

Welcome to CS 61A!

Lecture will begin at 2:10pm.

(Until then, you just get to watch us type.)

Looking for something to do?
Try reading the "Syllabus" link at the top of cs61a.org

Feel free to ask questions in the Zoom chat.

Welcome to CS 61A

Your Instructors

John DeNero
denero@berkeley.edu

Justin Yokota
jyokota@berkeley.edu

Your Instructors

John DeNero

denero@berkeley.edu

CS 61A instructor every Fall since 2011
(and some Spring semesters as well)

Justin Yokota

jyokota@berkeley.edu

Your Instructors

John DeNero

denero@berkeley.edu

CS 61A instructor every Fall since 2011
(and some Spring semesters as well)

Teaching Professor in EECS and help with
the Data Science undergraduate program

Justin Yokota

jyokota@berkeley.edu

Your Instructors

John DeNero

denero@berkeley.edu

CS 61A instructor every Fall since 2011
(and some Spring semesters as well)

Teaching Professor in EECS and help with
the Data Science undergraduate program

Research focused on machine translation

Justin Yokota

jyokota@berkeley.edu

Your Instructors

John DeNero

denero@berkeley.edu

CS 61A instructor every Fall since 2011
(and some Spring semesters as well)

Teaching Professor in EECS and help with
the Data Science undergraduate program

Research focused on machine translation

Office hours start next week (Aug 30):

Justin Yokota

jyokota@berkeley.edu

Your Instructors

John DeNero

denero@berkeley.edu

CS 61A instructor every Fall since 2011
(and some Spring semesters as well)

Teaching Professor in EECS and help with
the Data Science undergraduate program

Research focused on machine translation

Office hours start next week (Aug 30):

- 2–3 Tuesday in 101A Warren Hall

Justin Yokota

jyokota@berkeley.edu

Your Instructors

John DeNero

denero@berkeley.edu

CS 61A instructor every Fall since 2011
(and some Spring semesters as well)

Teaching Professor in EECS and help with
the Data Science undergraduate program

Research focused on machine translation

Office hours start next week (Aug 30):

- 2–3 Tuesday in 101A Warren Hall
- 8:45–9:30 Friday on Clark Kerr Track

Justin Yokota

jyokota@berkeley.edu

Your Instructors

John DeNero

denero@berkeley.edu

CS 61A instructor every Fall since 2011
(and some Spring semesters as well)

Teaching Professor in EECS and help with
the Data Science undergraduate program

Research focused on machine translation

Office hours start next week (Aug 30):

- 2–3 Tuesday in 101A Warren Hall
- 8:45–9:30 Friday on Clark Kerr Track
- 10:15–10:45 Friday on Memorial Glade

Justin Yokota

jyokota@berkeley.edu

Your Instructors

John DeNero

denero@berkeley.edu

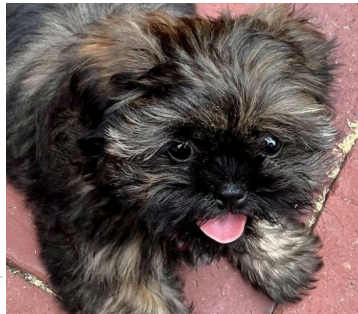
CS 61A instructor every Fall since 2011
(and some Spring semesters as well)

Teaching Professor in EECS and help with
the Data Science undergraduate program

Research focused on machine translation

Office hours start next week (Aug 30):

- 2–3 Tuesday in 101A Warren Hall
- 8:45–9:30 Friday on Clark Kerr Track
- 10:15–10:45 Friday on Memorial Glade



Justin Yokota

jyokota@berkeley.edu

Your Instructors

John DeNero

denero@berkeley.edu

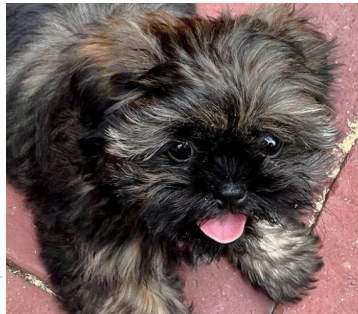
CS 61A instructor every Fall since 2011
(and some Spring semesters as well)

Teaching Professor in EECS and help with
the Data Science undergraduate program

Research focused on machine translation

Office hours start next week (Aug 30):

- 2–3 Tuesday in 101A Warren Hall
- 8:45–9:30 Friday on Clark Kerr Track
- 10:15–10:45 Friday on Memorial Glade



Justin Yokota

jyokota@berkeley.edu

First semester teaching 61A!

Your Instructors

John DeNero

denero@berkeley.edu

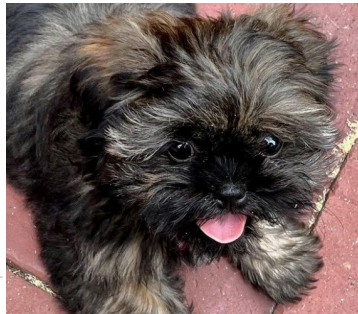
CS 61A instructor every Fall since 2011
(and some Spring semesters as well)

Teaching Professor in EECS and help with
the Data Science undergraduate program

Research focused on machine translation

Office hours start next week (Aug 30):

- 2–3 Tuesday in 101A Warren Hall
- 8:45–9:30 Friday on Clark Kerr Track
- 10:15–10:45 Friday on Memorial Glade



Justin Yokota

jyokota@berkeley.edu

First semester teaching 61A!

Prior to this, taught 61C as a TA/Head
TA/Instructor for 9 semesters

Your Instructors

John DeNero

denero@berkeley.edu

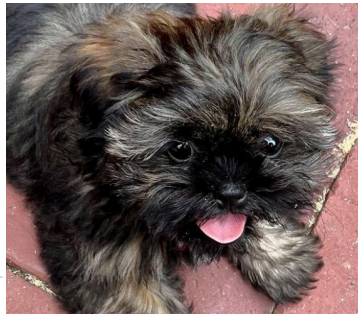
CS 61A instructor every Fall since 2011
(and some Spring semesters as well)

Teaching Professor in EECS and help with
the Data Science undergraduate program

Research focused on machine translation

Office hours start next week (Aug 30):

- 2–3 Tuesday in 101A Warren Hall
- 8:45–9:30 Friday on Clark Kerr Track
- 10:15–10:45 Friday on Memorial Glade



Justin Yokota

jyokota@berkeley.edu

First semester teaching 61A!

Prior to this, taught 61C as a TA/Head
TA/Instructor for 9 semesters

Master's at Berkeley working in
computational game theory

Your Instructors

John DeNero

denero@berkeley.edu

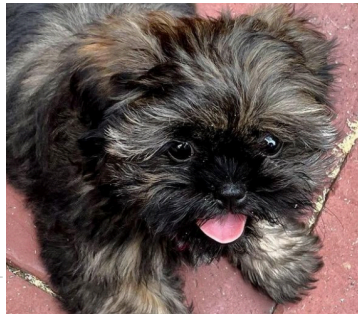
CS 61A instructor every Fall since 2011
(and some Spring semesters as well)

Teaching Professor in EECS and help with
the Data Science undergraduate program

Research focused on machine translation

Office hours start next week (Aug 30):

- 2–3 Tuesday in 101A Warren Hall
- 8:45–9:30 Friday on Clark Kerr Track
- 10:15–10:45 Friday on Memorial Glade



Justin Yokota

jyokota@berkeley.edu

First semester teaching 61A!

Prior to this, taught 61C as a TA/Head
TA/Instructor for 9 semesters

Master's at Berkeley working in
computational game theory

Office hours start next week (Aug 30):

Your Instructors

John DeNero

denero@berkeley.edu

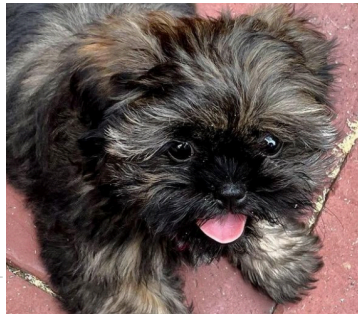
CS 61A instructor every Fall since 2011
(and some Spring semesters as well)

Teaching Professor in EECS and help with
the Data Science undergraduate program

Research focused on machine translation

Office hours start next week (Aug 30):

- 2–3 Tuesday in 101A Warren Hall
- 8:45–9:30 Friday on Clark Kerr Track
- 10:15–10:45 Friday on Memorial Glade



Justin Yokota

jyokota@berkeley.edu

First semester teaching 61A!

