

MDST 7803

Computational Media

Instructor

Professor Kevin Driscoll

210 Wilson Hall

kdriscoll@virginia.edu

<https://calendly.com/driscollofficehours>

Class meetings

Tuesday and Thursday

9:30 AM – 10:45 AM

Ruffner Hall 175

Course description

Computers are universal media. Our intimacy with computers shapes how we think about our communities, histories, cultures, society, and ourselves. Learn to program these "thinking machines" as an act of philosophical inquiry and personal expression, challenging your beliefs about creativity, intelligence, randomness, and communication. Students with no previous programming experience are especially welcome!

Motivating questions

- How do programming languages encode human ideas?
- Can you make an argument with a computer program?
- Can a computer program be beautiful?
- Is writing about software different from writing about a film or a novel?
- Why should anyone learn to code?

Course goals

Learning to program a computer is a strange experience. Like learning to play a musical instrument, programming is a technical skill that enables expression, collaboration, and inquiry. But, also like playing a musical instrument, learning to program a computer requires time, patience, and practice. Whether you want to play jazz on a piano or make beats on a sampler, your creative potential will be constrained, in part, by your technical proficiency with the instrument.

Your challenge this semester is to dedicate yourself to the craft of computer programming. This journey will be, by turns, fun, frustrating, bewildering, and enlightening. By developing the habits of mind and practical techniques of computer programming, you will transform your computer from an information appliance—a microwave oven for data—into an electronic instrument for computation and creativity.

Assignments

Weekly reflections

Each week, I will start a discussion thread on Collab. This will usually involve a short reading. You may reply directly to my original post, pick up a topic introduced by another student, or introduce your own tangent. Your responses do not need to be long (~100-200 words) but they should demonstrate thoughtful engagement with the prompt. **Due on Monday before noon.**

Competency exams

You will take four competency exams. Each exam will (1) assess your understanding of the technical language used by computer programmers, (2) ask you to interpret the meaning of example programs, and (3) challenge you to find and correct mistakes in example programs. There is no final exam.

Programming projects

You will complete four exploratory programming projects. The first project will be about representations of time. The second and third involve the automated manipulation of text and images. The fourth requires communication between your program and other machines, human beings, or the natural world. At the start of each project, I will give you a prompt and set of creative constraints to work with. During the last week of class, each student will give a short presentation about one of their projects.

Conference paper

You will draft a conference paper (2,400-3,000 words) to accompany your fourth programming project. Writing for an expert audience engaged with software studies, critical code studies, and related fields, the central argument of your paper should involve a critical analysis of a software system. The objects of study may be speculative, historical, commercial, or artistic. You will meet with me after submitting your second project to discuss potential topics for the paper. All students will be required to complete additional reading assignments tailored to the needs of the paper.

Assessment

- Attendance and participation (10%)
- Weekly posts on Collab (10%)
- Exams (20%)
- Projects (40%)
- Paper (20%)

Collab will convert your final grade into a letter using the default settings:

<http://its.virginia.edu/sis/grading/gradethresholds.html>

Technology Policy

You are required to bring a laptop to every class. You should also consider bringing a pencil and notebook to class. You may be surprised to find how useful these tools can be when planning a new program or trying to fix a bug. Cellphones should remain out of sight.

Attendance Policy

Regular attendance and active participation are requirements of this course. You are allowed two unexcused absences to take at your discretion. No explanation required. Each additional absence will lower your final grade. If you anticipate missing more than two class meetings, please contact me as soon as possible to determine if an alternate arrangement is possible. If you need to leave early or arrive late to a class, let me know in advance and make your entrance or exit as quietly as possible.

Resources

Technology

All of our work, in and out of the classroom, will require access to a laptop of your own. (The brand and operating system are not important.)

If your laptop is not set up to access the UVA VPN, following the instructions provided by ITS:

- <http://its.virginia.edu/vpn/>

If you use Windows, you will need PuTTY to access our server. Download and install the 32-bit version:

- <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

Plan to bring your laptop to every class meeting.

Books

The required books for this course are on sale at the University of Virginia bookstore. I recommend print copies so that you can write notes on the pages.

- Montfort, Nick. *Exploratory Programming for the Arts and Humanities*. The MIT Press, 2016.
- Vee, Annette. *Coding Literacy*. The MIT Press, 2017.

We will also read several chapters from *Code* by Charles Petzold. An eBook is available through the UVA Library at no cost but you may prefer a print copy.

- Petzold, Charles. *Code: The Hidden Language of Computer Hardware and Software*. Microsoft Press, 1999. <https://proquest.safaribooksonline.com/9780735634688>.

Finally, *Think Python* is an excellent companion text that offers a more straightforward overview of the Python programming language than *Exploratory Programming*. *Think Python* is available in multiple formats including HTML, PDF, and print. Whichever you choose, be sure that you have the 1st edition.

- Downey, Allen B. *Think Python: How to Think Like a Computer Scientist*. 1st edition. Sebastopol, CA: O'Reilly Media, 2015. <http://greenteapress.com/wp/think-python/>.

Schedule

Tuesday, January 14 **What is data? What is code?**

Thursday, January 16 **PART 1: TIME**

Reading: Chapters 7, 8, and 9 in *Code*

Tuesday, January 21 **Why do we call these machines "computers"?**

Reading: Chapter 2, "Calculating", in *Exploratory*
 Chapter 3, "Double, double," in *Exploratory*
 Chapter 15 in *Code*

Thursday, January 23

Reading: *Code*, Chapter 23
 Exploratory, Chapter 4, "Programming fundamentals"

Tuesday, January 28 **Can a computer program surprise you?**

Thursday, January 30

Due: Project 1

Tuesday, February 4 **PART 2: TEXT**

Do computer programs read?

Reading: *Exploratory*, Chapter 6, "Text 1"
 Exploratory, Chapter 7, "Text 2"

Due: Exam 1

Thursday, February 6

Reading: *Code*, Chapter 20

Tuesday, February 11 **Can a computer program help you read differently?**

Reading: *Exploratory*, Chapter 10, "Text 3"

Thursday, February 13

Tuesday, February 18 **Does writing code change how you think about human language?**

Thursday, February 20

Due: Project 2

Meet with instructor to discuss paper topics

Tuesday, February 25 PART 3: IMAGE

What do computer programs see?

Due: Exam 2

Reading: *Exploratory*, Chapter 8

Thursday, February 27

Reading: *Exploratory*, Chapter 9

Tuesday, March 3 What makes two images alike or different?

Thursday, March 5

Tuesday, March 10 Spring break

No class

Thursday, March 12 Spring break

No class

Tuesday, March 17 What do flipbooks and GIFs have in common?

Thursday, March 19

Due: Project 3

Tuesday, March 24 PART 4: COMMUNICATION

How do computer programs communicate with people?

Due: Exam 3

Reading: *Exploratory*, Chapter 14, "Interaction" (skip section 14.2)

Thursday, March 26

