EECS 1710
Programming for Digital Media

Dr. Matthew Kyan
Dept. of Electrical Engineering & Computer Science
Fall 2022

YORK UNIVERSITÉ UNIVERSITY

EECS 1710
Programming for Digital Media

Week 1 :: Course Introduction

YORK UNIVERSITÉ UNIVERSITY

Week 1 - Course Introduction

- Lecture 1:
  - Motivations
  - Administrative
  - Syllabus/Themes
  - Assessment

- Lecture 2:
  - Lab Accounts
  - Tools & Resources
  - Getting Setup (walkthrough)

Contact

- Instructor

Dr. Matthew Kyan
Associate Professor
Dept. of Electrical Engineering & Computer Science

Virtual/Actual Office: via ZOOM (office hours: Wednesday 3pm-5pm)

>> **click here** to access during office hours (also linked on eClass)

>> otherwise by appointment → via email: **mkyan (at) yorku (dot) ca**

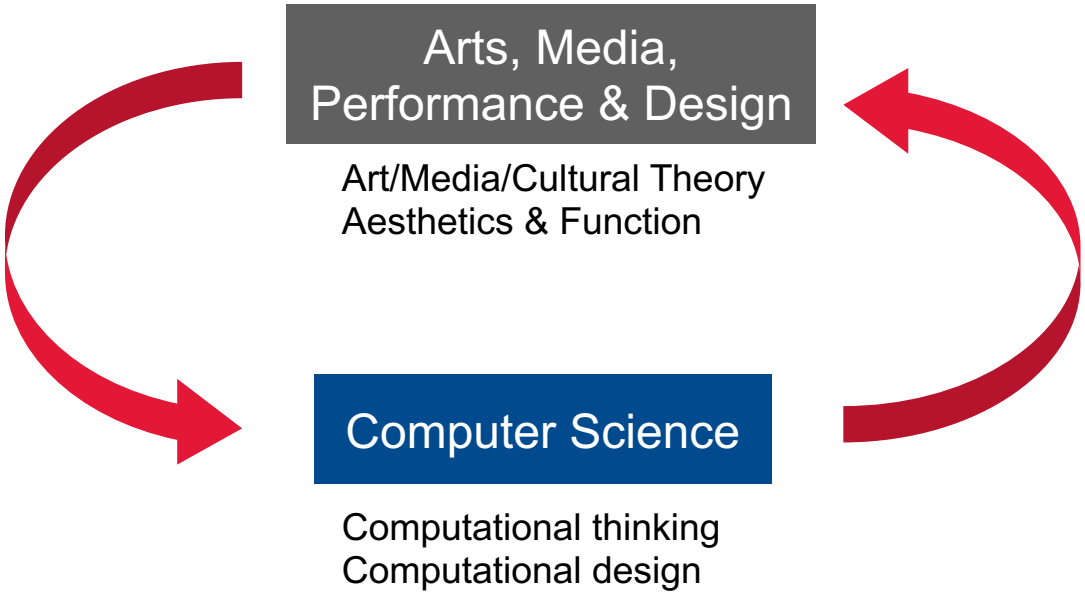
*IMPORTANT: please use "**EECS1710_F2022**" in the subject when emailing*

- TA's
  - TBD (info will be posted on eClass)

YORK UNIVERSITÉ UNIVERSITY

Digital Media @ YorkU

- Provides a foundation in the **computational basis** for the creation of digital media
  - imagery and sound, animation, simulation, 2D/3D environments
  - physical & tangible computing
  - human computer interaction



YORK UNIVERSITÉ UNIVERSITY

Digital Media @ YorkU

- Explores theoretical, artistic and experiential ideas
  - underpinning aesthetic & functional aspects of digital media
- Engages in the practice of creating digital media works
  - to investigate ways in which culture is (or can be) produced through technology and its subsequent effects on society

YORK UNIVERSITÉ UNIVERSITY

Computational Basis ??

Science of computation (computer science)

- Study of *process* = **how** we do things
  - how we **store/represent** the stuff we want to process
  - how we specify **what** we do with that stuff
    - what steps are involved?