Prior to this, taught 61C as a TA/Head
TA/Instructor for 9 semesters

Master's at Berkeley working in
computational game theory

Office hours start next week (Aug 30):

- 1–2 Tuesday & Thursday in 781 Soda

Your Instructors

John DeNero

denero@berkeley.edu

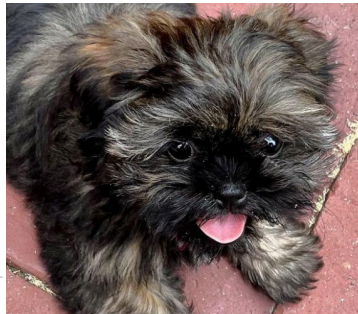
CS 61A instructor every Fall since 2011
(and some Spring semesters as well)

Teaching Professor in EECS and help with
the Data Science undergraduate program

Research focused on machine translation

Office hours start next week (Aug 30):

- 2–3 Tuesday in 101A Warren Hall
- 8:45–9:30 Friday on Clark Kerr Track
- 10:15–10:45 Friday on Memorial Glade



Justin Yokota

jyokota@berkeley.edu

First semester teaching 61A!

Prior to this, taught 61C as a TA/Head
TA/Instructor for 9 semesters

Master's at Berkeley working in
computational game theory

Office hours start next week (Aug 30):

- 1–2 Tuesday & Thursday in 781 Soda
- 4–5 Tuesday in 101A Warren Hall

Your Instructors

John DeNero

denero@berkeley.edu

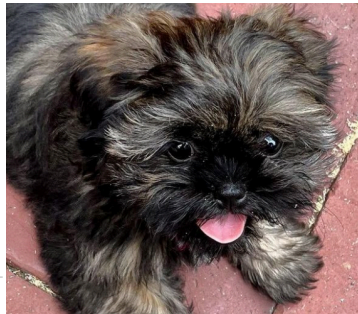
CS 61A instructor every Fall since 2011
(and some Spring semesters as well)

Teaching Professor in EECS and help with
the Data Science undergraduate program

Research focused on machine translation

Office hours start next week (Aug 30):

- 2–3 Tuesday in 101A Warren Hall
- 8:45–9:30 Friday on Clark Kerr Track
- 10:15–10:45 Friday on Memorial Glade



Justin Yokota

jyokota@berkeley.edu

First semester teaching 61A!

Prior to this, taught 61C as a TA/Head
TA/Instructor for 9 semesters

Master's at Berkeley working in
computational game theory

Office hours start next week (Aug 30):

- 1–2 Tuesday & Thursday in 781 Soda
- 4–5 Tuesday in 101A Warren Hall
- 2–3 Monday & Wednesday in 781 Soda

61A Course Staff



<https://cs61a.org/TAs/>

<https://cs61a.org/tutors/>

About the Course

Parts of the Course

Parts of the Course

Lecture: Videos posted to cs61a.org ***before*** live lecture

Parts of the Course

Lecture: Videos posted to cs61a.org ***before*** live lecture

Lab: Practice ideas from lecture on a computer (often in pairs)

Parts of the Course

Lecture: Videos posted to cs61a.org ***before*** live lecture

Lab: Practice ideas from lecture on a computer (often in pairs)

Discussion: Practice ideas from lecture on paper (often in groups)

Parts of the Course

Lecture: Videos posted to cs61a.org ***before*** live lecture

Lab: Practice ideas from lecture on a computer (often in pairs)

Discussion: Practice ideas from lecture on paper (often in groups)

Assignments: Weekly homework + 4 large projects

Parts of the Course

Lecture: Videos posted to cs61a.org ***before*** live lecture

Lab: Practice ideas from lecture on a computer (often in pairs)

Discussion: Practice ideas from lecture on paper (often in groups)

Assignments: Weekly homework + 4 large projects

Staff Office Hours: Get 1-on-1 help with assignments & work with peers

Parts of the Course

Lecture: Videos posted to cs61a.org ***before*** live lecture

Lab: Practice ideas from lecture on a computer (often in pairs)

Discussion: Practice ideas from lecture on paper (often in groups)

Assignments: Weekly homework + 4 large projects

Staff Office Hours: Get 1-on-1 help with assignments & work with peers

Online textbook: <http://composingprograms.com>

Parts of the Course

Lecture: Videos posted to cs61a.org ***before*** live lecture

Lab: Practice ideas from lecture on a computer (often in pairs)

Discussion: Practice ideas from lecture on paper (often in groups)

Assignments: Weekly homework + 4 large projects

Staff Office Hours: Get 1-on-1 help with assignments & work with peers

Online textbook: <http://composingprograms.com>

Monday–Wednesday: Attend lab and complete the lab assignment

Parts of the Course

Lecture: Videos posted to cs61a.org ***before*** live lecture

Lab: Practice ideas from lecture on a computer (often in pairs)

Discussion: Practice ideas from lecture on paper (often in groups)

Assignments: Weekly homework + 4 large projects

Staff Office Hours: Get 1-on-1 help with assignments & work with peers

Online textbook: <http://composingprograms.com>

Monday–Wednesday: Attend lab and complete the lab assignment

Wednesday–Friday: Attend discussion

Parts of the Course

Lecture: Videos posted to cs61a.org ***before*** live lecture

Lab: Practice ideas from lecture on a computer (often in pairs)

Discussion: Practice ideas from lecture on paper (often in groups)

Assignments: Weekly homework + 4 large projects

Staff Office Hours: Get 1-on-1 help with assignments & work with peers

Online textbook: <http://composingprograms.com>

Monday–Wednesday: Attend lab and complete the lab assignment

Wednesday–Friday: Attend discussion

Watch (or to go) lecture before you show up to lab/discussion!

Parts of the Course

Lecture: Videos posted to cs61a.org ***before*** live lecture

Lab: Practice ideas from lecture on a computer (often in pairs)

Discussion: Practice ideas from lecture on paper (often in groups)

Assignments: Weekly homework + 4 large projects

Staff Office Hours: Get 1-on-1 help with assignments & work with peers

Online textbook: <http://composingprograms.com>

Monday–Wednesday: Attend lab and complete the lab assignment

Wednesday–Friday: Attend discussion

Watch (or to go) lecture before you show up to lab/discussion!

Sunday/Monday/Tuesday: Start on the homework or project

Parts of the Course

Lecture: Videos posted to cs61a.org ***before*** live lecture

Lab: Practice ideas from lecture on a computer (often in pairs)

Discussion: Practice ideas from lecture on paper (often in groups)

Assignments: Weekly homework + 4 large projects

Staff Office Hours: Get 1-on-1 help with assignments & work with peers

Online textbook: <http://composingprograms.com>

Monday–Wednesday: Attend lab and complete the lab assignment

Wednesday–Friday: Attend discussion

Watch (or to go) lecture before you show up to lab/discussion!

Sunday/Monday/Tuesday: Start on the homework or project

Wednesday/Thursday: Finish the homework or project

Parts of the Course

Lecture: Videos posted to cs61a.org ***before*** live lecture

Lab: Practice ideas from lecture on a computer (often in pairs)

Discussion: Practice ideas from lecture on paper (often in groups)

Assignments: Weekly homework + 4 large projects

Staff Office Hours: Get 1-on-1 help with assignments & work with peers

Online textbook: <http://composingprograms.com>

Monday–Wednesday: Attend lab and complete the lab assignment

Wednesday–Friday: Attend discussion

Watch (or to go) lecture before you show up to lab/discussion!

Sunday/Monday/Tuesday: Start on the homework or project

Wednesday/Thursday: Finish the homework or project

Friday: Finish projects you didn't finish on Thursday

Asking Questions

?

Asking Questions



Ed: All staff (private posts) and students (public posts)

Asking Questions

?

Ed: All staff (private posts) and students (public posts)

cs61a@berkeley.edu: Head TAs and both instructors

Asking Questions

?

Ed: All staff (private posts) and students (public posts)

cs61a@berkeley.edu: Head TAs and both instructors

denero@berkeley.edu or **jyokota@berkeley.edu:** Often the slowest option

Asking Questions

?

