



Course Information

Course Title: Creative Coding for Digital Media

Course Number: IMDM127

Term: Summer 2022

Credits: 3

Course Times: MoTuWedTh 11:00-2:45pm

Prerequisites: None

Professor: Johab Silva

Pronouns: He, Him, His

Email; jnsilva@umd.edu

Office hours: TBD

Restriction: *Restricted to students in Terp Young Scholars.*

Course Description:

An introduction to the principles of Computer Science supported by exercises in computer programming with an emphasis on creative coding, algorithmic image creation and manipulation, and interactive experiences. Students will make use of both exploratory coding approaches, and problem/solution-driven approaches, to design and implement software with visual and auditory output. The course also includes an introduction to a wide variety of issues relating to computer science and software, including software design and construction, problem-solving, and fundamental questions about the nature, limitations, and ethical use of computers and algorithms. It also explores how creativity tools can be used and as well as providing some insight into how they are implemented. The course is targeted to students with a broad diversity in backgrounds and interests. No prerequisites are assumed beyond high school algebra..

Learning Outcomes:

Upon completion of IMDM 127 you will be able:

- Conceptualize, enter, compile, debug and run interactive programs in a relevant computer language.
- Design and implement software and webpages that generate creative output in images and sounds.
- Use existing creativity support tools to create, edit and store images and 3D shapes.
- Describe and use a systematic process to analyze a computational problem so as to construct a program or algorithm to solve that problem.
- Explain limitations on algorithmic problem solving, and identify some types of problems that are not easily or ultimately solved by algorithm.
- Describe ethical issues in computer use, and analyze case studies of computer use to identify these issues and offer solutions.

Required resources:

A laptop or desktop with the ability to run multimedia packages is strongly recommended, but not required. Students may choose to use appropriately equipped campus labs or loaner laptops. Access to a smartphone or VR headset will not be required, but useful. Assignments will be done in a creative programming system such as Processing for Java.

Any software necessary for this course will either be open source, free for (at least) students, or be available via TERPware (terpware.umd.edu).

Recommended textbooks:

Processing: A Programming Handbook for Visual Designers, Second Edition. Casey Reas and Ben Fry. December 2014, The MIT Press. ISBN-13 : 978-0262028288

Learning Processing, Second Edition: A Beginner's Guide to Programming Images, Animation, and Interaction, Daniel Shiffman. August 2015, Morgan Kaufmann. ISBN-13 : 978-0123944436

Getting Started with p5.js: Making Interactive Graphics in JavaScript and Processing (Make: Technology on Your Time) 1st Edition, Lauren McCarthy, ISBN-10 : 1457186772

Assignment and Grading Policies:

As an introduction to computer science and programming, assignments in this course will include a mix of assignments: some written homeworks to demonstrate mastery of concepts, some programming laboratories in which you create algorithmic art, and exams to demonstrate mastery of concepts and programming elements.

The programming laboratories will include a technical component, where you work on a new element of programming, and an artistic component, where you are encouraged to be creative. Your grade will depend on appropriate use of the technical component as well as a clear effort on the creative component.

Some assignments will be individual while some will be team-based. Final grades will be computed according the following weights.

Individual Assignments: 30%
Team Assignments: 30%
Semester Exams: 20%
Final Exam 20%

Scores will be posted on the course ELMS page as each assignment is graded. Any formal grade disputes or other requests for reassessment must be submitted in writing and within one week of receiving the grade.

Semester grades will be assigned based on the following anticipated ranges. It should be noted that some range cut-offs may be shifted down based upon results obtained during the semester, but they will not be shifted up. The very lower and upper parts of each range will be reserved for any +/- grades.

Range	Grade
90 - 100	A
80 - 89	B
70 - 79	C
60 - 69	D
0 - 59	F

Late assignments will not be accepted for grading so please plan to have work submitted well before the scheduled deadline.

Schedule and assignments

Example assignments:

Develop generative art by writing programs in the Processing environment that can be viewed as a standalone art piece.

Write a program that takes webcam input and modifies in random or systematic ways to create a "digital mirror", or analyzes webcam data for facial position to control output graphics.

Introduce the idea of utilizing math to manipulate images. Apply and explore some basic examples of these via implementing helper methods in GUI apps using Processing. Examples include how to do simple rotations and scaling, and how to posterize color images.

Read about "faked" images and videos in different contexts (eg: magazine covers, news story illustrations, social media posts) and discuss the ethical issues in using such images as well as in creating the tools that can generate such media.

Tentative course schedule:

Session 1

Introduction to creative coding

Day 1

An introduction to the Processing Programming environment, including drawing, animation and interaction. You will write small programs that use the mouse and keyboard to control drawing "tools" for 2D images, with tools that exhibit different behaviors from a simple line to spirals and symmetry.