Consider & describe the steps needed to …

Brush your teeth

Make a sandwich

Make & bake cookies

YORK UNIVERSITÉ UNIVERSITY

Consider & describe the steps needed to …

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

generate/solve a Sudoku puzzle?

YORK UNIVERSITÉ UNIVERSITY

Computer Programming

≈ the study of “recipes”

- recipes to **model/simulate** real/imagined world environments/events..
- recipes to directly **sense/control/respond** to real/imagined world environments/events

- Recipes have aspects which are **flexible/variable** …

  (e.g. amount of ingredients)

- … and aspects which are quite **fixed/restrictive**

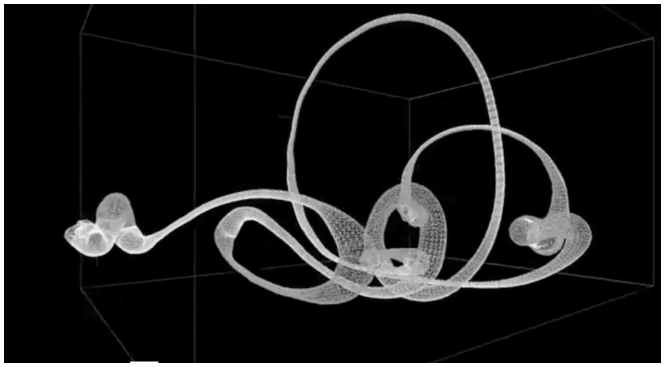
  (e.g. type of some ingredients, order & number of steps, etc.)

- ultimately they serve as a **template** for a *process* in which: certain ingredients are first **encoded**, then **transformed** into a result)

”computing recipes/programs”…

- Can be specified via a multitude of different “languages”
  - Max, Java, Processing, Javascript, Python, C/C++, C#, Objective C, and many many more!!
- though their syntax/grammar can differ..
  - *representations* and especially *mechanisms* we use to simulate/control/create are quite similar across languages!

some inspiration ...
sample generative & physical artwork reels



OpenFrameworks (C/C++): https://vimeo.com/74124094

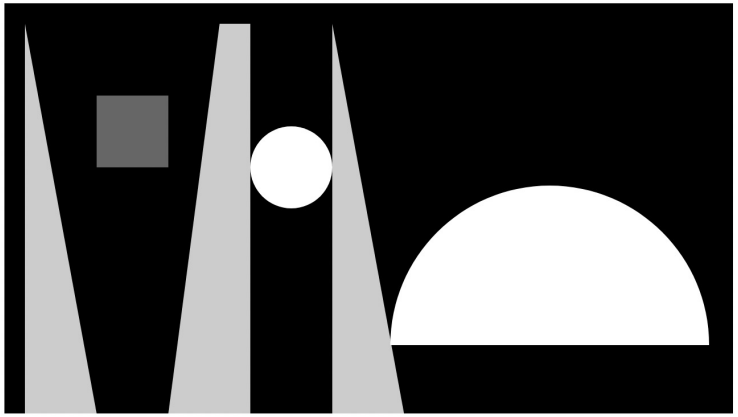
Processing: https://www.youtube.com/watch?v=_qeq6w0C5UM

Java/Javascript/Similar: https://www.youtube.com/watch?v=Z9NLxrkMWM4

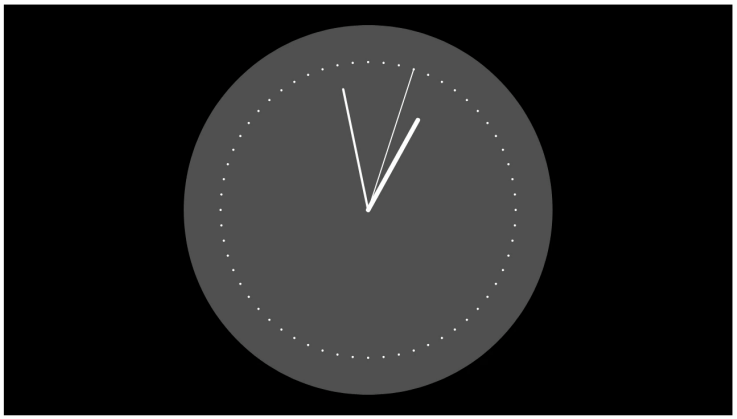
Unity 3D + various : https://www.youtube.com/watch?v=4DEvr43s2xs