Ed: All staff (private posts) and students (public posts)

cs61a@berkeley.edu: Head TAs and both instructors

denero@berkeley.edu or **jyokota@berkeley.edu:** Often the slowest option

cs61a.org: Self-service answers to many questions

Asking Questions



Ed: All staff (private posts) and students (public posts)

cs61a@berkeley.edu: Head TAs and both instructors

denero@berkeley.edu or **jyokota@berkeley.edu:** Often the slowest option

cs61a.org: Self-service answers to many questions

cs61a.org/contact/: Even more ways to reach the course staff

An Introduction to Computer Science

What is Computer Science?

What is Computer Science?

The study of

What is Computer Science?

The study of

What problems can be solved using computation,

What is Computer Science?

The study of

What problems can be solved using computation,
How to solve those problems, and

What is Computer Science?

The study of

- What problems can be solved using computation,
- How to solve those problems, and
- What techniques lead to effective solutions

What is Computer Science?

The study of Systems

What problems can be solved using computation,
How to solve those problems, and
What techniques lead to effective solutions

What is Computer Science?

The study of

- What problems can be solved using computation,
- How to solve those problems, and
- What techniques lead to effective solutions

Systems

Artificial Intelligence

What is Computer Science?

The study of

- What problems can be solved using computation,
- How to solve those problems, and
- What techniques lead to effective solutions

Systems

Artificial Intelligence

Graphics

What is Computer Science?

The study of

- What problems can be solved using computation,
- How to solve those problems, and
- What techniques lead to effective solutions

Systems

Artificial Intelligence

Graphics

Security

What is Computer Science?

The study of

- What problems can be solved using computation,
- How to solve those problems, and
- What techniques lead to effective solutions

Systems

Artificial Intelligence

Graphics

Security

Networking

Programming Languages

Theory

Scientific Computing

...

What is Computer Science?

The study of

- What problems can be solved using computation,
- How to solve those problems, and
- What techniques lead to effective solutions

Systems

Artificial Intelligence

Graphics

Security

Networking

Programming Languages

Theory

Scientific Computing

...

What is Computer Science?

The study of

- What problems can be solved using computation,
- How to solve those problems, and
- What techniques lead to effective solutions

Systems

Artificial Intelligence

Graphics

Security

Networking

Programming Languages

Theory

Scientific Computing

...

Decision Making

What is Computer Science?

The study of

- What problems can be solved using computation,
- How to solve those problems, and
- What techniques lead to effective solutions

Systems

Artificial Intelligence

- Decision Making

Graphics

- Robotics

Security

Networking

Programming Languages

Theory

Scientific Computing

...

What is Computer Science?

The study of

- What problems can be solved using computation,
- How to solve those problems, and
- What techniques lead to effective solutions

Systems

Artificial Intelligence

Graphics

Security

Networking

Programming Languages

Theory

Scientific Computing

...

Decision Making

Robotics

Natural Language Processing

What is Computer Science?

The study of

- What problems can be solved using computation,
- How to solve those problems, and
- What techniques lead to effective solutions

Systems

Artificial Intelligence

Graphics

Security

Networking

Programming Languages

Theory

Scientific Computing

...

Decision Making

Robotics

Natural Language Processing

...

What is Computer Science?

The study of

- What problems can be solved using computation,
- How to solve those problems, and
- What techniques lead to effective solutions

Systems

Artificial Intelligence

Graphics

Security

Networking

Programming Languages

Theory

Scientific Computing

...

Decision Making

Robotics

Natural Language Processing

...

What is Computer Science?

The study of

- What problems can be solved using computation,
- How to solve those problems, and
- What techniques lead to effective solutions

Systems

Artificial Intelligence

Graphics

Security

Networking

Programming Languages

Theory

Scientific Computing

...

Decision Making

Robotics

Natural Language Processing

...

Answering Questions

What is Computer Science?

The study of

- What problems can be solved using computation,
- How to solve those problems, and
- What techniques lead to effective solutions

Systems

Artificial Intelligence

Graphics

Security

Networking

Programming Languages

Theory

Scientific Computing

...

Decision Making

Robotics

Natural Language Processing

...

Answering Questions

Dialog

What is Computer Science?

The study of

- What problems can be solved using computation,
- How to solve those problems, and
- What techniques lead to effective solutions

Systems

Artificial Intelligence

Graphics

Security

Networking

Programming Languages

Theory

Scientific Computing

...

Decision Making

Robotics

Natural Language Processing

...

Answering Questions

Dialog

Translation

What is Computer Science?

The study of

- What problems can be solved using computation,
- How to solve those problems, and
- What techniques lead to effective solutions

Systems

Artificial Intelligence

Graphics

Security

Networking

Programming Languages

Theory

Scientific Computing

...

Decision Making

Robotics

Natural Language Processing

...

Answering Questions

Dialog

Translation

...

What is This Course About?

What is This Course About?

A course about managing complexity

What is This Course About?

A course about managing complexity

Mastering abstraction

What is This Course About?

A course about managing complexity

Mastering abstraction

Programming paradigms

What is This Course About?

A course about managing complexity

Mastering abstraction

Programming paradigms

An introduction to programming

What is This Course About?

A course about managing complexity

Mastering abstraction

Programming paradigms

An introduction to programming

Full understanding of Python fundamentals



What is This Course About?

A course about managing complexity

Mastering abstraction

Programming paradigms

An introduction to programming

Full understanding of Python fundamentals

Combining multiple ideas in large projects



What is This Course About?

A course about managing complexity

Mastering abstraction

Programming paradigms

An introduction to programming

Full understanding of Python fundamentals

Combining multiple ideas in large projects

How computers interpret programming languages



What is This Course About?

A course about managing complexity

Mastering abstraction

Programming paradigms

An introduction to programming

Full understanding of Python fundamentals

Combining multiple ideas in large projects

How computers interpret programming languages

Different types of languages: Scheme & SQL



What is This Course About?

A course about managing complexity

Mastering abstraction

Programming paradigms

An introduction to programming

Full understanding of Python fundamentals

Combining multiple ideas in large projects

How computers interpret programming languages

Different types of languages: Scheme & SQL

A challenging course that will demand a lot of you



Should you take CS 61A?

According to the Syllabus

There is no formal programming-related prerequisite for CS 61A, but...

- Taking the course without any prior programming experience is typically very challenging.
- Most CS 61A students have had significant prior programming experience.
- Students who take the course without prior programming experience typically must work substantially harder to master the material and tend to receive lower final grades in the course.

Students who take the course later often get more out of it due to increased understanding.

CS 10: The Beauty and Joy of Computing

CS 10: The Beauty and Joy of Computing



CS 10: The Beauty and Joy of Computing



CS 10: The Beauty and Joy of Computing

Designed for students without prior experience



CS 10: The Beauty and Joy of Computing

Designed for students without prior experience

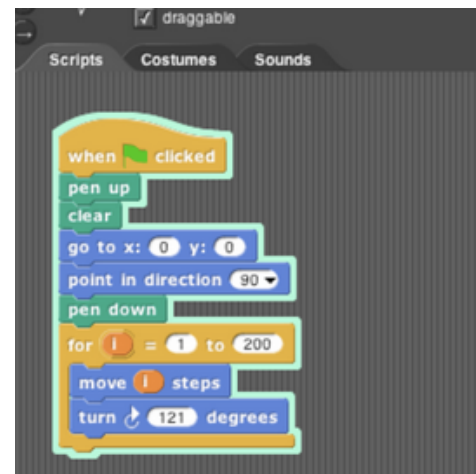
A programming environment created by Berkeley,
now used in courses around the world and online



CS 10: The Beauty and Joy of Computing

Designed for students without prior experience

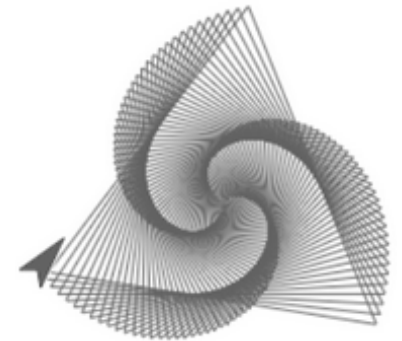
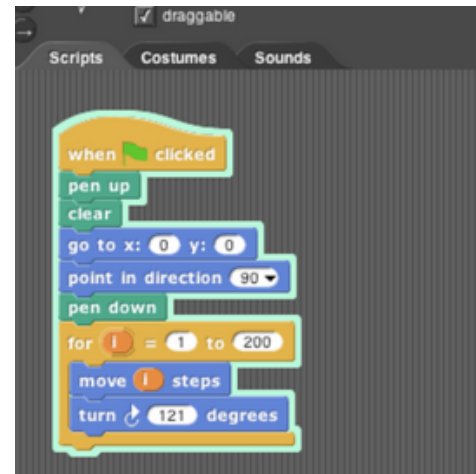
A programming environment created by Berkeley,
now used in courses around the world and online



