming environment)

Recommended IDE. Custom IntelliJ 2023.2.1 environment.

- Embedded Bash terminal.
- Autoformat, autoimport, autocomplete, ....
- Continuous code inspection; integrated Checkstyle and SpotBugs.
- ...

upgrade to our Fall 2023 version
(see lab TAs for troubleshooting)
Resources (web)

Course website.
- Syllabus and course policies.
- Lecture videos and slides.
- Precept lessons.
- Programming assignments.

Booksites.
- Download code from book.
- Brief summary of content.
- For use while online.
- NOT the textbook.
- NOT the course website.

See “Resources” tab on course website for additional information!
Help (people)

Ed Discussion forum.
- Quick questions.
- Mark post private only when necessary.
- Please use Ed, not email.
- NEVER POST SOLUTION CODE HERE.

Intro COS Lab.
- Staffed by undergraduates happy to help you.
- For help with debugging code.
- introlab.cs.princeton.edu

Office hours.
- Longer discussions. protip: attend!
- See course website → Schedule.

McGraw.
- Peer tutoring.
- Consultations.
- …
- mcgraw.princeton.edu/undergraduates

See “Help” tab on course website for additional information!
## Resources (ed tech)

<table>
<thead>
<tr>
<th>Platform</th>
<th>What</th>
</tr>
</thead>
</table>
| ![Ed](image) | **Ed**  
Discussion forum, precept exercises    |
| ![IntelliJ](image) | **IntelliJ**  
Java IDE                      |
| ![TigerFile](image) | **TigerFile**  
Assignment submissions            |
| ![codePost](image) | **codePost**  
Assignment feedback               |
| ![Gradescope](image) | **Gradescope**  
Exam feedback                     |
| ![Canvas](image) | **Canvas**  
Check grades                      |
| ![Zoom](image) | **Zoom**  
Hopefully, not needed!
(except for makeup precept)        |

Also use for communication with course staff.
A typical week (but not this one!)

<table>
<thead>
<tr>
<th>Sun</th>
<th>Mon</th>
<th>Tue</th>
<th>Wed</th>
<th>Thu</th>
<th>Fri</th>
<th>Sat</th>
</tr>
</thead>
<tbody>
<tr>
<td>35</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>11</td>
<td>Precept</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>Video</td>
<td>Precept</td>
<td>Video</td>
<td>Precept</td>
<td>15</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>Assignment Released</td>
<td>Class Meeting</td>
<td>Assignment Feedback</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Assignment</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Support lecture videos & reading/prepare for assignment

YOU ARE HERE

Assignment overview; exams

Read this

Content based on week's material

Introduce new material
If I don’t understand a TigerFile test my code is failing, what should I do?

A. Attend office hours.
B. Get help from a lab TA.
C. Post a question on Ed Discussion.
D. Email my preceptor.
E. Drop the course.
If I don’t understand a key programming concept, what should I do?

A. Attend office hours.
B. Get help from a lab TA.
C. Post a question on Ed Discussion.
D. Email my preceptor.
E. Copy a classmate’s solution.
Q&A

raise your hand and ask

ask on Ed
(use ❤️ to upvote)