

CS21: INTRODUCTION TO COMPUTER SCIENCE

Prof. Mathieson

Fall 2018

Swarthmore College

Welcome to CS21!

TODO:

- **Registered**: sit at a computer and sign the attendance sheet by the end of class
- **Waitlist**: find a seat not at a computer and sign the waitlist sheet from Lauri
- **Everyone**: pick up Handout 1

Outline Sept 3:

- Staff introductions
- Algorithm example / what is CS?
- Log in to lab machines & Lab 0 info
- **Wednesday:**
 - Student introductions
 - Syllabus highlights
 - Areas of computer science
 - Begin: Python programming language
- Notes: this course has NO prerequisites. If you have extensive CS experience let me know.

Why should I take CS21?

Why should I take CS21?

~~Computer Science is about computers~~

Why should I take CS21?

~~Computer Science is about computers~~

“When human beings acquired language, we learned not just how to listen but how to speak. When we gained literacy, we learned not just how to read but how to write. And as we move into an increasingly digital reality, we must learn not just how to use programs but how to make them.”

-Douglas Rushkoff

Staff Introductions

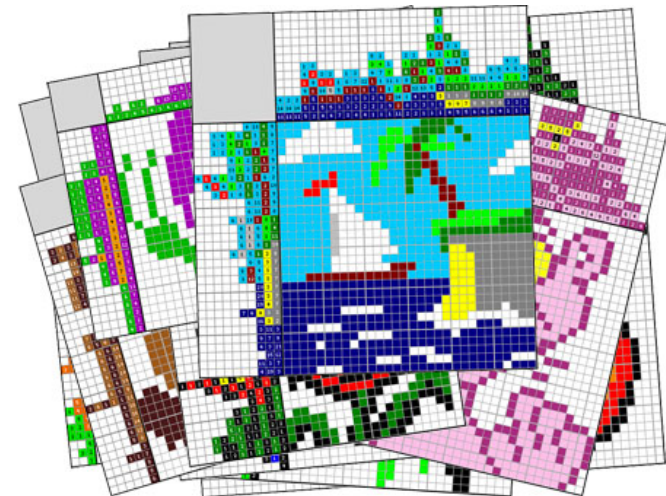
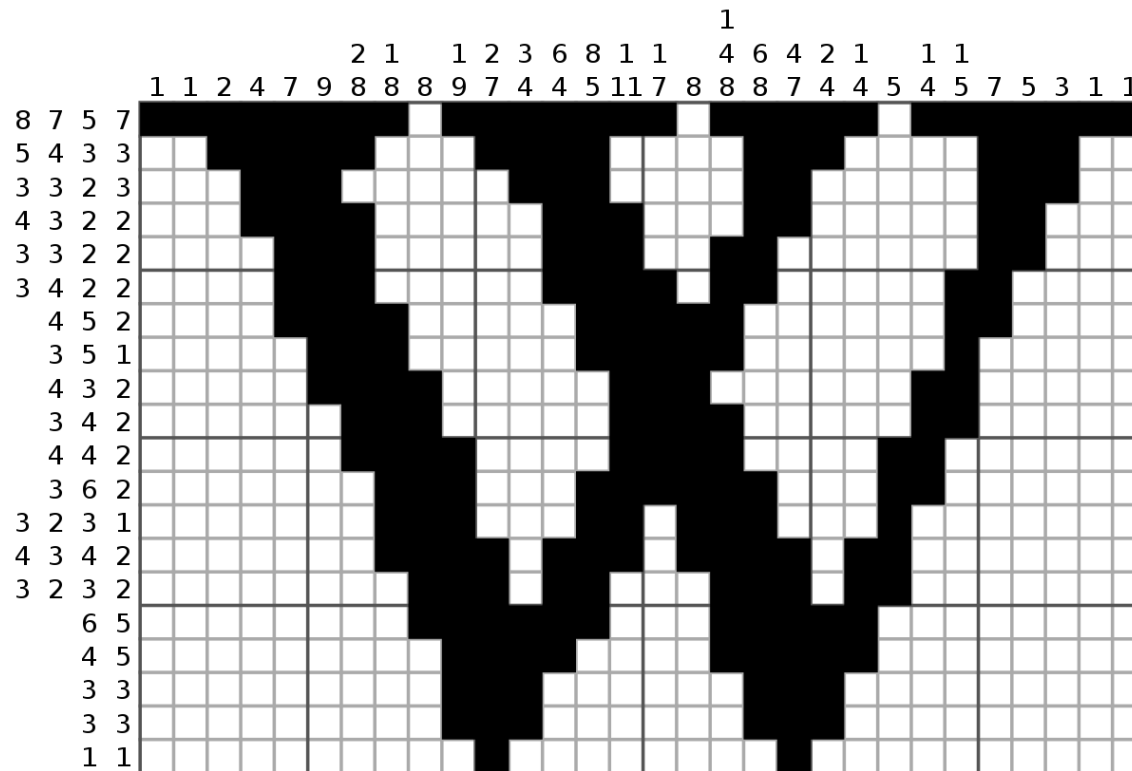
Course Staff