CS 10: The Beauty and Joy of Computing

Designed for students without prior experience

A programming environment created by Berkeley,
now used in courses around the world and online

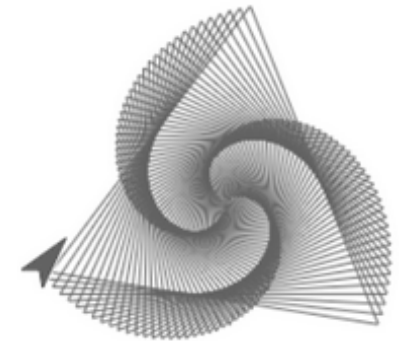
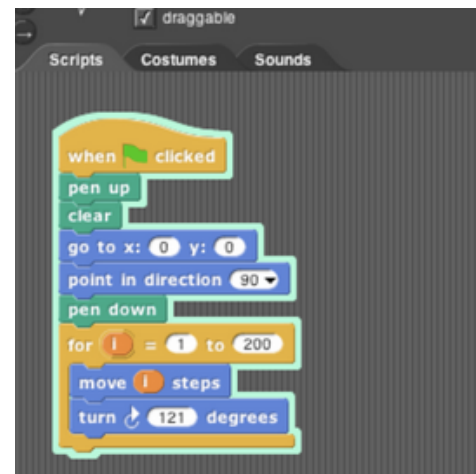


CS 10: The Beauty and Joy of Computing

Designed for students without prior experience

A programming environment created by Berkeley,
now used in courses around the world and online

An introduction to fundamentals (& Python)
that sets students up for success in CS 61A



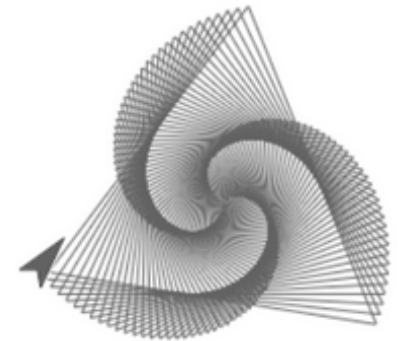
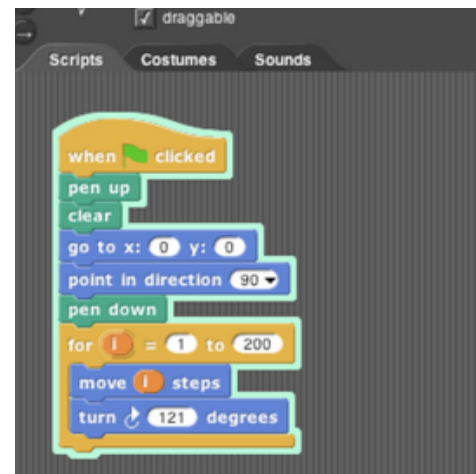
CS 10: The Beauty and Joy of Computing

Designed for students without prior experience

A programming environment created by Berkeley,
now used in courses around the world and online

An introduction to fundamentals (& Python)
that sets students up for success in CS 61A

It looks full, but seats are reserved for
students currently enrolled in CS 61A



CS 10: The Beauty and Joy of Computing

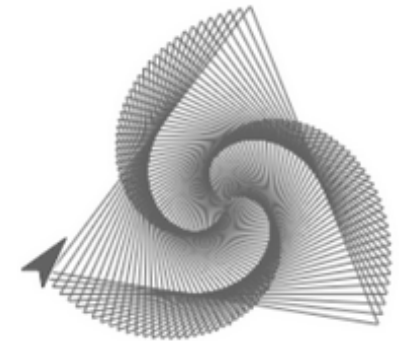
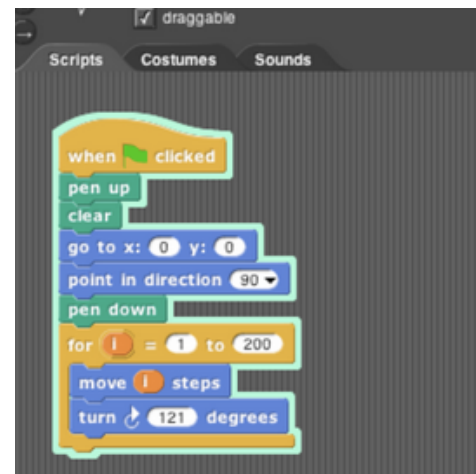
Designed for students without prior experience

A programming environment created by Berkeley,
now used in courses around the world and online

An introduction to fundamentals (& Python)
that sets students up for success in CS 61A

It looks full, but seats are reserved for
students currently enrolled in CS 61A

More info: <http://cs10.org/>



Data C88C (Formerly CS 88): Computational Structures in Data Science

Data C88C (Formerly CS 88): Computational Structures in Data Science

Based on CS 61A, but covers only 3 out of 4 units worth of the content:

Data C88C (Formerly CS 88): Computational Structures in Data Science

Based on CS 61A, but covers only 3 out of 4 units worth of the content:

- Two programming projects (instead of four) that are adapted from CS 61A projects

Data C88C (Formerly CS 88): Computational Structures in Data Science

Based on CS 61A, but covers only 3 out of 4 units worth of the content:

- Two programming projects (instead of four) that are adapted from CS 61A projects
- Everything you need to know to continue on to CS 61B

Data C88C (Formerly CS 88): Computational Structures in Data Science

Based on CS 61A, but covers only 3 out of 4 units worth of the content:

- Two programming projects (instead of four) that are adapted from CS 61A projects
- Everything you need to know to continue on to CS 61B
- Omits the unit on how programs run other programs

Data C88C (Formerly CS 88): Computational Structures in Data Science

Based on CS 61A, but covers only 3 out of 4 units worth of the content:

- Two programming projects (instead of four) that are adapted from CS 61A projects
- Everything you need to know to continue on to CS 61B
- Omits the unit on how programs run other programs

Designed for students taking Data 8 (Foundations of Data Science), but is now independent

Data C88C (Formerly CS 88): Computational Structures in Data Science

Based on CS 61A, but covers only 3 out of 4 units worth of the content:

- Two programming projects (instead of four) that are adapted from CS 61A projects
- Everything you need to know to continue on to CS 61B
- Omits the unit on how programs run other programs

Designed for students taking Data 8 (Foundations of Data Science), but is now independent

The course might be full, but we're investigating expansion options.

Course Policies

Course Policies

Learning

Learning
Community

Learning

Community

Course Staff

Learning
Community
Course Staff

Details...

<https://cs61a.org/articles/about/>

Collaboration

Collaboration

Working together is highly encouraged

Collaboration

Working together is highly encouraged

- Discuss everything with each other; learn from your fellow students!

Collaboration

Working together is highly encouraged

- Discuss everything with each other; learn from your fellow students!
- Some projects can be completed with a partner

Collaboration

Working together is highly encouraged

- Discuss everything with each other; learn from your fellow students!
- Some projects can be completed with a partner
- Choose a partner from your discussion section

Collaboration

Working together is highly encouraged

- Discuss everything with each other; learn from your fellow students!
- Some projects can be completed with a partner
- Choose a partner from your discussion section

What constitutes academic misconduct?

Collaboration

Working together is highly encouraged

- Discuss everything with each other; learn from your fellow students!
- Some projects can be completed with a partner
- Choose a partner from your discussion section

What constitutes academic misconduct?

- *Please* don't look at someone else's code!
Exceptions: lab, your project partner, or **after you already solved the problem**

Collaboration

Working together is highly encouraged

- Discuss everything with each other; learn from your fellow students!
- Some projects can be completed with a partner
- Choose a partner from your discussion section

What constitutes academic misconduct?

- *Please* don't look at someone else's code!
Exceptions: lab, your project partner, or **after you already solved the problem**
- *Please* don't tell other people the answers! You can point them to what is wrong and describe how to fix it or show them a related example

Collaboration

Working together is highly encouraged

- Discuss everything with each other; learn from your fellow students!
- Some projects can be completed with a partner
- Choose a partner from your discussion section

What constitutes academic misconduct?

