

Fall 9-1-2018

MART 120.50: Creative Coding I

Michael A. Musick
University of Montana, Missoula

Let us know how access to this document benefits you.

Follow this and additional works at: <https://scholarworks.umt.edu/syllabi>

Recommended Citation

Musick, Michael A., "MART 120.50: Creative Coding I" (2018). *Syllabi*. 8857.
<https://scholarworks.umt.edu/syllabi/8857>

This Syllabus is brought to you for free and open access by the Course Syllabi at ScholarWorks at University of Montana. It has been accepted for inclusion in Syllabi by an authorized administrator of ScholarWorks at University of Montana. For more information, please contact scholarworks@mso.umt.edu.

Syllabus

Creative Coding 1, MART 120

Date modified: 01-21-2018

Table of Contents

- [Course Overview](#)
 - [Professor](#)
 - [Instructor and Lab Assistants](#)
 - [Course Website](#)
 - [Course Requirements](#)
 - [Technology, Tools, and Required Software](#)
 - [Books](#)
 - [Additional Resources](#)
- [Policies](#)
 - [Course Evaluation](#)
 - [Grades %](#)
 - [Final Grades %](#)
 - [Assignment Grades %](#)
 - [Plagiarism & Cheating Policy](#)
 - [Students with Disabilities](#)
 - [Changes to the Course](#)
- [Proposed Schedule](#)

Course Overview .

This course is designed to teach code so that it can become a tool for art and creativity.

The goal is to begin a journey in which the creative and technical possibilities of the computer become a tool in your palate. These tools will become an extension of you, something you think through, express yourself through, and can contribute to the world through. Ultimately, this course is designed to offer you a deeper understanding of the essential possibilities of computation. The course focuses on the fundamentals of programming the computer (variables, conditionals, iteration, functions, and objects) while also touching on a few advanced topics, and relating this all back to the creative and expressive possibilities of media art. The JavaScript-based 'p5.js', programming framework will be utilized within this class. The course will conclude, with your design, development, and implementation of a final project of your choosing.

Professor and Developer

The instructor of record and developer of this course is;

- [Dr. Michael Musick](#)
- E-Mail: michael.musick@umontana.edu.
- Office: McGill Hall, 232.

Teaching Assistants and Lab Assistants

The delivery of this course will primarily be handled by;

- [Stephanie Whitney](#)
- E-Mail: stephanie1.whitney@umconnect.umt.edu
- Office: Media Arts MFA Suite and 127 Computer Lab

They will handle grading and daily operations of the course.

This course will also utilize two additional lab assistants to help you with problem solving. Their information is available from the ["Instructors" Page](#)

Course Websites

- [Creative Coding 1, MART 120 --- Main Website](#)
- [Course GitHub Repo](#) (This git repository holds code examples, and the issues board.)
- [Direct Link to Course Moodle Shell](#)
- [Syllabus Link](#)

Course Requirements

Time

You should plan to spend 3-9 hours of work a week on homework and coding outside of class time (in this case, the time it takes you to read/watch the weeks content). **This is a very intense course that will require a significant time commitment on your part.**

This means you will need excellent time management skills. Schedule time for this course in your week, or you will not be capable of being successful.

Creativity

You are taking this course because it offers the opportunity to learn coding skills through creative means. Please take the time to be creative in your weekly code assignments. Pushing the boundary creatively will make this course more meaningful.

Cleanliness

Coding is hard. But, messy code, documents, and files will make it more difficult. Try to be as clean and tidy as possible in your organization for this class. When writing code, try to keep it organized, and provide many comments. This will make it easier for the grader, the instructor, and you.