Algorithmic art

Day 2, 3, and 4

Using Processing to create art by code, harnessing randomness and algorithms to automatically generate artistic patterns, with consideration for artistic principles such as composition and color theory.

Session 2

Animation

Day 5

This unit will animate the patterns, allowing them to evolve over time.

A deep dive into coding

Day 6 and 7

This unit will look in detail at the Java programming language, and the nature of a program including its overall design and object oriented concepts.

Digital media representation and manipulation Day 8 and 9

A look at how images and sounds are presented and stored, and how we can use both algorithms and standard software creativity tools to modify, compress and otherwise manipulate visual and auditory art.

Session 3

Breaking out of 2D

Day 10 and 11

This unit moves beyond 2D images and mouse/keyboard input, and looks at how to create images from 3D data, and how we can interact with our programs using microphones, webcams and other inputs.

Social and technical limitations of computing

Day 12

A look at social implications of digital media, where we ask fundamental questions about the nature, limitations, and ethical use of computers and algorithms

Exhibition

Day 13

NET-Project Portfolio With major projects and Critique

Course participation

Success in any course depends on regular and attentive participation. As this may be your first programming course, being offered in an online environment, we will work through the semester on how you should engage the course material and participate in lectures.

Content delivery and lecture structure:

The technical component of the course is similar to many STEM courses. There are principles and techniques to learn, and these require attention to the details, self-study, and effective practice. We will have recorded lectures, readings, tutorial material, links to online tutorials, and practice exercises for mastery. This will be like learning the keyboard and scales for the piano. We will use live lecture for to review this material and give you an opportunity to ask questions.

However, in this course you'll quickly get to composition and create your own work. This aspect of the course will require exploration and discussion your work with others, getting feedback and critiques, working with fellow students to get new ideas and approaches. Here collaboration will be encouraged, and class periods will be designed for conversation.

There are two 75-minute classes scheduled for each week. We expect to spend the first class period on technical issues, so we review concepts and examples. This will be a how to lecture. The second class period each week we expect to spend in demonstrations, critiques and discussion.

Outside of the standard lecture period we will have pre-recorded videos that explain particular course elements and problems which you can watch on your own. Some of these you'll be encouraged to watch before lecture periods.

Communication:

Email: Please email me at jnsilva@umd.edu. Please do not email me with questions that are easily found in the syllabus or on ELMS (i.e. When is this assignment due? How much is it worth? etc.) but please do reach out about personal, academic, and intellectual concerns/questions. (A direct email is more useful than a Canvas message, so please use email instead.)

ELMS: I will regularly send course info and updates via the ELMS announcements feature. You must make sure that your email & announcement notifications (including changes in assignments and/or due dates) are enabled in ELMS so you do not miss any messages. You are responsible for checking your email and Canvas/ELMS inbox with regular frequency.

Canvas Discussion Board: We will use the Discussion board in Canvas as forum for announcements, for clarifications, and for your questions. The TAs (TBD) and I will check Canvas regularly to answer your questions, and encourage discussion and student answers. If you post code or assignment questions specific to your work please make it private to yourself and instructors.

Personal working environment:

As noted previously, a laptop or desktop with the ability to run multimedia packages is strongly recommended. We will work in this course for you to assemble a working environment of software packages that enable work with digital media by programming and direct manipulation. All will be free to you as either open source, or licensed by the campus. It is not cheating in this class to seek help and advice in setting up software.

Course Policies

University of Maryland Policies:

It is our shared responsibility to know and abide by the University of Maryland's policies that relate to all courses, which include topics like:

- Academic integrity
- Student and instructor conduct
- Accessibility and accommodations
- Attendance and excused absences
- Grades and appeals
- Copyright and intellectual property

Please visit www.ugst.umd.edu/courserelatedpolicies.html for the Office of Undergraduate Studies' full list of campus-wide policies and follow up with me if you have questions.

Academic Integrity

For this course, some of your assignments will be collected via Turnitin on our course ELMS page. I have chosen to use this tool because it can help you improve your scholarly writing and help me verify the integrity of student work. For information about Turnitin, how it works, and the feedback reports you may have access to, visit Turnitin Originality Checker for Students

Copyright:

Course materials are copyrighted and may not be reproduced for anything other than personal use without written permission. But, on the other hand, anything you create for this course is copyright by you. Many of your assignments will lead to creative works that

Accessibility and Disability Services

In this course we will follow the University of Maryland commitment to creating and maintaining a welcoming and inclusive educational, working, and living environment for people of all abilities. Please

contact me with any concerns and issues that you encounter in accessing and using course material. If you have exiting ADS arrangements for accommodation please let me know early in the semester, and if due to the nature of the course an issue arises, please let me know so we can work out the best option for your success.