- *Please* don't look at someone else's code!
Exceptions: lab, your project partner, or **after you already solved the problem**
- *Please* don't tell other people the answers! You can point them to what is wrong and describe how to fix it or show them a related example
- Copying project solutions causes people to fail the course

Collaboration

Working together is highly encouraged

- Discuss everything with each other; learn from your fellow students!
- Some projects can be completed with a partner
- Choose a partner from your discussion section

What constitutes academic misconduct?

- *Please* don't look at someone else's code!
Exceptions: lab, your project partner, or **after you already solved the problem**
- *Please* don't tell other people the answers! You can point them to what is wrong and describe how to fix it or show them a related example
- Copying project solutions causes people to fail the course
- We really do catch people who violate the rules, and we're getting better at it

Collaboration

Working together is highly encouraged

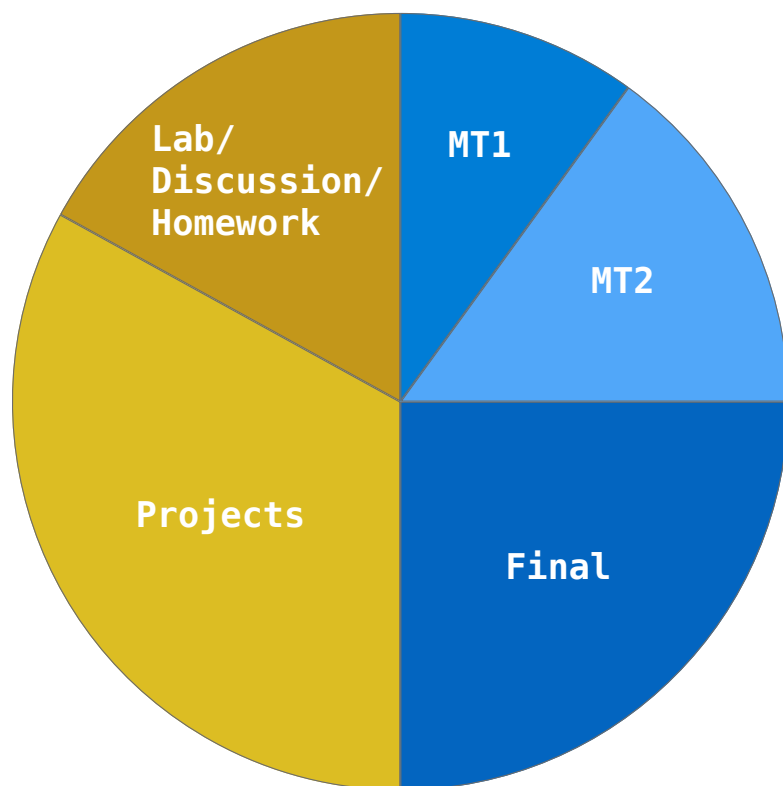
- Discuss everything with each other; learn from your fellow students!
- Some projects can be completed with a partner
- Choose a partner from your discussion section

What constitutes academic misconduct?

- *Please* don't look at someone else's code!
Exceptions: lab, your project partner, or **after you already solved the problem**
- *Please* don't tell other people the answers! You can point them to what is wrong and describe how to fix it or show them a related example
- Copying project solutions causes people to fail the course
- We really do catch people who violate the rules, and we're getting better at it

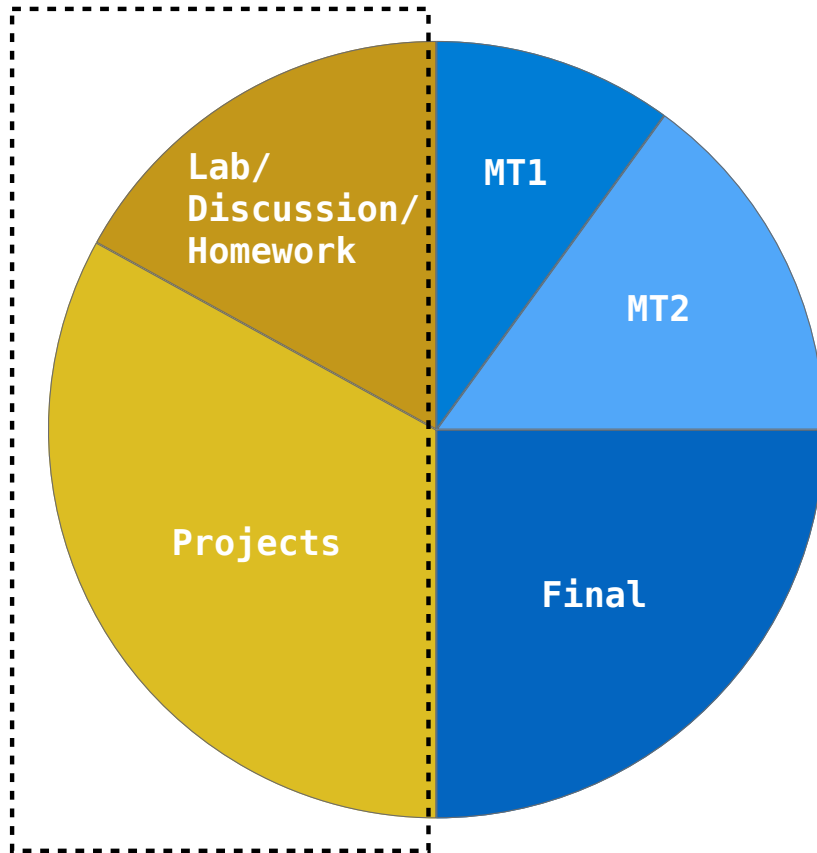
Build good habits now

How to Succeed in CS 61A



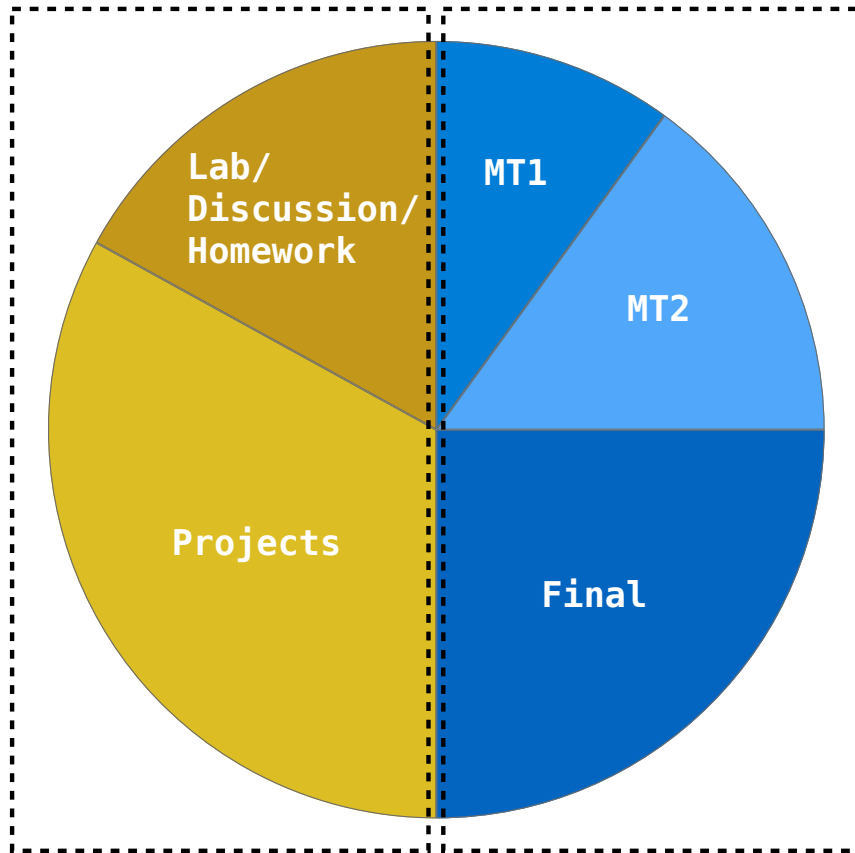
How to Succeed in CS 61A

Do all the work!



