Computational Lab Skills for Cognitive Science

Joachim Vandekerckhove

Winter 2025

Course Description

This course provides an in-depth introduction to skills needed to conduct projects of a computational nature, with a focus on research in cognitive sciences.

Course Description

This course provides an in-depth introduction to skills needed to conduct projects of a computational nature, with a focus on research in cognitive sciences.

Topics Covered:

- Computational reproducibility

Course Description

This course provides an in-depth introduction to skills needed to conduct projects of a computational nature, with a focus on research in cognitive sciences.

Topics Covered:

- Computational reproducibility
- Version control

Course Description

This course provides an in-depth introduction to skills needed to conduct projects of a computational nature, with a focus on research in cognitive sciences.

Topics Covered:

- Computational reproducibility
- Version control
- Writing high-quality, maintainable code

Course Description

This course provides an in-depth introduction to skills needed to conduct projects of a computational nature, with a focus on research in cognitive sciences.

Topics Covered:

- Computational reproducibility
- Version control
- Writing high-quality, maintainable code
- Numerical simulations

Course Description

This course provides an in-depth introduction to skills needed to conduct projects of a computational nature, with a focus on research in cognitive sciences.

Topics Covered:

- Computational reproducibility
- Version control
- Writing high-quality, maintainable code
- Numerical simulations
- Model fitting

Course Description

This course provides an in-depth introduction to skills needed to conduct projects of a computational nature, with a focus on research in cognitive sciences.

Topics Covered:

- Computational reproducibility
- Version control
- Writing high-quality, maintainable code
- Numerical simulations
- Model fitting
- Numerical methods for integration and optimization

Tools

Tools and Environment:

- Docker Desktop

Tools

Tools and Environment:

- Docker Desktop
- Ubuntu Linux environment

Tools

Tools and Environment:

- Docker Desktop
- Ubuntu Linux environment
- Python programming language

Prerequisites

- (PSYC 14M or PSYC 114M or COGS 14P or ICS 31)

Prerequisites

- (PSYC 14M or PSYC 114M or COGS 14P or ICS 31)
- AND (PSYC 10C or STAT 7 or STAT 110)

Flipped Lectures

Recorded Lectures:

Flipped Lectures

Recorded Lectures:

- Published Monday afternoons

Flipped Lectures

Recorded Lectures:

- Published Monday afternoons
- Review before class time

Flipped Lectures

Recorded Lectures:

- Published Monday afternoons
- Review before class time

In-Person Meetings:

Flipped Lectures

Recorded Lectures:

- Published Monday afternoons
- Review before class time

In-Person Meetings:

- Wednesdays, 4:00 PM–4:50 PM

Flipped Lectures

Recorded Lectures:

- Published Monday afternoons
- Review before class time

In-Person Meetings:

- Wednesdays, 4:00 PM–4:50 PM

Virtual Meetings (Optional):

Flipped Lectures

Recorded Lectures:

- Published Monday afternoons
- Review before class time

In-Person Meetings:

- Wednesdays, 4:00 PM–4:50 PM

Virtual Meetings (Optional):

- Fridays, 4:00 PM–4:50 PM

Assessment

Individual Assignments:

- Done individually

Assessment

Individual Assignments:

- Done individually
- Graded based on meeting all requirements by the deadline

Assessment

Individual Assignments:

- Done individually
- Graded based on meeting all requirements by the deadline
- Grade of ‘A‘ requires all assignments on time and functional

Assessment

Group Assignments:

- Work in groups of 3–5

Assessment

Group Assignments:

- Work in groups of 3–5
- Notify the instructor of group members by the end of January

Assessment

Group Assignments:

- Work in groups of 3–5
- Notify the instructor of group members by the end of January
- Grading primarily based on group product quality

Assessment

Group Assignments:

- Work in groups of 3–5
- Notify the instructor of group members by the end of January
- Grading primarily based on group product quality
- Individual contributions evaluated via GitHub commits if needed

Assessment

Submission Policy:

- No late submissions (assignments due at noon on the due date)

Assessment

Submission Policy:

- No late submissions (assignments due at noon on the due date)
- GitHub snapshot taken at the deadline

Assessment

Submission Policy:

- No late submissions (assignments due at noon on the due date)
- GitHub snapshot taken at the deadline
- Recommendation: Set an internal deadline a day earlier than the due date

Week 2: Computational Reproducibility

Learn how to create reproducible computational environments using containers.

Making sure your computations can be reproduced is a foundational skill that—among other things—ensures that analyses can be replicated reliably.

- Assessment: Create a container and run Python in it

Week 3: Version Control

This week covers version control basics using GitHub, focusing on building a workflow that includes smooth and secure interaction with GitHub from a virtual environment.