YORK UNIVERSITÉ UNIVERSITY

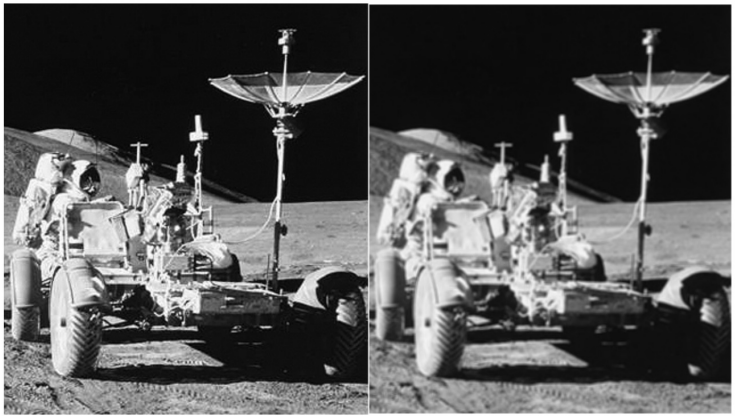
Consider & describe the steps needed to …



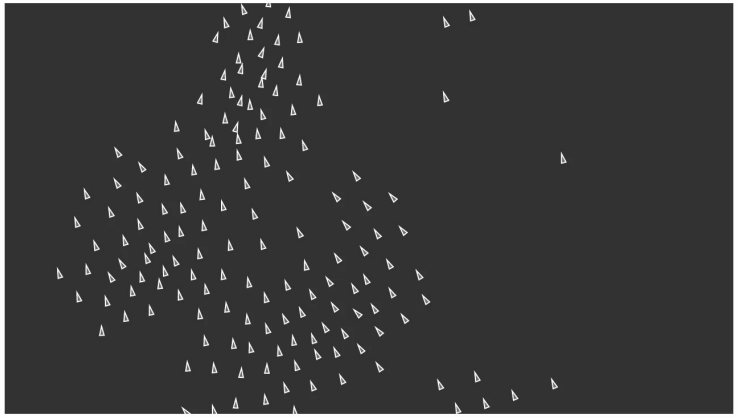
draw simple shapes



create/visualize a clock



Blur an image



simulate flocks

https://processing.org/examples

YORK UNIVERSITÉ UNIVERSITY

Media Computation :: why study computation?

- Computers allow us to easily:
  - digitize/store, but most importantly, deconstruct, manipulate, & reconstruct media
- Digitizing also makes it easier to communicate:
  - Securely send media over large distances (e.g. streaming parcels over the air or internet)
- However, in our discipline (digital media):
  - it allows us to be **more** than just content authors
  - allows us to get “under-the-hood” and understand at a fundamental level, how to construct/deconstruct and refashion media at the source!
- BONUS:
  - studying computers is **empowering** almost a necessary literacy in today’s world (useful for any industry)

YORK UNIVERSITÉ UNIVERSITY

Digital Media Pathways @ York

- DMD (Digital Media Developer)
  - Primarily concerned with building digital media tools
- DMA (Digital Media Arts)
  - Using computational tools to construct artworks, enhance performance, etc.
- DMGA (Digital Media Game Arts)
  - More specific engagement with media for games
  - Or use of game technology for artistic expression (even applications of game tech in non-gaming scenarios)

YORK UNIVERSITÉ UNIVERSITY

Central themes explored (EECS intro courses)

- EECS 1710 & 1720
  - Programming from ***Client*s** point of view
  - 1710 – basic constructs and concepts of programming
  - 1720 – assembly of more complex interactive programs
- EECS 2030
  - Programming from ***Implementers*** point of view
  - (more challenging – more related to tool development)