How to Succeed in CS 61A

Do all the work! *Exams matter a lot*

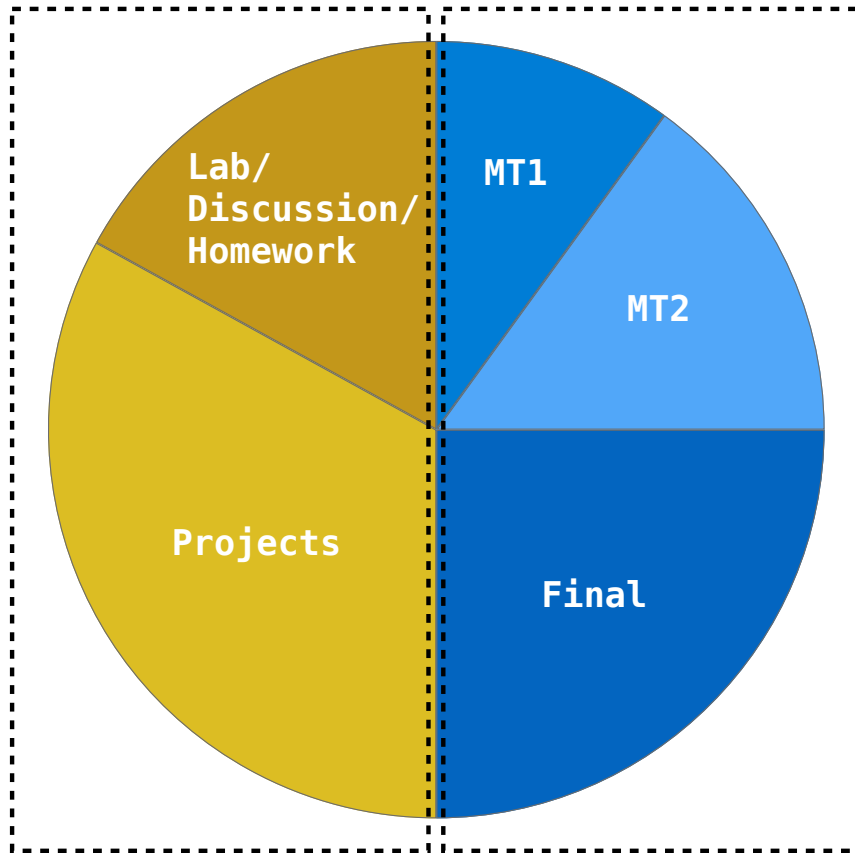


How to Succeed in CS 61A

Do all the work!

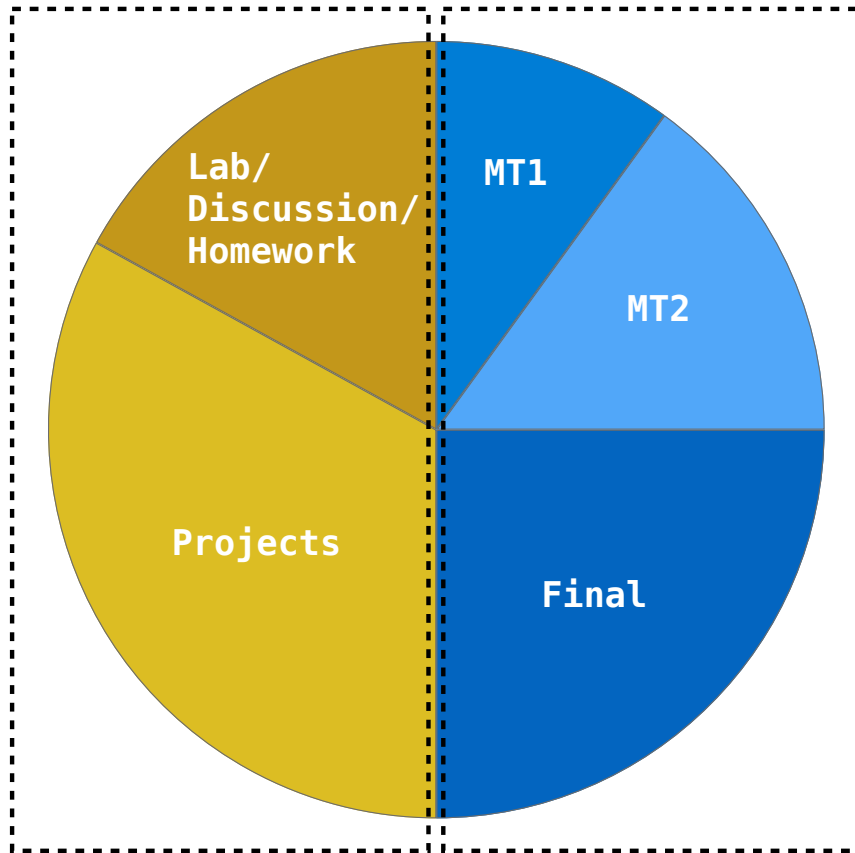
Exams matter a lot

Exam advice:



How to Succeed in CS 61A

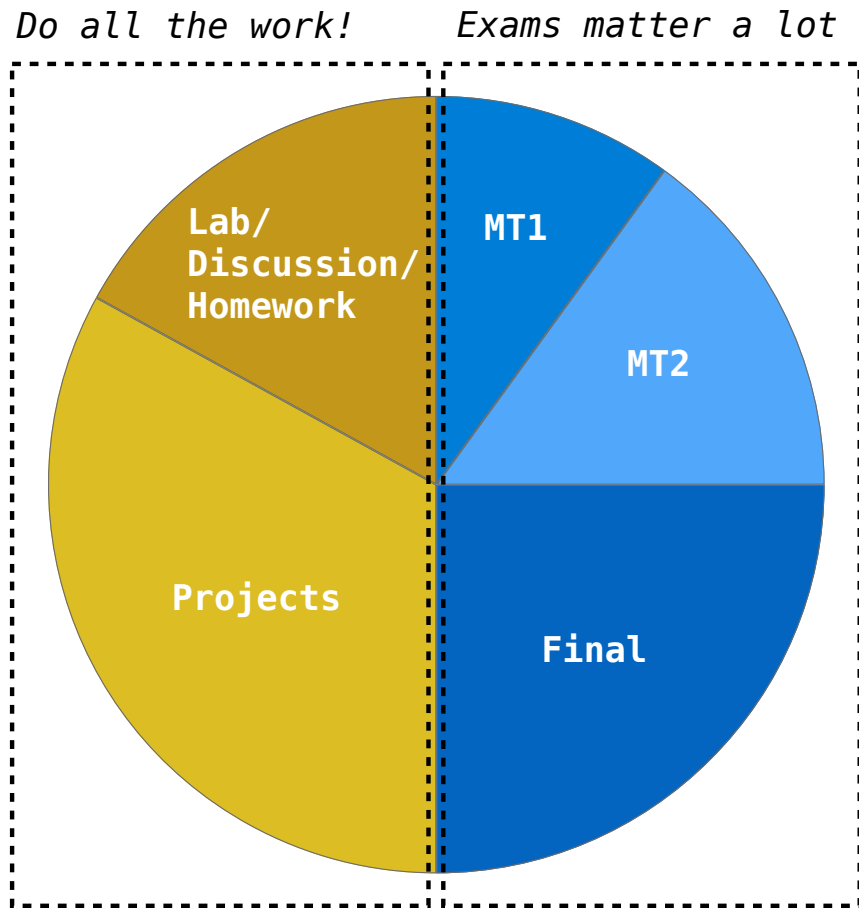
Do all the work! *Exams matter a lot*



Exam advice:

