"The limits of my language mean the limits of my world."

– Ludwig Wittgenstein

"The process of preparing programs for a digital computer is especially attractive, not only because it can be economically and scientifically rewarding, but also because it can be an aesthetic experience much like composing poetry or music."

– Donald Knuth

CREATIVE PROGRAMMING 1

DETAILS

Instructor	Prof. Jeff Thompson
Email	jeff.thompson@stevens.edu
Office/hours	Morton 208, Tues/Thurs 2–3pm
Meeting times	Tuesdays 9.00am–12.50pm
Location	Visual Arts & Technology Lab
Course materials	www.github.com/jeffThompson/ CreativeProgramming1

COURSE DESCRIPTION

In this class, we will explore the computer as a tool capable of powerful creative possibility, not via pre-built software, but instead by writing code ourselves. We will look at the basic structures and affordances of code as inspiration for making artworks, as a tool capable of creating things that would be impossible by hand, and as a fallible system that encapsulates our cultural and personal biases.

During the course of the semester, you'll learn how to write code for a variety of visual projects, including image, text, and interaction. We'll primarily be using the Processing platform, a toolkit created specifically for artists and designers built on the language Java, with a brief exploration of Python as a tool for building Twitter bots.

Along the way, we'll also look at historical and contemporary figures in the arts and computer science who have shaped how we use computers as creative tools, and we'll explore code from a critical, humanistic perspective.

ATTENDANCE

Due to the condensed, technical, and collaborative nature of this class, attendance is mandatory. You are allowed two absences per semester to use at your discretion – each additional absence will result in your final grade being lowered by ½-letter. Late arrivals will be marked tardy, with 3 tardies equaling one absence. The only exception is severe illness – if this is the case,

please let me know as soon as possible and provide a doctor's note documenting your illness.

HOMEWORK

Homework in this class is meant to be exploratory, a way to expand on the experiences and ideas in class. I encourage wide-ranging interpretation of assignments: consider ways that you can fulfill the requirements in a way that is creatively and intellectually exciting for you, not just the obvious requirements. Of course, this is much harder than just reading a chapter or studying for a quiz! I expect considerable engagement from you this semester, and you should expect the material to be rigorous and thorough.

All assignments are due by the start of class – details of projects will be available on the class GitHub page (see link on the first page) including how to turn your work in.

You will have 24/7 access to the Lab and Studio, and use of the Fab Lab during open hours for printing and equipment checkout.

GRADING

The goal of all assignments is for you to think and make. Everyone comes from a different background and experience, so the goal is improvement – I want to see curiosity, engagement, and willingness to experiment. A grading rubric will be provided

with each assignment to help you understand what is expected and how you did.

To get a C (an average grade) you should:

+ Put time into your projects each week

+ Complete everything on time

+ Participate in critiques and discussions

For a B or an A, you should additionally:

+ Take risks and try things enthusiastically

+ Be an active and unsolicited participant in critiques and discussions

+ Take assignments beyond their minimum requirements

Final grades will be determined as follows:

+ Homework: 60%

+ Class participation: 25%

+ Final project: 15%

REQUIRED MATERIALS

Required and suggested readings will be provided as PDFs on GitHub – there is no required textbook.

+   Laptop and charger, capable of running Processing/Java and with reliable internet connection – *bring every week!*

+   A notebook or sketchbook for taking notes and drawing ideas – *bring every week!*

+   Some kind of writing implement – an assortment of various pens and pencils may be helpful for working on project ideas

+   Parts and supplies for Physical Memory project

+   Other art supplies (paper, etc) and printing as needed

Topics noted with ⊨ are short explorations of topics that jump off from or surround programming in a creative context. The symbol⊨ is used in the field of logic to mean that an idea semantically entails another.

COURSE CALENDAR

Please note this is subject to change – be sure to check GitHub and your email regularly. Readings are listed for the days they are due.

WEEK 00: INTRODUCTION

In class:	Hello, introductions and syllabus, programming a person to sit in a chair
Reading:	"Alan Kay's Universal Machine" from *Software Takes Command* (Manovich)
Homework:	Install Processing

WEEK 01: DRAWING && COORDINATE SYSTEMS

In class:	Creating sketches, "Hello World," RGB color, drawing shapes, fill and stroke, saving images, getting help
Homework:	Drawings of Robots

WEEK 02: ITERATION && LOOPS

In class:	For loops, nested loops, driving parameters with loops, push/popMatrix(), creating functions, historical quilt research
Homework:	Algorithmic Quilts

WEEK 03: RECURSION && FRACTALS

In class:	Creating recursive functions, Pythagoras Tree, Koch Snowflake, pixel[] and Mandelbrot Set
Homework:	Fractal Exploration

WEEK 04: TIME

In class:	Conditionals, strings, fonts and text, exporting video files
Reading:	Excerpts from *Coding Freedom* (Coleman), Debian Constitution
Homework:	Three Clocks

WEEK 05: INTERACTION 1
(And ⊨ Open Source)

In class:	Discuss readings, contributing to open source projects, mouseX/Y, mousePressed()/Dragged(), background with alpha
Homework:	Drawing App

WEEK 06: RANDOMNESS

In class:	random(), map() and constrain(), arrays, writing to PDF, using the pen plotter
Homework:	Exploration of Randomness, get supplies for Physical Memory project

WEEK 07: ⊨ PHYSICAL MEMORY

In class:	Historical examples of memory storage systems, ideating and prototyping new memory formats
Homework:	Physical Memory prototype, make sure Python is installed

MARCH 13–17
Spring break – no class!

WEEK 08: LANGUAGE 1

In class:	Creating a Python script, printing "Hello World," running a script in the Terminal, loops and conditionals, lists, cleaning text, templating strings, saving to file
Homework:	Bot (Part 1), create a Twitter account

WEEK 09: LANGUAGE 2
(And ⊨ Machine Learning)

In class:	Installing Python modules, creating an app on Twitter, OAuth, posting to Twitter automatically, RPi bot sever demo, super basic intro to machine learning and how to train a text-generating network
Homework:	Bot (Part 2), train a neural net on found text

WEEK 10: ⊨ CRITICAL CODE STUDIES

In class: Reading code as text, porting code as critical
              research
Homework: Critical Code Analysis, Final Project proposal

WEEK 11: INTERACTION 2

In class: Creating classes, ArrayLists, particle systems
Mini project: Digital Fireworks
Homework: Rough Final Project prototype

WEEK 12: INTERACTION 3

In class: Gravity and physics, simulating motion
Mini project: Finish Digital Fireworks
Homework: Continue working on Final Project

WEEK 13: WORK DAY

In class: Work on Final Projects, get help as needed
Homework: Finish Final Project

EXAM PERIOD: FINAL CRITIQUE
Date TBD, please don't book travel until after exam period ends!

In class: Crit and goodbye