- Assessment: Pull/push to GitHub from a container

Week 4: Object-Oriented Programming

Explore how to design modular and reusable code using object-oriented programming. Focus on creating objects tailored to specific tasks.

- Assessment: Write an object according to specifications

Week 5: No Class

A lighter week you can use for catching up and reviewing material, or focus on your midterms in other classes. Extra time for last week's assignment.

Week 6: Test-Driven Development

Learn to write robust, maintainable code by using test-driven development. Write tests first to guide your coding process. Test-driven development as a way of life.

- Assessment: Develop specifications for a project

Week 7: Optimization

Understand and apply optimization techniques to solve computational and statistical problems.

- Assessment: TBD

Week 8: Code Smells and Refactoring

Learn to identify “code smells” that indicate poor design and how to refactor for cleaner, more maintainable code.

- Assessment: Identify code smells and refactor as needed

Week 9: Simulation

Design and conduct numerical experiments using simulation techniques. Gain insights into the behavior of computational models.

- Assessment: Design and conduct a numerical experiment

Week 10: Integration & MCMC

Explore numerical integration and Markov Chain Monte Carlo methods for solving complex problems in cognitive science.

- Assessment: TBD

Academic Dishonesty

There is no tolerance for academic dishonesty or fraud. Any form of fraud designed to circumvent course policies will result in a failing grade. The professor makes no judgment calls regarding academic dishonesty. Any academic dishonesty, no matter how small, will be escalated to academic authorities.

Resources

Disability Services: https://dsc.uci.edu/

Academic Dishonesty: https://aisc.uci.edu/students/academic-integrity/index.php

Copyright Policy: http://copyright.universityofcalifornia.edu/use/teaching.html

Key Philosophy: Figure It Out

If you obtain a single skill in this class, I hope it is **Figuring It Out.**

Key Philosophy: Figure It Out

If you obtain a single skill in this class, I hope it is **Figuring It Out.**

> *The new education must teach the individual how to classify and reclassify information, how to evaluate its veracity, how to change categories when necessary, how to move from the concrete to the abstract and back, how to look at problems from a new direction—how to teach himself. Tomorrow's illiterate will not be the man who can't read; he will be the man who has not learned how to learn. (– Herbert Gerjuoy)*

Key Philosophy: Figure It Out

If you obtain a single skill in this class, I hope it is **Figuring It Out.**

- The first few weeks will focus on setting up safe sandboxes in which you can screw up your code or even your operating system.

Key Philosophy: Figure It Out

If you obtain a single skill in this class, I hope it is **Figuring It Out.**

- The first few weeks will focus on setting up safe sandboxes in which you can screw up your code or even your operating system.
- Once that is set up, you should get into a habit of Googling for solutions or asking an AI and experimenting and trying things out to see if they work.

Key Philosophy: Figure It Out

If you obtain a single skill in this class, I hope it is **Figuring It Out.**

- The first few weeks will focus on setting up safe sandboxes in which you can screw up your code or even your operating system.
- Once that is set up, you should get into a habit of Googling for solutions or asking an AI and experimenting and trying things out to see if they work.
- We will talk a lot about testing your own code and software.

Key Philosophy: Figure It Out

If you obtain a single skill in this class, I hope it is **Figuring It Out.**

- The first few weeks will focus on setting up safe sandboxes in which you can screw up your code or even your operating system.
- Once that is set up, you should get into a habit of Googling for solutions or asking an AI and experimenting and trying things out to see if they work.
- We will talk a lot about testing your own code and software.
- Finding partial solutions from unverified sources, implementing them, and then conducting rigorous tests is a highly generalizable coding paradigm.

AI Policy

General AI Policy:

AI Policy

General AI Policy:

- Use of tools like Copilot, ChatGPT, and generative AI is permitted.

AI Policy

General AI Policy:

- Use of tools like Copilot, ChatGPT, and generative AI is permitted.
- AI is a productivity enhancer but not a substitute for knowing what you are doing.

AI Policy

General AI Policy:

- Use of tools like Copilot, ChatGPT, and generative AI is permitted.
- AI is a productivity enhancer but not a substitute for knowing what you are doing.

Rules:

AI Policy

General AI Policy:

- Use of tools like Copilot, ChatGPT, and generative AI is permitted.
- AI is a productivity enhancer but not a substitute for knowing what you are doing.

Rules:

1. Students are individually responsible for all submissions.

AI Policy

General AI Policy:

- Use of tools like Copilot, ChatGPT, and generative AI is permitted.
- AI is a productivity enhancer but not a substitute for knowing what you are doing.

Rules:

1. Students are individually responsible for all submissions.
2. Acknowledge use of reference works, websites, and AI tools in comments.

Computational Lab Skills for Cognitive Science

Joachim Vandekerckhove

Winter 2025