Study early & often

How to Succeed in CS 61A

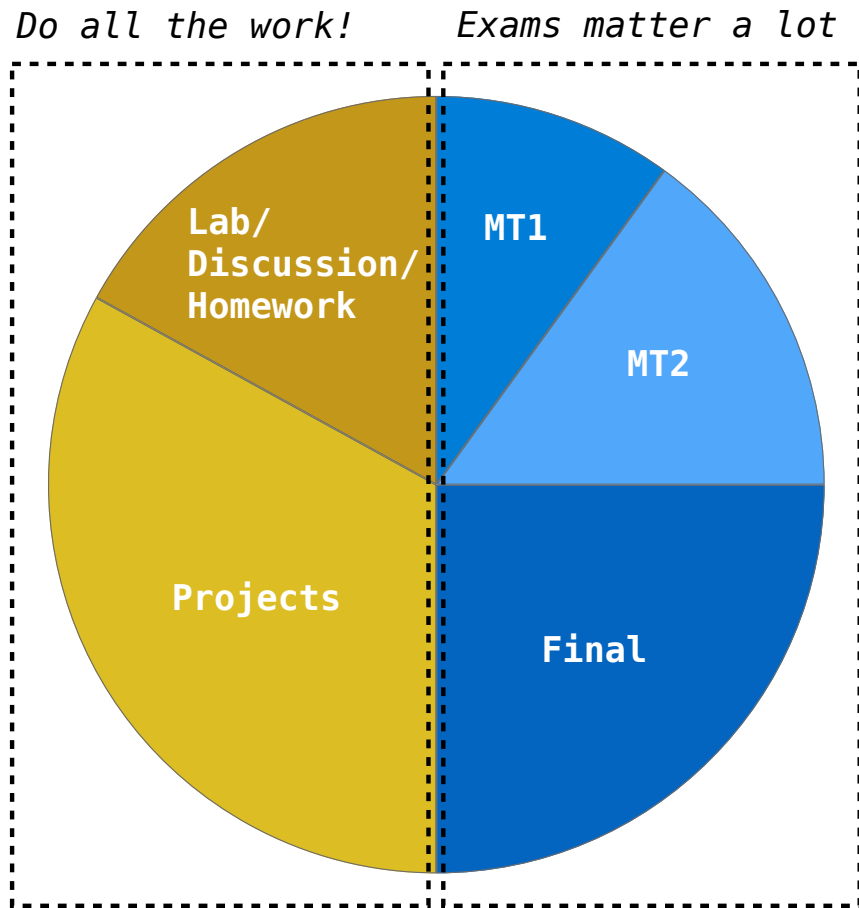


Exam advice:

Study early & often

Previous exams are a great way to test understanding & problem-solving skills

How to Succeed in CS 61A

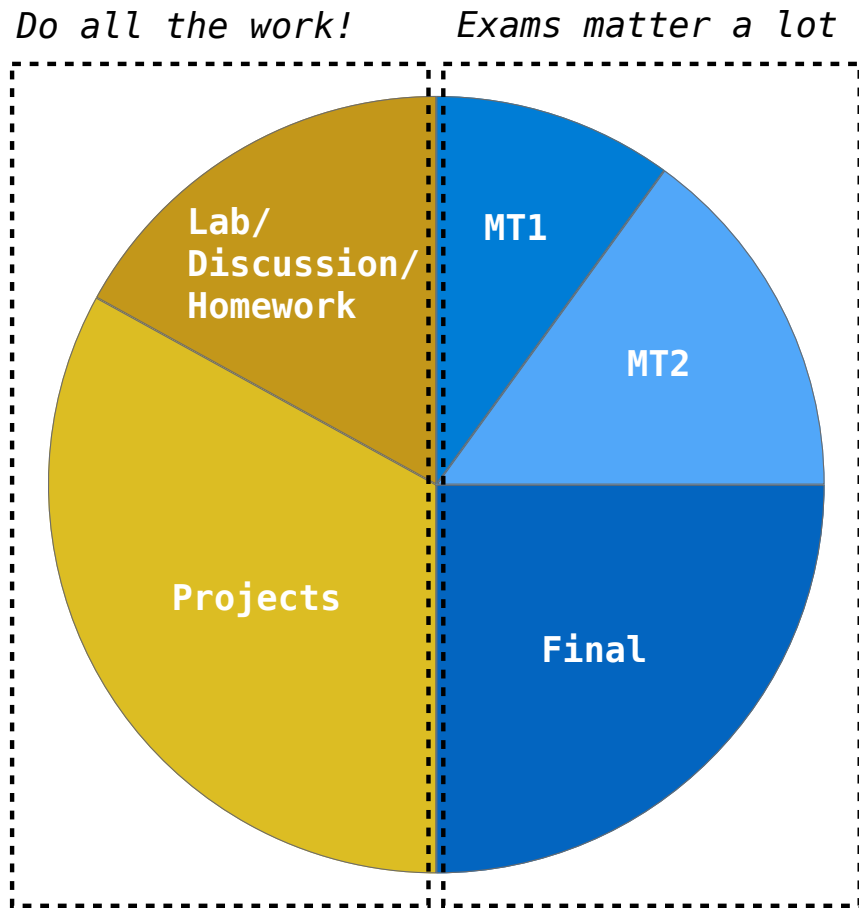


Exam advice:

Study early & often

Previous exams are a great way to test understanding & problem-solving skills

How to Succeed in CS 61A



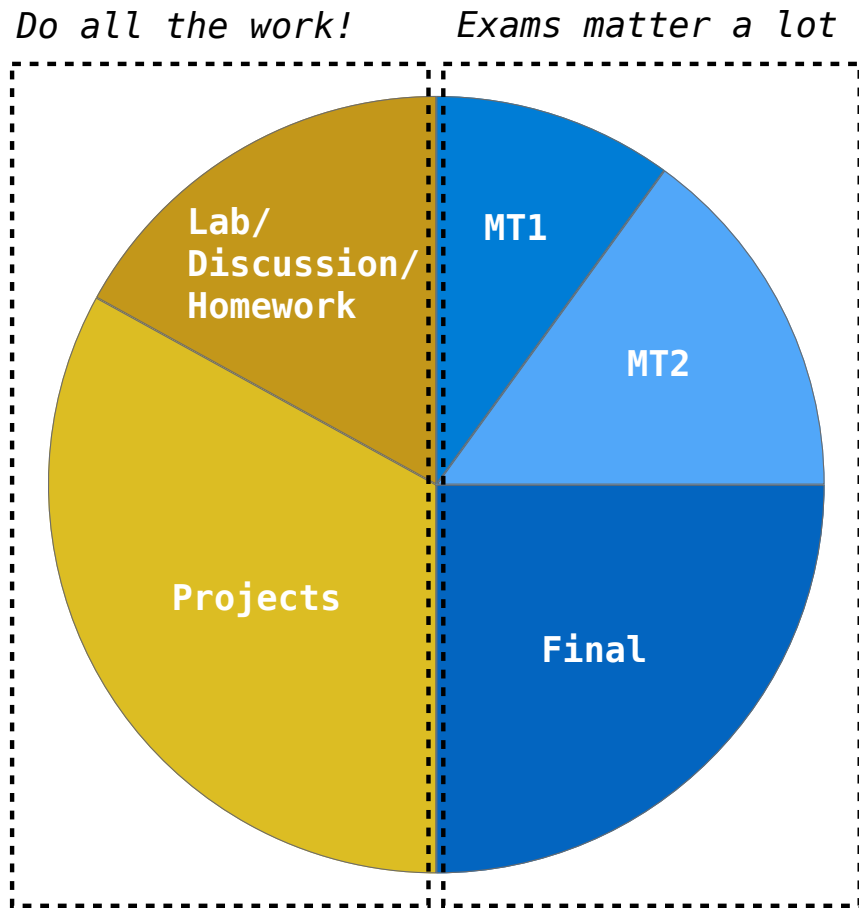
Exam advice:

Study early & often

Previous exams are a great way to test understanding & problem-solving skills

Assignment advice:

How to Succeed in CS 61A



Exam advice:

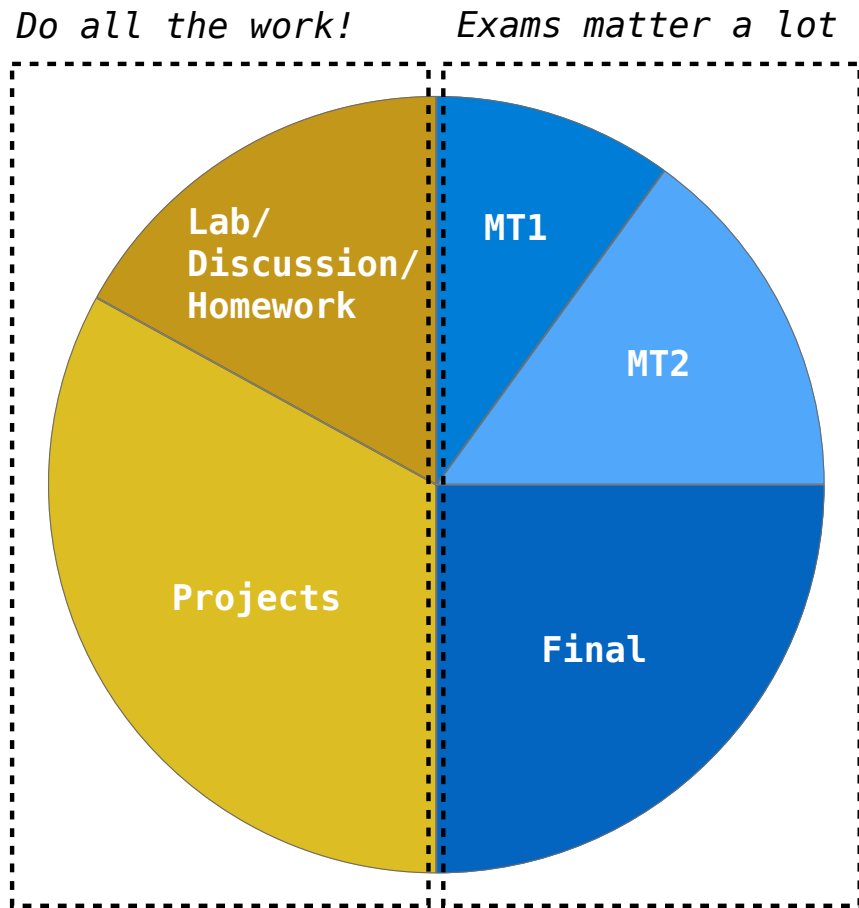
Study early & often

Previous exams are a great way to test understanding & problem-solving skills

Assignment advice:

Don't be afraid to ask for help!

How to Succeed in CS 61A



Exam advice:

Study early & often

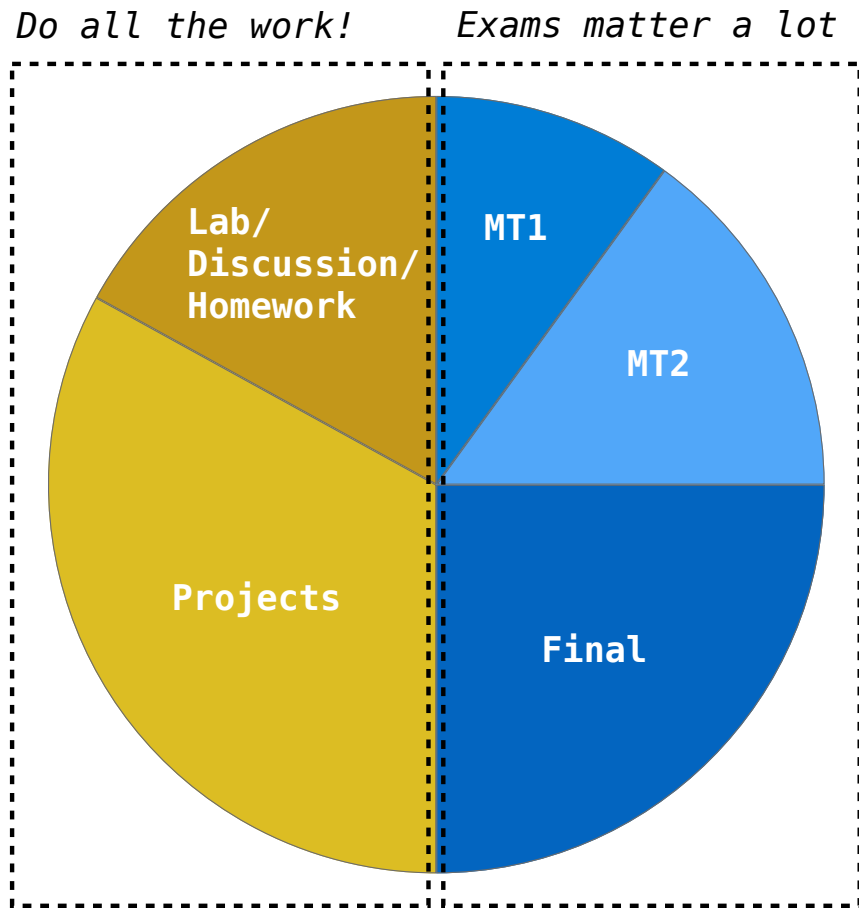
Previous exams are a great way to test understanding & problem-solving skills

Assignment advice:

Don't be afraid to ask for help!

Try using external resources (e.g., Google) to understand errors

How to Succeed in CS 61A



Exam advice:

Study early & often

Previous exams are a great way to test understanding & problem-solving skills

Assignment advice:

Don't be afraid to ask for help!

Try using external resources (e.g., Google) to understand errors

Try a personal project if you have time

Getting Help

If you're struggling, let us know.

If you need more time, ask for it.

If you need special accommodations, make an appointment.

Course Climate

Let's Stop Harassment & Discrimination

Let's Stop Harassment & Discrimination

Disparaging remarks targeting a particular gender, race, or ethnicity are not acceptable.

Let's Stop Harassment & Discrimination

Disparaging remarks targeting a particular gender, race, or ethnicity are not acceptable.

From the Berkeley Principles of Community:

Let's Stop Harassment & Discrimination

Disparaging remarks targeting a particular gender, race, or ethnicity are not acceptable.

From the Berkeley Principles of Community:

"We affirm the dignity of all individuals and strive to uphold a just community in which discrimination and hate are not tolerated."

Let's Stop Harassment & Discrimination

Disparaging remarks targeting a particular gender, race, or ethnicity are not acceptable.

From the Berkeley Principles of Community:

"We affirm the dignity of all individuals and strive to uphold a just community in which discrimination and hate are not tolerated."

From the EECS department mission:

Let's Stop Harassment & Discrimination

Disparaging remarks targeting a particular gender, race, or ethnicity are not acceptable.

From the Berkeley Principles of Community:

"We affirm the dignity of all individuals and strive to uphold a just community in which discrimination and hate are not tolerated."

From the EECS department mission:

"Diversity, equity, and inclusion are core values in the Department of Electrical Engineering and Computer Sciences. Our excellence can only be fully realized by faculty, students, and staff who share our commitment to these values."

Let's Stop Harassment & Discrimination

Disparaging remarks targeting a particular gender, race, or ethnicity are not acceptable.

From the Berkeley Principles of Community:

"We affirm the dignity of all individuals and strive to uphold a just community in which discrimination and hate are not tolerated."

From the EECS department mission:

"Diversity, equity, and inclusion are core values in the Department of Electrical Engineering and Computer Sciences. Our excellence can only be fully realized by faculty, students, and staff who share our commitment to these values."

All faculty and staff members are *mandated reporters*. If we ever receive a report of harassment, we must report to the Office for the Prevention of Harassment & Discrimination.

Let's Stop Harassment & Discrimination

Disparaging remarks targeting a particular gender, race, or ethnicity are not acceptable.

From the Berkeley Principles of Community:

"We affirm the dignity of all individuals and strive to uphold a just community in which discrimination and hate are not tolerated."

From the EECS department mission:

"Diversity, equity, and inclusion are core values in the Department of Electrical Engineering and Computer Sciences. Our excellence can only be fully realized by faculty, students, and staff who share our commitment to these values."

All faculty and staff members are *mandated reporters*. If we ever receive a report of harassment, we must report to the Office for the Prevention of Harassment & Discrimination.

- [CS61A Anonymous feedback form](#): If you want to stay anonymous but make us aware of something happening in the course.

Let's Stop Harassment & Discrimination

Disparaging remarks targeting a particular gender, race, or ethnicity are not acceptable.

From the Berkeley Principles of Community:

"We affirm the dignity of all individuals and strive to uphold a just community in which discrimination and hate are not tolerated."

From the EECS department mission:

"Diversity, equity, and inclusion are core values in the Department of Electrical Engineering and Computer Sciences. Our excellence can only be fully realized by faculty, students, and staff who share our commitment to these values."

All faculty and staff members are *mandated reporters*. If we ever receive a report of harassment, we must report to the Office for the Prevention of Harassment & Discrimination.

- [CS61A Anonymous feedback form](#): If you want to stay anonymous but make us aware of something happening in the course.
- [EECS Student Climate & Incident Reporting Form](#): Informs the EECS department of any issues. You can also contact Susanne Kauer (skauer@berkeley.edu) directly.

The Best Approach to CS 61A

Help each other understand concepts in the class, whether in section, on Ed, or in study groups, without expectation of anything in return.

Be great project partners by listening to what your partner suggests and helping them understand the work you've done together.

Recognize that we're all valuable members of the CS community!

Composing Programs

(Demo)