- Instructors
 - Section 1, Lab C&D: Aline Normoyle
 - **Section 2&3, Lab A: Sara Mathieson**
 - Lab B: Scout Sinclair
- Academic Support Coordinator
 - **Lauri Courtenay**
- Ninjas for Sections 2 & 3
 - **Kendre Thomas** (Section 2)
 - **Rohan Hejmadi** (Section 2)
 - **Ayaka Yorihiro** (Section 3)
 - **Maleyah Peterson** (Section 3)
- Graders
 - **TBA**

Lauri will introduce the ninja program

Algorithm example

Example of an algorithm: nonograms



Handout example

							2		
					2	2	1		
			1	5	1	3	1	5	1
		1							
		3							
	2	2							
	2	2							
		5							
1	1	1							
	1	3							

Handout example

							2		
					2	2	1		
			1	5	1	3	1	5	1
		1							
		3							
	2	2							
	2	2							
		5							
1	1	1							
	1	3							

Handout example

							2		
					2	2	1		
			1	5	1	3	1	5	1
		1							
		3							
	2	2							
	2	2							
		5							
1	1	1							
	1	3							

Handout example

							2		
					2	2	1		
			1	5	1	3	1	5	1
		1							
		3							
	2	2							
	2	2							
		5							
1	1	1							
	1	3							

Handout example

							2		
					2	2	1		
			1	5	1	3	1	5	1
		1							
		3							
	2	2							
	2	2							
		5							
1	1	1							
	1	3							

We are certain
these are filled in

With a partner...

- **Mini-bio** (name, year, where from, summer, major...) remember this for Wed - you'll introduce your partner then!
- Start the handout questions (more important to get to **question 4** than finish the puzzle)
- For question 4, you can **discuss ideas** instead of writing down everything if you're short on time

Example Algorithm for row checking

- Start at the beginning of the row

Example Algorithm for row checking

- Start at the beginning of the row
- Move to the right as long as the boxes are **E**'s

Example Algorithm for row checking

- Start at the beginning of the row
- Move to the right as long as the boxes are **E**'s
- When we find an **F**, start counting how many consecutive **F**'s there are

Example Algorithm for row checking

- Start at the beginning of the row
- Move to the right as long as the boxes are **E**'s
- When we find an **F**, start counting how many consecutive **F**'s there are
- Once we get to another **E**, store or save how many consecutive **F**'s we saw

Example Algorithm for row checking

- Start at the beginning of the row
- Move to the right as long as the boxes are **E**'s
- When we find an **F**, start counting how many consecutive **F**'s there are
- Once we get to another **E**, store or save how many consecutive **F**'s we saw
- Keep repeated the above 3 steps building up a list of counts of consecutive **F**'s

Example Algorithm for row checking

- Start at the beginning of the row
- Move to the right as long as the boxes are **E**'s
- When we find an **F**, start counting how many consecutive **F**'s there are
- Once we get to another **E**, store or save how many consecutive **F**'s we saw
- Keep repeated the above 3 steps building up a list of counts of consecutive **F**'s
- When we reach the end of the row and have a list of counts, compare it with the row numbers