Recipe Analogy

Comparing a computer program to a food recipe

Food Recipe	Computer Program
• a *chef* writes a set of instructions called a *recipe*	• a *programmer* writes a set of instructions called a *program*
• the recipe requires specific *ingredients*	• the program requires specific *inputs*
• the *cook* follows the instructions step-by-step	• the *computer* follows the instructions step-by-step
• the *food* will vary depending on the *amount of ingredients* and the *cook*	• the *output* will vary depending on the *values of the inputs* and the *computer*

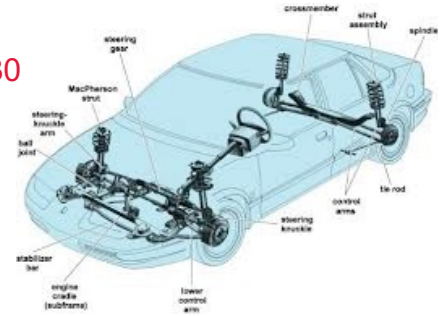
Client vs Implementer (analogy)

- CAR ≈ Program

EECS17xx view

program (recipe)

EECS2030 view



CLIENT

This view creates programs by assembling or 'using' different (implemented) pieces, working with each via their available interface.

IMPLEMENTOR

This view creates program internals from scratch (or assembly), then defines an interface that can serve the needs of possible clients (hiding inner workings from client).

YORK UNIVERSITÉ UNIVERSITY

Course Description (extended)

- Introduction to program design and implementation focusing on digital media elements, and focusing on *CLIENT* perspective
  - sound, images, and animation; algorithms, simple data structures, control structures, and debugging techniques
- Lays the conceptual foundation for the development and implementation of Digital Media artefacts

Course Text

- **NO OFFICIAL TEXTBOOK !!**

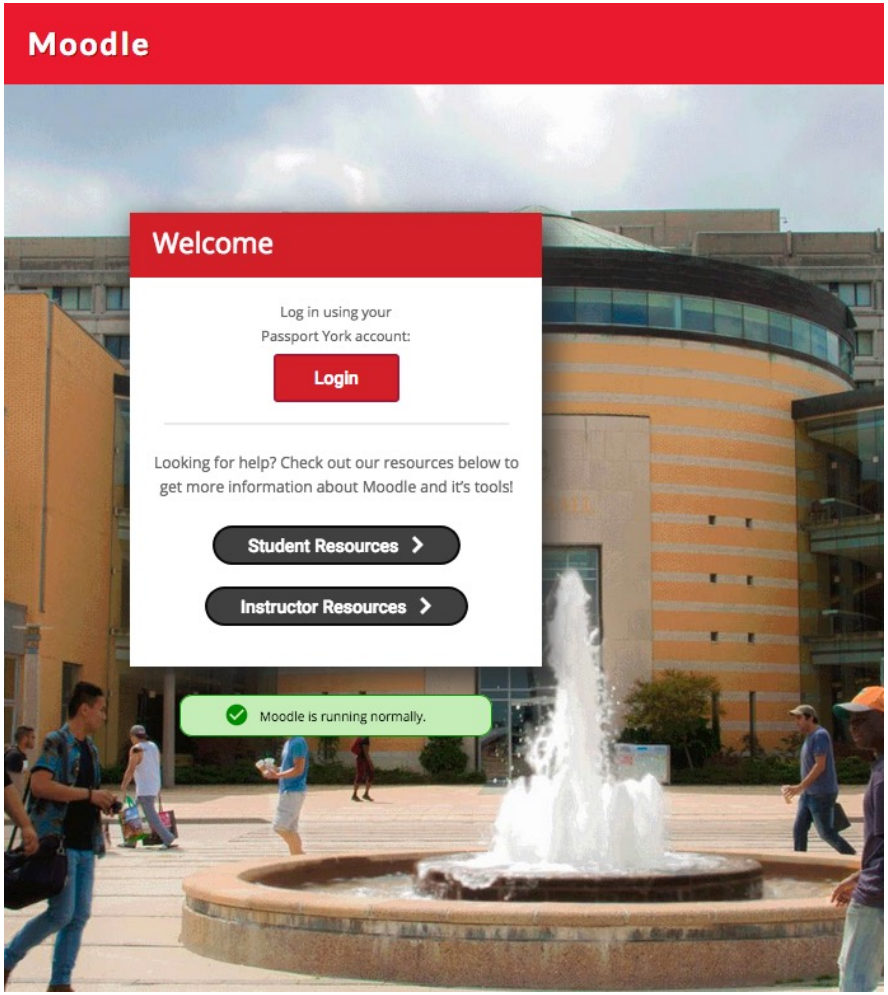
- Useful Reference Texts
  - Shiffman, Daniel. *Learning Processing: A Beginner's Guide to Programming Images, Animation, and Interaction*. 2nd ed., Morgan Kaufmann, 2015.
  - Shiffman, Daniel. *The Nature of Code: Simulating Natural Systems with Processing*. 2012.
  - Levin, Golan, and Tega Brain. *Code as Creative Medium: A Handbook for Computational Art and Design.* MIT Press, 2021.
  - Gonzalez Vivo, Patricio, and Jen Lowe. *The Book of Shaders.* 2015.