Technology, tools, and required software

- **p5.js**
 - The Javascript-based p5.js will be used as the primary development environment.
- **Markdown**
 - All blog and response entries for each homework will be written using the markdown language. This is a simple, light-weight markup language that you will learn in week 2.
- **Editor**
 - You should install a text editor. If you do not already have a text editor installed I suggest;
 - GitHub's [Atom](#).
 - Microsoft's [Visual Studio Code](#)
- **Git**
 - You will need to use Git; a free and open source distributed version control system. This should already be installed on your computer.
 - You can work with git via the terminal or [command line](#).
 - Although it is recommended you use a GUI based front-end application. Such as;
 - [GitHub Desktop](#)
- **GitHub**
 - Additionally, you will be expected to maintain a [github.io](#) repository and website via [GitHub](#) where you will post your weekly homework assignments.
- **Browser**
 - You should ensure you have a modern browser installed on your computer, such as Safari, Google Chrome, or Firefox.
- **Moodle**
 - We will use Moodle for all major communication, as well as submission of homework, and posting of grades.

Books

Required

The following book is required.

- [Getting Started with p5.js](#) by Lauren McCarthy, Casey Reas, and Ben Fry. Published October 2015, Maker Media.

Optional

The following are additional optional resources which may prove useful.

- [Learning Processing](#) is another useful resource for beginning creative coders.
- If you are interested in some supplemental reading on HTML and CSS, The [HTML & CSS book](#) is a nice one.
- Learn the terminal, it is a powerful tool! [Unix for the Beginning Mage](#)

Additional Resources

The following may prove to be useful additional resources for you.

Videos

- [Kadenze course](#)
- [Shiffman Vimeo channel](#) -- no ads
- [Shiffman Youtube playlist](#) -- has ads

Sites and Forums

- [p5.js reference](#) %
- [Intro to Visual Programming with p5.js](#) - online video tutorials (free with signup) %

Policies .

Course Evaluation .

Students work and progress will be assessed through;

- A final project.
- Weekly Homework Assignments
 - These typically will be code-based assignments
 - Code-based assignments will always include a reflection report
 - Weekly assignments may also include or instead be problem sheets that will be completed via Moodle.
- Class participation via group problem solving, support, and forum discussions.

Homework

We will have weekly assignments that are relevant to material from recent content. These assignments are required and you will need to talk about them in your technical reports. It is expected that everyone in the class will create and maintain a github repository for their assignments.

The assignments are your opportunity to put your skills to the test. They're designed to test and deepen your knowledge while giving me valuable information about how you're progressing in the class. [Finished is better than perfect](#). An attempt at them is better than nothing at all.

In addition to the weekly code assignments, you will be expected to also complete a weekly reflection report. In this, you will discuss;

- your weeks progress,
- what did & did not work,
- comments on any frustrations you had with the language,
- and critically analysis/evaluation of how much time was spent learning syntax, structure, & programming concepts vs. actually % programming. %

Homework will be submitted via weekly Moodle forums. You will submit links to your code, sketch, and technical report (both hosted via your GitHub repository) for every coding assignment. Every homework assignment will have an entry in the Moodle Course Page.

Due to the size of the class and the nature of grading code-based homework, you may not receive individual feedback for assignments. Please feel free to send specific queries about homework or reasons for why you received the grade you did to the instructor via email. Grades for all assignments will be based primarily on the student's ability to:

1. Demonstrate an understanding of the specific characteristics and integrative capabilities of the assigned topic in your own words and code.
2. Articulate a clear and concise perspective. Cutting and pasting or copying word for word off the Internet will result in loss of points.
3. Present an organized file/program, as well as technical report; including proper and punctual delivery of the assignment files.
4. Demonstrate creativity beyond the expected technical requirements.

IMPORTANT: Assignments handed in after the due date and time will have points deducted for lateness. This will be in addition to any points deducted for content. Those that are uploaded late but within one week of the due date will lose 10% for lateness. For those uploaded after that, the number of deducted points will be at the discretion of the faculty.

Participation

This class will be participatory, you are expected to participate in discussions and give feedback to other students through on-line participation with the "issues forum" and project submission forums.

Final Project

This course will culminate with final projects. You are expected to push your abilities to produce something that utilizes what you have learned in the class that is useful in some manner to yourself or the world.

The final project will be a self-developed creative work utilizing the principles and ideas developed throughout this course. This project should equal roughly the same amount of work as 3 regular homework assignments. In other words, this is a significant creative piece.

Required: Please note, the final project is required. Failure to complete a final project will result in a letter grade of F for the course. This is regardless of the students current grade standing in the course. There are no exceptions to this policy.