Example Algorithm for row checking

- Start at the beginning of the row
- Move to the right as long as the boxes are **E**'s
- When we find an **F**, start counting how many consecutive **F**'s there are
- Once we get to another **E**, store or save how many consecutive **F**'s we saw
- Keep repeated the above 3 steps building up a list of counts of consecutive **F**'s
- When we reach the end of the row and have a list of counts, compare it with the row numbers
- If the lists of numbers are the same, the solution is valid
- If not, the solution is invalid

In these algorithms, can you find...

- Input/output

In these algorithms, can you find...

- Input/output
 - Input: target row numbers, candidate solution
 - Output: yes/no

In these algorithms, can you find...

- Input/output
 - Input: target row numbers, candidate solution
 - Output: yes/no
- Branching

In these algorithms, can you find...

- Input/output

- Input: target row numbers, candidate solution
- Output: yes/no

- Branching

- Operation changes based on **E** vs. **F**

In these algorithms, can you find...

- Input/output
 - Input: target row numbers, candidate solution
 - Output: yes/no
- Branching
 - Operation changes based on **E** vs. **F**
- Looping

In these algorithms, can you find...

- Input/output

- Input: target row numbers, candidate solution
- Output: yes/no

- Branching

- Operation changes based on **E** vs. **F**

- Looping

- Repeated the steps to count each block of **F**'s

In these algorithms, can you find...

- Input/output
 - Input: target row numbers, candidate solution
 - Output: yes/no
- Branching
 - Operation changes based on **E** vs. **F**
- Looping
 - Repeated the steps to count each block of **F**'s
- Data structures

In these algorithms, can you find...

- Input/output

- Input: target row numbers, candidate solution
- Output: yes/no

- Branching

- Operation changes based on **E** vs. **F**

- Looping


- Repeated the steps to count each block of **F**'s

- Data structures

- Way to store the input as well as the counts

Log in to lab machines

Steps for lab machines today

- Find your account sheet with username and password
- Click on the terminal icon 
- Change your password by typing “passwd”

```
cilantro[~]$  
cilantro[~]$ passwd  
Enter login(LDAP) password:  
New CompSci password:  
Retype new CompSci password:  
LDAP password information changed for smathieson  
passwd: password updated successfully  
cilantro[~]$
```

- Bookmark course webpage (click on the star in Firefox)

<https://www.cs.swarthmore.edu/courses/CS21Labs/f18/index.php?section=2>

- Sign and return user agreement by the end of class

Lab 0 Notes

65/102 students have already posted on Piazza, thank you!

- **Lab attendance is mandatory!** Lab 0 is Tuesday (tomorrow) and Wednesday, due Saturday Sept 8
- As part of Lab 0, make sure to **visit my office (249)** by Friday at 5pm

Syllabus highlights

Syllabus highlights and notes

- Notes will be be posted *after* class on the webpage

Syllabus highlights and notes

- Notes will be be posted *after* class on the webpage
- Textbook – free and online 😊

Syllabus highlights and notes

- Notes will be be posted *after* class on the webpage
- Textbook – free and online 😊
- Make sure to attend lab! (mandatory, attendance taken)

Syllabus highlights and notes

- Notes will be be posted *after* class on the webpage
- Textbook – free and online 😊
- Make sure to attend lab! (mandatory, attendance taken)
- Labs are (almost always) due Saturdays before midnight

Syllabus highlights and notes

- Notes will be be posted *after* class on the webpage
- Textbook – free and online 😊
- Make sure to attend lab! (mandatory, attendance taken)
- Labs are (almost always) due Saturdays before midnight
- **Extensions**
 - Everyone gets 2 “late days” to be used for any reason
 - Known conflicts: must be arranged **now**
 - Emergencies: must talk to your class dean

How to indicate you're using a late day

QUESTIONS-00.txt

CS 21 Lab 0

Name:

Approximately how many hours did you take to complete this lab?
Provide your answer as a single integer on the line below.

