Microcredentials	bronze	silver	gold	platinum
git				
markdown				
vi				
unix				
python				
sequencing				
conditionals				
booleans				
libraries				
iteration				
abstraction				
graphics library				
style				
debugger				
forum guru				
forum questioner				
notes				
reflection				
enrichment				
writing				
sustainable worker				
attendance				

Homework & Deadlines

The deadline achievements aren't required to get the other homework achievements. They are an extra achievement for being prompt. For homework that involves programming, it won't count towards an achievement unless it runs.

E	09/01: homework1 (by add/drop 09/09 ok)	
E	homework 1	
E	09/08: homework2 (by add/drop 09/09 ok)	
E	homework 2	
E	homework 2 is clear & algorithm works	
E	09/15: homework3	
E	homework 3 (don't skip to 4 without committing!)	
E	test 4 times with different values (commit each time!)	
E	09/22: homework4	
E	homework 4	
E	clear input prompt	
E	09/29: homework5	
E	homework 5	
E	nicely formatted output	
E	10/06: homework6	
E	homework 6	
E	aesthetically pleasing	
E	10/13: homework7	
E	homework 7	
E	aesthetically pleasing	
E	10/20: homework8	
E	homework 8	
E	aesthetically pleasing	
E	10/27: homework9	
E	homework 9	
E	aesthetically pleasing	
E	11/03: homework10	
E	homework 10	
E	Figure out the code from the "problems" section	
E	Encode with random rotation (not fixed) rotation	
E	11/10: homework11	
E	homework 11	
E	aesthetically pleasing	

Individual git repository

C	Create repository	
C	Clone repository	
C	Branch repository	
C	Make first edit to repository	
C	Stage a change to your individual repository	
C	First commit to individual repository	
C	First push to individual repository	
B	Set up shortcut to make future pushing easier	

Individual commits

C	At least one commit message each week	
B	Average at least 5 commit messages per week	
A	Average at least 10 commit messages per week	
Each of the numbers accumulated below is worth 1 achievement		
E	Number of weeks with at least 1 commit	
E	Number of weeks with at least 1 well-written commit message	
E	Number of weeks with at least 5 commits	
E	Number of weeks with at least 5 well-written commit messages	
E	Number of weeks with at least 10 commits	
E	Number of weeks with at least 10 well-written commit messages	
E	Number of weeks with at least 20 commits	
E	Number of weeks with at least 20 well-written commit messages	

Individual project board

C	Create individual project board	
B	Organize individual project board	
B	Break down one assignment into tasks on the project board	
Each of the numbers accumulated below is worth 1 achievement		
E	Number of assignments represented on project board	
E	Number of assignments broken down into multiple tasks	
E	Number of assignments that use markdownfor subtasks	

Team git repository

C	Join team repository	
C	Clone team repository	
C	Branch team repository	
C	Make first edit to team repository	
C	Stage a change to your team repository	
C	First commit to team repository	
C	First push to team repository	
B	Set up shortcut to make future pushing easier	
E	merge different edits to same file with some kind of merge tool	

tools

Using git tools with git doesn't show up on git. To acquire one of these achievements, submit a short reflection on what you did, how you did it, and how you felt about it.		
E	use git with command line (linux/mac)	
E	use git bash (windows only)	
E	use git with stand-alone gui	
E	use git integrated with IDE	
	advanced (these probably won't come up)	
E	create a gitignore file	
E	revert a git mistake	

Use markdown (instead of no formatting) in your README files

E	Make a title	
E	Make a header	
E	Make something bold	
E	Make something italic	
E	Make a bulleted list	
E	Make an unchecked checkbox	
E	Make a checked checkbox	
E	Make a link	

Participation

C	Ask/Answer/Edit on piazza at least 1 time	
B	Ask/Answer/Edit on piazza at least 3 times	
A	Ask/Answer/Edit on piazza at least 5 times	
E	`Number of piazza Esks`	
E	`Number of piazza Answers`	

Office Hours

E	Number of time attended	

Enrichment Discussions

C	Contribute substantively to at least 1	
B	Contribute substantively to at least 2	
A	Contribute substantively to at least 3	
E	Contribute substantively to at least 4	

Write a short reflection on...

E	McLuhan	
E	O'Neil	
E	Technology as Prosthesis	
E	Ten Questions	

Reading

E	book notes, unit 1	
E	book notes, unit 1, before: Sep 1	
E	book notes, unit 2	
E	book notes, unit 2, before: Sep 15	
E	O'Neil notes	
E	O'Neil notes before: Sep 22	
E	book notes, unit 3	
E	book notes, unit 3, before: Sep 29	
E	TenQuestions notes	
E	TenQuestions notes before Oct 6	
E	book notes, appendix	
E	book notes, unit 4	
E	book notes, unit 4, before: Oct 14	
E	McLuhan notes	
E	McLuhan notes before: Oct 20	
E	Prosthesis notes	
E	Prosthesis notes before Nov 3	
E	Book Appendix notes	

Final project proposal

C	Write it	
A	Write it before 11/17	
E	Clear writing style	
E	Thorough but concise	