Grades

Final Grades

Grades will be determined according to the following breakdown:

Grade Category	Total %
Regular Assignments	75%
Final Project	15%
Participation	10%

Letters are assigned according to the following final course percentages:

Grade	% Range
A	93-100
A-	90-93
B+	87-90
B	83-87
B-	80-83
C+	77-80
C	73-77
C-	70-73
D	60-70
F	0-59

Assignment Grades

For each project your grade will be assessed upon the following:

- Principles
- Creativity/Thoughtfulness
- Craft.

Below gives you a sense of the spectrum, giving examples for high, middle and low grades.

Please note:

A letter grade of "C" is an average grade. Average is not bad. It means average. Most of us are average, in some way. An average level of work is being done. It does not mean perfection. It means that the work still has room to develop at it's current state. It also means that the

work has noticeable achievements and evidence of some acquisition of knowledge. I look at this on an individual basis as well as across the whole class.

Please read examples of each assessment point below to get an idea of my expectations. Note that a "B" is between Satisfactory and Superior.

- **Principles:**
 - Superior - A - Your work shows evidence and understanding of programming concepts discussed in readings, lectures, and exercises, as you bend them to your will.
 - Satisfactory - C - Your work shows evidence of concepts and is still developing an understanding of course material. You understand some aspects, but aren't utilizing them fully. Usage is cursory. Keep pushing your work and review the material to revisit how you can integrate it to your work.
 - Unacceptable - D/F - Your work shows some evidence of concepts discussed, but lacks key understanding, confidence, robustness and authority. Aspects are lacking. Reviewing course material is required. Ask questions in class. Manage your time better.
- **Creativity, Thoughtfulness:**
 - Superior - A - Your work demonstrates your personality and a great depth of engagement with the material. It's extremely evident that you are thinking, exploring, playing and taking risks. You are creating wonderful experiences.
 - Satisfactory - C - You are executing your ideas, but more time is needed to consider more deeply about what is conceptually and physically happening. They are barely getting off the ground. Or have large bits that are broken. Do more research and exploring. Play.
 - Unacceptable - D/F - Your work is so straightforward that it's flat. It's barely coded (or badly broken). It's copied from elsewhere and not expanded upon. Question and Iterate your work to push your it further. Read. Play. Get off the lame track and get inspired.
- **Craft:**
 - Superior - A - Your work shows delicate care and consideration to presentation and professionalism. Your code is neat, clean, commented and structured. Your friends consider you "Type A." Your style is evident.
 - Satisfactory - C - Your work shows the birth of your ideas, but further time and iteration can really push your work to excel. Your code is there, but messy. It can be simplified and made cleaner. What you put into it, is what you get out of it. Practice makes perfect!
 - Unacceptable - D/F - Your work is rushed and looks like it was done on your train-ride in or the night before. Make your work something you are proud of. You are here to build your portfolio after all, aren't you?

Plagiarism & Cheating Policy

Students are expected to adhere to academic conduct policies of the University of Montana as explained in Section V of your [University of Montana Student Conduct Code](#): "Academic misconduct is subject to an academic penalty by the course instructor and/or a disciplinary sanction by the University. Academic misconduct is defined as all forms of academic dishonesty, including but not limited to: (1) plagiarism, (2) misconduct during an examination or academic exercise, (3) unauthorized possession of examination or other course materials, (4) tampering with course materials, (5) submitting false information, (6) submitting work previously presented in another course, (7) improperly influencing conduct, (8) substituting, or arranging substitution, for another student during an examination or other academic exercise, (9) facilitating academic dishonesty, and (10) Altering transcripts, grades, examinations, or other academically related documents."

Dishonesty will not be tolerated in this course. This includes, but is not limited to, cheating on tests, cheating on assignments, fabricating information or citations, having unauthorized possession of examinations, submitting work of another person or work previously used, or tampering with the academic work of other students.

Plagiarism is the presentation of the work of another without acknowledgement. As defined by the [University of Montana's Student Conduct Code](#), plagiarism is "Representing another person's words, ideas, data, or materials as one's own." Students may use information and ideas expressed by others, but this use must be identified by appropriate referencing.