How difficult did you find this lab?
(1-5, with 5 being very difficult and 1 being very easy)

Describe the biggest challenge you faced on this lab.

Number of Late Days Using for this lab:

0, 1, or 2



Syllabus highlights and notes

- Notes will be be posted *after* class on the webpage
- Textbook – free and online 😊
- Make sure to attend lab! (mandatory, attendance taken)
- Labs are (almost always) due Saturdays before midnight
- **Extensions**
 - Everyone gets 2 “late days” to be used for any reason
 - Known conflicts: must be arranged **now**
 - Emergencies: must talk to your class dean
- **Email** – allow 24 hours for a response (more over the weekend)

Syllabus highlights and notes

- Notes will be be posted *after* class on the webpage
- Textbook – free and online 😊
- Make sure to attend lab! (mandatory, attendance taken)
- Labs are (almost always) due Saturdays before midnight
- **Extensions**
 - Everyone gets 2 “late days” to be used for any reason
 - Known conflicts: must be arranged **now**
 - Emergencies: must talk to your class dean
- **Email** – allow 24 hours for a response (more over the weekend)
- **Piazza** should be used for all content/logistics questions

Syllabus highlights and notes

- Notes will be be posted *after* class on the webpage
- Textbook – free and online 😊
- Make sure to attend lab! (mandatory, attendance taken)
- Labs are (almost always) due Saturdays before midnight
- **Extensions**
 - Everyone gets 2 “late days” to be used for any reason
 - Known conflicts: must be arranged **now**
 - Emergencies: must talk to your class dean
- **Email** – allow 24 hours for a response (more over the weekend)
- **Piazza** should be used for all content/logistics questions
- Office hours: **3-5pm on Fridays in 249** (often moved to lab)

Class Deans contact info

CLASS	DEAN	To SCHEDULE AN APPOINTMENT WITH YOUR DEAN
First-Year	Dean Karen Henry	Betsy Durning 610-690-5744 edurnin1@swarthmore.edu
Sophomore	Dean Thomas Alexander III (Interim)	Stephanie Holznagel (assists with schedule only) 610-690-3999 sholzna1@swarthmore.edu
Junior	Dean Dion Lewis	Bonnie Lytle 610-328-8456 dlytle1@swarthmore.edu
Senior	Dean Michelle D. Ray	Stephanie Holznagel 610-690-3999 sholzna1@swarthmore.edu

Disability Services

<http://www.swarthmore.edu/academic-advising-support/welcome-to-student-disability-service>

Registering with Student Disability Services

1. Schedule a meeting with a Student Disability Services staff member to discuss your request(s). Any student is welcome to contact either Monica or Jenna, or email studentdisabilityservices@swarthmore.edu.
2. If you are a student and your last name starts with letters A through K, please contact Monica. Monica can be reached by emailing mvance1@swarthmore.edu, or calling 610-328-7358. If you are a student and your last name starts with letters L through Z, please contact Jenna. Jenna can be reached by emailing jrose2@swarthmore.edu, or calling 610-690-5538.
3. If you have never requested accommodations from Student Disability Services previously, please fill out a student accommodation request. This can be done by logging into [mySwarthmore](#) and completing the Accommodation form located under the Personal Information menu.
4. Meet with Student Disability Services staff for an intake meeting or phone conversation. At this meeting, you will discuss your experiences and how they are impacted by your condition. You may also discuss potential accommodations you may find helpful. The accommodations process is designed to be collaborative. Student Disability Services will ask you some questions regarding your experiences to better understand and support your request for accommodations.
5. Have a licensed professional fill out the [Disability Verification Form](#). Submit the completed form to the office. Other formal documentation will be accepted as well.
6. Student Disability Services will review your request in a timely fashion and notify you when a decision has been made regarding your accommodations request.

If you have questions or concerns, please reach out to Director [Monica Vance](#) (610-328-7358) or Assistant Director [Jenna Rose](#) (610-690-5538) or see the [Appeals Process](#).