

Programming for Interactive Audio-Visual Arts

CS 2, Winter 2015

[About](#) [Schedule](#)

About

[Course Description](#) | [Administrative Info](#) | [Coursework and Grading](#) | [Honor Code](#) | [Disabilities](#)

Course description

This course introduces programming techniques necessary to generate interactive visual art on a computer. Although it briefly covers computer-generated media art, the course focuses on the programming skills required for creating *interactive* works. Rather than using commercial software, students write their own programs to create compositions with which users can interact. The course introduces fundamental concepts of how to represent and manipulate color, two-dimensional shapes, images, motion, and video. Coursework includes short programming assignments to practice the concepts introduced during lectures and projects to explore visual compositions. The course assumes no prior knowledge of programming, and is not open to students who have already had a programming course like CS 1 or CS 10.

We'll be using a language called [Processing](#), which does a lot of the "dirty work" to let us focus on creating fun interactive works, and still provides all the power of a full programming language.

For a taste of the material we'll cover, see [the last version](#) of the course.

Administrative Info

Instructor

Michael Casey
Hallgarten Hall, Room 103

Teaching assistants

Andy Sarroff (sarroff-at-cs-dot-dartmouth-dot-edu)
Carlos Dominguez (Carlos-dot-A-dot-Dominguez-at-Dartmouth-dot-edu)

Course staff email

cs2-at-cs-dot-dartmouth-dot-edu

Office Hours

Andy Sarroff | We 7-9 | Hallgarten Hall
Carlos Dominguez | Su 4-6 | Hallgarten Hall
Michael Casey | Fr 3-5, and arrange | Sudikoff, Room 207

Note: We will schedule additional special office hours as needed (e.g. on a Sunday before a long homework is due). Please email to the course staff *early* if a special office hour session will be useful.

Announcements and discussion board

[Canvas](#)

Lectures

MWF 1:45-2:50 | x-hour Th 1-1:50 | LSC 105

Textbook

Daniel Shiffman, *Learning Processing: A Beginner's Guide to Programming Images, Animation, and Interaction*

Other materials

[Processing reference](#)
Casey Reas and Ben Fry, *Processing: A Programming Handbook for Visual Designers and Artists*

Coursework and grading

While this class doesn't satisfy the language requirement, you are learning a new language and a new way of thinking. Most of us can't wait a week and cram material from a language class just in time for a paper or oral presentation. Likewise, most of us can't wait until the night before a performance and learn new things about a musical instrument and perfect the piece we are going to perform. In both of those cases, as well as in this class, it is best to stay on top of the material and practice often.

In that same vein, the best way to learn the material is by doing. After lecture try the examples on your own, modify them, come up with variations, etc. The lectures are constructed to help you do that. There aren't many core concepts, so most of each lecture is spent illustrating how to put a new concept to use in a number of different ways. The same is true of the textbook, and it even has short exercises to help you. It's like learning new words or grammatical constructs by using them in a conversation. Practice talking to yourself in this new language (don't worry about the looks people give you), and show off the results to your friends.

The lectures will largely parallel the textbook, where possible. While we will cover the same concepts, I will mostly illustrate them with different examples, trusting that you can read the text on your own later (or earlier), and that additional examples can only help you see how to put the concepts to use. I will clarify anything from the examples in the text that I think is particularly tricky, and if something from the text bothers you, by all means bring it up in class. For some chapters I won't cover all the material in the text, and for some material that I cover in class there isn't anything in the text. Lecture notes, including the examples from class, will be available on the [schedule page](#) for each day.

To tie together both the larger goal of creating fun interactive works, as well as the reality that practice makes perfect, the coursework will include a combination of assignments at various granularities.

Short assignments (20% total)

To help you keep on top of the material (think language lab or rehearsal). Accompanying a lecture (due by the next one); done individually.

Homeworks ($4 \times 12.5\% = 50\%$ total)

To help you put the basic techniques into practice in creating new works. Week-long; done individually.

Final project (20% total)

To allow you to synthesize the material to do something bigger of your own design. Done individually.

Mid Term (10% total)

To give a quick check-up on your understanding of the material. Middle of term; done individually.

Assignments are to be turned in via [Canvas](#) *before* class on the due date. Ample time is provided for each assignment, so in general, late submissions will not be accepted. Under extenuating circumstances (prolonged illness, death in the family, etc.), arrangements must be made with the instructor *before* the due date.

Grades will be assessed according to correctness and quality of both the *product* (does it look and act like it should?) and the *code* (is it well-organized and easy to follow, does it effectively use the proper programming constructs?). For exceptionally creative and interesting work, it is possible to receive extra credit points. Extra credit is tabulated separately from the regular grades on homework and exams, and is used to adjust final grades after the final average is computed. Thus extra credit is always optional, and not doing any extra credit work will never reduce your final grade, even if everyone else in the class does a lot of extra credit.

Honor code

Dartmouth's honor code applies to this course, and academic misconduct policies will be strictly enforced. If you have questions, ask!

You may discuss the assignments with other current CS2 students, but your submitted work must be entirely your own. That is, your code and any other solutions you submit must be created, written/typed, and documented by you alone. You may not copy anything directly from another student's work. For example, memorizing or copying onto paper a portion of someone else's solution would violate the honor code, even if you eventually turn in a different answer. Similarly, e-mailing a portion of your code to another student, or posting it on-line for them to see would violate the honor code. We do encourage discussion of assignments between students, subject to these rules.

If you make use of any code taken from outside references, i.e. a web page or a book, you must clearly attribute the source of the code with clear comments in the code that you submit. Proper respect for copyright laws as applied to printed materials and software products is subsumed by Dartmouth College's [Computing Policies](#).

On quizzes, you may not collaborate with, copy from, or otherwise share information with anyone. (You may of course ask questions of the course staff.)

Disabilities

Students with disabilities enrolled in this course and who may need disability-related classroom accommodations are encouraged to make an appointment to see the instructor before the end of the second week of the term. All discussions will remain confidential, although the [Student Accessibility Services](#) office may be consulted to discuss appropriate implementation of any accommodation requested.