Students who cheat or plagiarize will receive academic sanctions, which may include an "F" grade on the assignment, examination, and/or in the course. Students will also be reported to the Dean of Students for possible further disciplinary action.

Using Code or Media Found Elsewhere

It is easy to find code and media (i.e. videos, sounds, images, etc.) online. If you use code or media from elsewhere (which you will at times), I expect you to cite the work and author.

If you use found code, you are expected to comment each line, as to what each line does programmatically. *Do not* summarize several lines of code from a high level (i.e., TV Guide). I expect you to comment each line on a granular level. In addition, in these cases, I am also looking for significant modification of the code, for you to enact your own ideas and to experiment heavily. Significant modification means

beyond variable name and value changes. It is bending these concepts to your idea, especially graphically. It is not a copy and paste job. Also, never more than 40% of your code may be supplied from elsewhere. Period. If you use code from online, whether for inspiration, modification or reference, I expect to see a link in your comments from where you got the code and who wrote it. Otherwise it will be considered as plagiarism, and you will fail the assignment. The code must have a reference, along with URL and be commented out LINE BY LINE.

If you use found media, *YOU* are responsible to ensure it is used according to fair-use guidelines. The pieces you make in this course are intended to be portfolio-quality works. Therefore, you should not utilize found media with restrictive use guidelines or licenses. You can read more about various licenses at;

- [opensource.guide](#)
- [choosealicense](#)
- [GNU Licenses](#)
- [Creative Commons Licenses](#)
- [opensource.org](#)

For found media, you are also expected to cite the media in your documentation for the project.

Students with Disabilities

The University of Montana assures equal access to instruction through collaboration between students with disabilities, instructors, and Disability Services for Students. Students with disabilities are encouraged to plan ahead and can contact [Disability Services for Students \(DSS\)](#). If you think you may have a disability adversely affecting your academic performance, and you have not already registered with Disability Services, please contact Disability Services in Lommasson Center 154, or call (406)243-2243. I will work with you and Disability Services to provide an appropriate modification.

Changes to the Course

I reserve the right to change the intended content of this course throughout the semester. This may be done to adjust for the speed of the class, to better meet educational goals, or to account for changes in technology.

Weekly Overview of the Course &

Week 1 &

- Course Overview
- The Syllabus
- What is computational media?
- Your computer as **YOUR** development environment.
- Git Version Control

Week 2

- Basic Technologies
 - Text Documents
 - Text Editors
 - Git
 - Git GUI
 - GitHub.com
 - Markup
 - Markdown

Week 3

- Intro to p5
- "Hello World!"
- Basics of `setup()` & `draw()`
- What is the canvas, and how do you make one?
- Setting the background
- Comments

Week 4

- The basics of drawing
- 2D Primitive Shapes
- Color in p5
- Shape function transformations
- Order of Operations
- The Error Console

Week 5 &

- Basics of animation
- Variables
 - p5 variables '
 - mouseX, mouseY '
 - width, height '
 - Contestants (PI)
 - Defining you own variables
 - Global environment variables
 - Variable scope
- Using variables and math to make things move.
- Simple math
- Statements
- White space and notes on code cleanliness

Week 6

- Math
- console.log()
- The map function (map())
- Random Number Generators
- JavaScript Objects
- Data Types

Week 7

- Conditional Statements
 - Booleans
 - if statements
 - comparison operators
 - else , else if statements
- Bouncing Ball
- Additional Mouse and Keyboard Events
- Buttons, Rollovers, & Switches

Week 8

- Loops
 - For
 - While
 - Nested Loops

Week 9

- Arrays
 - Creating an array
 - length
 - Working through arrays
 - push
 - splice

Week 10

- Functions

Week 11&

- Classes and Objects (OOP)

Week 12

- Interacting Objects

Week 13

Homework Due: {{ site.data.dates[13][1] }} {{site.data.dates.finalProposal}}

- **Introduce the Final**
- Using a Local Server
- Images
- Video
- Sound

Week 14

- **Additional** Info on Submitting your Final
- Work on your Final

Finals &

Final Project Due: {{site.data.dates.Final-Project[1]}}