Showcase

E	Present a poster at Showcase	
E	Poster is clear	
E	Poster is aesthetically pleasing	
E	Presentation is effective	

Writing

Write an essay on an enrichment reading or the topic. If you aren't sure what to write on and want to aim for one of these achievements, use a short reflection (above) to propose a topic

Topic: McLuhan and/or Media as it applies to people creating or engaging with technology

E	Write essay	
E	Clear writing style	
E	Insightful	
E	Engage with other sources	

Topic: O'Neil and/or bias in algorithms/technology

E	Write essay	
E	Clear writing style	
E	Insightful	
E	Engage with other sources	

Topic: Technology as Prosthesis

E	Write essay	
E	Clear writing style	
E	Insightful	
E	Engage with other sources	

Topic: Treat one or more of the 10 questions as the prompt for your paper

E	Write essay	
E	Clear writing style	
E	Insightful	
E	Engage with other sources	

Organization

E	Avoid spaces in all your file/folder names	
E	Give all your files and folders clear-to-other-people names	
E	Use folders appropriately instead of folder-free chaos	
E	Achievements for programs developed in branch (1 each)	

Team Organization

E	Avoid spaces in all your file/folder names	
E	Give all your files and folders clear-to-other-people names	
E	Use folders appropriately instead of folder-free chaos	

Learn keyboard shortcuts

	To receive credit for finding and using keyboard shortcuts, create a file called keyboardShortcuts.md in your git repository and document the shortcuts you find and use in it.	
E	Achievements for each documented shortcut	

	Learn Unix	
	To receive credit for learning unix, create a file called unixCheatSheet.md in your git repository and document the commands you find and use in it. (mac terminal = unix; win git bash = unix)	
	Achievements for each documented command	

	Learn vi (unix text editor)	
	To receive credit for learning vi, create a file called viCheatSheet.md in your git repository and document the commands you find and use in it. If you don't want to use vi, you can use emacs instead, but I use vi.	
E	Achievements for each documented command	

Style

(for achievements, they must be used throughout 1 program, not once in a program)

C	Use clear-to-you variable names	
B	Use clear-to-familiar variable names	
A	Use clear-to-anyone variable names	
C	Use clear-to-you function names	
B	Use clear-to-familiar function names	
A	Use clear-to-anyone function names	
	Number of programs with self-documenting variable names	
	Number of programs with self-documenting function names	
	Refactor at least once	
	Number of programs refactored	

advanced

E	Achievements for programs where docstring is used	
E	Achievements for programs with README.md	
E	Number of times comments are used to explain counterintuitive code that can't be done a clearer way (this probably won't come up)	

Bugs

E	Achievements for new syntax/runtime bugs found	
E	Achievements for new syntax/runtime bugs fixed	
	List new bugs below & on back	

Programming symbols

C	Store a value in a variable	
C	add +	
C	subtract -	
C	multiply *	
C	divide /	
B	increment a variable (+=)	
E	decrement a variable (-=)	
E	use **	
E	use () to force order of operations	

Use built-in python functions:

C	print	
C	input	
C	float	
B	str	
B	open	
B	len	
B	range	
E	Number of other built-in used (list on back)	

Libraries

C	Import graphics library from repository	
B	Import the built-in math library	
A	Import the built-in random library	
B	pow (from math library)	
A	random (from random library)	
E	*# of library functions used (list on back)*	

Sequencing

C	Use a sequence of at least 3 instructions	
B	Use a sequence of at least 5 instructions	
A	Use a sequence of 7-9 instructions	

Conditionals

C	Use if	
B	use else	
A	use elif	

Booleans

E	use >	
E	use <	
E	use ==	
E	use >=	
E	use <=	
E	use and	
E	use or	
E	use not	
E	combine at least 2 of and/or/not	

Abstraction

C	Define a function	
C	Use a function	
B	Define and use at least 3 functions	
A	Define and use at least 5 functions	
E	Number of functions	

Iteration

C	Create a loop	
B	Use while	
B	Use for	
B	Loop over a range	
B	Loop over an iterable (e.g. list)	
A	Use loops at least 4 times	
	Number of loops used	
E	Number of nested loops	
E	Advanced: iteration without explicit loop (deprecated; rarely useful)	

Graphics Library

C	GraphWin	
C	draw	
C	Point	
C	Line	
C	Rectangle	
B	Circle	
B	Oval	
A	Polygon	
A	Text	
E	Entry	
E	Image	
E	get information from a graphics object	
E	change an existing graphics object	
E	colorize a graphics object	

In-Class Activities

No individual activity is attached to a particular grade threshold because attending specific classes isn't required. Things might come up that make you miss class & you have a certain amount of headroom with which to manage such absences.

To the right is an estimated number of activities you need to accomplish for each grade level. This is an estimation based on projected in-class activities & might change if things shake out differently in class. Specific activities will be added below as you do them. There may be occasional individual in-class activities, which can be made up. By their nature, missed team activities can't be made up.

C	29
B	34
A	38