  - M. Guzdial and B. Eriscon, "*Introduction to Computing & Programming with JAVA - A Multimedia Approach*", ISBN 0-13-149698-0

- Useful Reference Texts & tutorial links (will be posted on website)
- Selected Readings
- Website (eClass)

YORK UNIVERSITÉ UNIVERSITY

eClass ??



Course is here!

https://eclass.yorku.ca/

Course Schedule

Week	Topics	Dates	Activity
1	Course Introduction; Tools & Setup	Sep 7 – Sep 9	No Lab
2	Statements, Flow & Data Basic sketch, tracing, errors	Sep 12 – Sep 16	Lab 0 exercises
3	Operators, Expressions & Methods	Sep 19 – Sep 23	Lab 1 exercises
4	Relations, Conditionals & Branching	Sep 26 – Sep 30	Lab 2 exercises
5	Looping & Repetition	Oct 3 – Oct 7	Midterm (20%) (in sched. lec)
	FALL READING DAYS	Oct 10 – Oct 14	
6	Working with Arrays	Oct 17- Oct 21	Lab test 1 (10%) (in sched. lab)
7	Working with Objects	Oct 24 – Oct 28	Lab 4 exercises
8	Working with Sound	Oct 31 – Nov 4	Lab 5 exercises
9	Working with Images	Nov 7 – Nov 11	Lab 6 exercises
10	Working with Video	Nov 14 – Nov 18	Lab test 2 (10%) (in sched. lab)
11	Working with Networks	Nov 21 – Nov 25	Lab – Project Work
12	Selected Topics, Project Q&A	Nov 28 – Dec 3	Lab – Project Work
13	Course Review	Dec 5	Project (20%)
	Exam Period	Dec 6 – Dec 21	Final (30%)

Important Dates

https://registrar.yorku.ca/enrol/dates/2022-2023/fall-winter

Add/Drop Deadlines

	FALL (TERM F)	YEAR (TERM Y)	WINTER (TERM W)
Last date to add a course **without permission** of instructor (also see Financial Deadlines)	Sept. 20	Sept. 20	Jan. 22
Last date to add a course **with permission** of instructor (also see Financial Deadlines)	Oct. 4	Oct. 25	Feb. 6
Drop deadline: Last date to drop a course without receiving a grade (also see Financial Deadlines)	Nov. 11	Feb. 10	March 17
Course Withdrawal Period (withdraw from a course and receive a grade of "W" on transcript – see note below)	Nov. 12 - Dec. 7	Feb. 11 - April 11	March 18 - April 11

YORK UNIVERSITÉ UNIVERSITY

Course Evaluation

Assessment	Weight	When (tentative schedule)
Labs (6 total)	15%	Weeks 1-10 (during regular lab time - due 1week following) Labs 1-3 (worth 2% each), Labs 4-6 (worth 3% each)
Lab Tests	20%	Labtest 1 (week 6 - 10%), Labtest 2 (week 10 - 10%)
Project	20%	Oct 7 - Dec 6, 2022 (dedicated lab time in weeks 11-12)
Midterm	20%	Oct 5, 2022 (in sched. lec.)
Final Exam	25%	TBD by the Registrar's Office (will be posted on eClass)

Yet to be scheduled (will be announced on course website)

