CAMP: A Practical Replication Guide

Computational and Applied Mathematics Practicum

Becca Thomases | Smith College

Developed with support from the Richard Lounsbery Foundation

What is CAMP?

CAMP is a two-week, full-time introduction to the practice of computational and applied mathematics research. It is intended for undergraduates with some mathematical preparation but not necessarily prior research experience or extensive programming skills. Students learn computational methods while practicing the less formal skills of research: defining a manageable question, keeping a record of work, troubleshooting, communicating unfinished results, and identifying next steps. The emphasis is on the *process* of research, not on obtaining a publishable result.

The program combines short instructional sessions with long work blocks. Initially students complete guided computational tasks; over the second week they increasingly pursue questions of their own choosing, individually or in small groups. The model was piloted at Smith College in May 2025 and January 2026.

Getting started

- **Students:** Recruit students with some introductory mathematics, ideally including linear algebra. Do not require an existing research project. Prior programming experience can vary.
- **Format:** Plan for ten full instructional days over approximately two weeks, with sustained blocks of time for independent computational work and a daily lunch break. Arrange a classroom suited to individual and group computing, with space for discussions and presentations.
- **Computing:** Arrange MATLAB access in advance and ensure students can install or access it. MATLAB's free self-paced Onramp lessons provide a common starting point; individual projects can use other software when appropriate.
- **Teaching support:** Set aside regular instructor consultation time. With 15–18 students pursuing different topics, extra assistance from a teaching assistant or occasional guest can be useful, though not essential to the model.
- **Project preparation:** Assemble several accessible datasets and small example questions before the program. These are starting points, not compulsory project assignments.

Daily rhythm and two-week arc

A typical day includes a morning review and goal-setting discussion, a short mini-lecture or guided activity, sustained computational/project time, individual consultations, and a brief written reflection at the end of the day. Adapt times to the local schedule.

Days	Main activities
1–4	Computational introductions and mini-investigations; MATLAB Onramps; methods such as plotting, data exploration, least squares/PCA/SVD; daily reflections and research memos.
5	Short midpoint presentations with questions and peer feedback; identify possible directions for independent work.
6–9	Open-ended project work supported by targeted short lessons, individual consultations, scheduled progress check-ins, documentation, and troubleshooting.
10	Final short presentations, discussion of what was learned and what remains unresolved, and a closing reflection.

The particular computational topics are illustrative, not prerequisites for replication. The archived January 2026 material includes MATLAB plotting, least squares/PCA/SVD, exploratory data analysis, Connect-K/reinforcement learning, ODEs, image processing, and Git/GitHub. An instructor can substitute examples aligned with their expertise.

Student work and instructional routines

- **Onramp plan:** Students begin with core MATLAB skills and identify an interest-based pathway (data/machine learning; signals/images; or systems/modeling/control). Revisit the plan when projects change.
- **Daily reflection (5–10 minutes):** Record what was accomplished, something learned, a question or obstacle, and the first task for tomorrow. Use these for brief check-ins rather than as graded demonstrations of success.
- **Research memo:** Periodically summarize a question, methods tried, results or observations, outstanding problems, and next steps; refer to code and figures by recognizable filenames.
- **Project consultations:** Ask each student or team to state a concrete question, identify a feasible next computation, and describe a fallback if the original plan proves too ambitious. Encourage an initial question that can produce a meaningful computational result within a day or two, even if the larger project remains open-ended.
- **Presentations:** Have students explain their question, tools used, one or two observations, obstacles, and next steps. Encourage questions about incomplete work.

Practical advice for running the program

Give students genuine freedom of topic, but help them transition into it. Provide examples and an early project-scoping check-in; some students need more structure than others. Model productive responses to setbacks and allow projects to change direction. Reserve enough time for instructor feedback, and emphasize a useful intermediate result and sound documentation over the appearance of a finished research product.

To assess an offering, collect daily reflections, research memos, and presentation artifacts, together with a short exit survey. These provide qualitative evidence about students' engagement and developing research habits; they should not be treated as a standardized measure of technical proficiency. For longer-term evaluation, consider a brief follow-up asking whether students pursued research and used their CAMP skills.

Companion resource package

This guide is accompanied by a downloadable collection of teaching materials developed through the CAMP pilots and subsequent course preparation. The materials are organized into four folders:

1. **Program Materials:** Sample syllabi, schedules, and program planning documents.
2. **Student Templates:** MATLAB Onramp planning, daily reflections, research memos, and project-scoping worksheets.
3. **Computational Activities:** Selected MATLAB scripts, Live Scripts, exercises, and sample datasets covering data analysis, linear algebra, dynamical systems, reinforcement learning, and image processing.
4. **Presentations and Assessment:** Presentation instructions, peer-feedback materials, beginning/end reflections, and sample assessment assignments.

The materials are intended to be adapted rather than followed as a fixed curriculum. They include resources used during the 2025–2026 pilots as well as new activities and assignments developed for subsequent offerings.

Resource location: `https://beccathomases.github.io/camp.html`

Sharing notes: Instructors should review redistribution permissions for any additional datasets or images they incorporate. Student work should be shared only with appropriate permission and removal of identifying information where needed.