YORK UNIVERSITÉ UNIVERSITY

Labs

- Practice, make, practice, make, learn … (DATT 1000)
- Practice, break, practice, break, learn … (EECS 1710)

- Lab 0 – starts week 2 (see course syllabus & calendar)

…

7 labs in total (labs 0-6)

Labs 1-6 will be formally submitted (15%)

Lab 1-3 (2%); Lab 4-6 (3% each)

Lab Tests

- Lockdown situation (1 person per computer)
- Time limited programming task, closed book

- 2 tests (will emphasize basic programming elements)
- Will supply practice tests, and walkthrough solutions

Project

- Last few weeks of lab time will be dedicated to working on a project (groups)
- Will allow 2-3 per group

- Task information will be posted around READING WEEK (so you can start early)
- Task will explore creative use of techniques and methods learned throughout the course
- Will assess aesthetic + functional aspects

Academic Integrity

What is Plagiarism?

Plagiarism is representing someone else's ideas, writing or other intellectual property as your own, and is another form of academic dishonesty.

Any use of the work of others, whether published, unpublished or posted electronically (e.g., on web sites), attributed or anonymous, must include proper acknowledgement.

You can find full definitions of plagiarism and other forms of conduct that are regarded as serious academic offences in York's Senate Policy on Academic Honesty.

Common Types of Plagiarism

In doubt? ASK!!

Plagiarism can take many forms. Some of the most common types of plagiarism include[1]:

- Downloading or buying research papers (Downloading a free paper from a web site or paying to download a paper and submitting it as your own work)
- Copying and Pasting (copying and pasting portions of text from online journal articles or websites without proper citation)
- Copying or submitting someone else's work (copying a paper/lab report/formula/design/computer code/music/choreography/assignment etc. and submitting it as your own work)

[1] Harris, R. A. (2002). *The plagiarism handbook: Strategies for preventing, detecting, and dealing with plagiarism.* Los Angeles: Pyrczak Publishing, p. 13.

Important: You must use York's standards when submitting your work even if you were taught to document your sources differently in the past.

TUTORIAL http://www.yorku.ca/tutorial/academic_integrity/plagdef.html
POLICIES http://secretariat-policies.info.yorku.ca/policies/academic-honesty-senate-policy-on/

YORK UNIVERSITÉ UNIVERSITY

Software/Tools

- Processing ←→ Java (programming language)
- Eclipse (Integrated development environment - IDE)
- Linux Environment (operating system - OS)

- (at YorkU) > LAS & WSC labs
- (at home) > remote lab access - linux (Rocky Linux)

Tutorials will be posted on course website

YORK UNIVERSITÉ UNIVERSITY

More on Friday!

- EECS Accounts
- Remote Access labs
- Walkthrough of tools
  - IDE's – integrated development environments
  - Processing (PDE)
  - Java (Eclipse)
  - How do these relate to other languages/courses later??
  - Will open and run some quick examples ☺

General Course Advice

- Make USE of available resources:
  - Academic advising services (every program of study has them)
    - what courses should I take? should I drop some courses?
    - should I change my program of study? I am unhappy with my program but what other options do I have?
  - Study supports (from your college, from your faculty)
    - http://bethune.yorku.ca/ , http://winters.yorku.ca
    - study groups, extra help
  - Learning Skills Services
    - http://lss.info.yorku.ca/
  - Study skills workshops, on-line resources, etc.

YORK UNIVERSITÉ UNIVERSITY

Your EECS System Account

- You require an EECS account to complete this course.
  - marked course material (submitting code for labs/projects)
  - course announcements

- Activate your account:
  - http://www.eecs.yorku.ca/activ8
  - This account can be used to access any of the PRISM Labs, such as LAS1002, LAS1004, LAS1006, WSC and others
  - Linux and Windows (machines can boot into either OS)

- disk quota, web space, print quota

Checklist – for next lecture

- get an EECS account (if you don't have it already)
  - warning!! it can take up to 24 hours to get the account. Plan ahead.

- I will run a tools session in the next lecture
  - (Friday Sept 9)
  - Will review how to get setup to code!!

- Get Familiar with eClass website, and Lecture Notes section