



EECS 1720

Building Interactive Systems

Dr. Matthew Kyan

Dept. of Electrical Engineering & Computer Science

Winter 2023



EECS 1720

Building Interactive Systems

Lecture 1 :: Course Introduction

Contact

- Instructor

Dr. Matthew Kyan

Associate Professor

Dept. of Electrical Engineering & Computer Science



Virtual/Actual Office: in-person+ZOOM (office hours: TBD)

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

- TA's

- TBD (info will be posted on eClass)

IMPORTANT: email protocol

- If you need to email me about anything related to the course, *please use “**EECS1720_W2023**” in the subject when emailing*

ZOOM LINKS

- Must use yu-passport login to access
 - login to eclass first, then access zoom links there, or
 - be logged into yu-passport then open the following zoom links
- **Lectures**
 - All lectures are streamed/recorded
 - Tuesday 2:30-4pm (in-person VH-A => will also stream)
 - Thursday 2:30-4pm (remote/zoom/hybrid => stream only **)
 - EECS1720 - W2023 - LECT
 - <https://yorku.zoom.us/j/98286340595>
 - Meeting ID: 982 8634 0595

*** Thursday lecture will be remote only (no in-person lecture in LAS-B)*

ZOOM LINKS

- **Office Hours**

- Office hours will be held immediately following the lab session
- Location: WSC 105, Wednesday's 2-3pm **
- EECS1720 - W2023 – OFFICE HOURS
- <https://yorku.zoom.us/j/96128552701>
- Meeting ID: 961 2855 2701

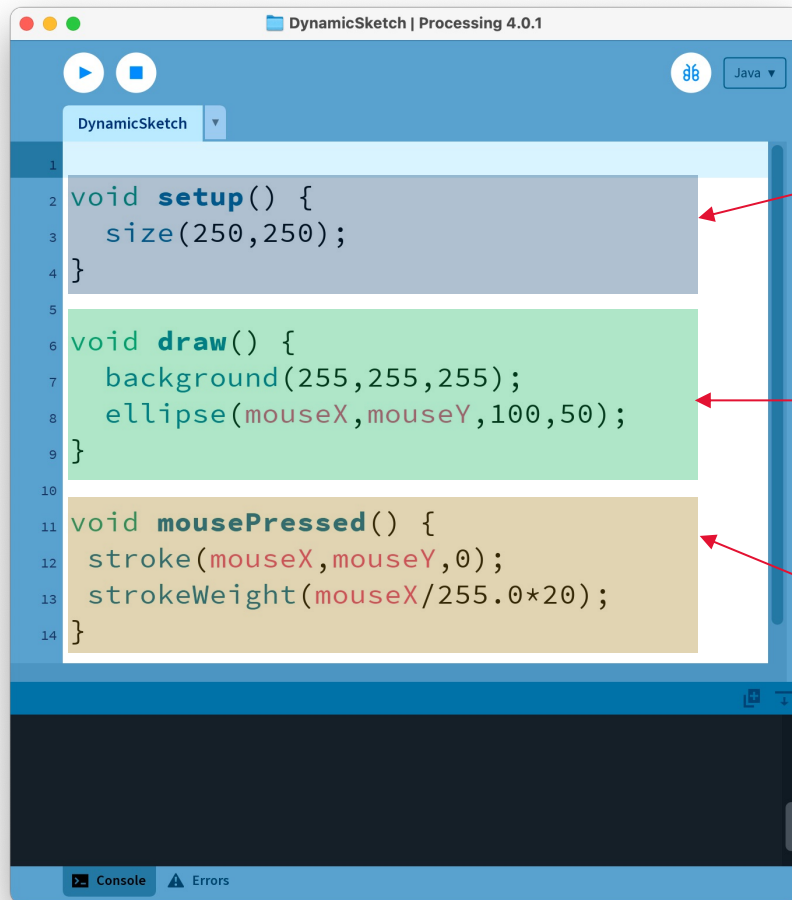
*** Additional office hours can be made by appointment*

Course Description

- A second course teaching more advanced programming concepts within the context of image, sound and interaction using an object-oriented language;
- Introduction to interactive systems, user interfaces, event-driven programming, object design and inheritance;
- Implementation using debuggers, integrated development environments, user interface builders.
- Pre-requisites: LE/EECS 1710 3.00. Course credit exclusions: LE/EECS 1020 3.00, LE/EECS 1022 3.00, AP/ITEC 1620 3.00. PRIOR TO FALL 2014: course credit exclusions: LE/CSE 1020 3.00. PRIOR TO SUMMER 2013: course credit exclusions: SC/CSE 1020 3.00.

Interaction in Processing (dynamic/events)

Processing anatomies...



blocks of statements

Block1
{
 // setup statements
}

Block2
{
 // draw statements
}

BlockX
{
 // other statements
}

flow

Block1 runs once

Block2 repeats forever

BlockX if event occurs interrupt Block2 run BlockX once resume Block2

In standard Java program:

Think about a while loop (menu interaction)

```
boolean done = false;
while (!done) {
    draw();

    // wait frameRate

    if (exitCondition)
        done = true;
}
```

- What are the limitations of this approach?
 - Cannot interrupt this loop (need event mechanism)
 - Cannot run other things in background (need threads)

Main Motivation

- to understand & develop **event-driven** java programs
 - Understand classes and objects in more depth
 - Understand how classes support instantiation of objects
 - Understand how associations & hierarchies of classes work together
 - Understand how Exceptions work (analogous to Events)
 - Understand how to trigger and handle Exceptions
 - Understand how Events work
 - Understand different kinds of events possible in Java programs
 - Understand how to trigger and handle Events effectively to support user interaction
 - Understand basic Threads
 - Understanding how to have Java programs trigger and handle events more autonomously (animation, etc)

Examples (video games)



<https://www.youtube.com/watch?v=UZIEXI9xgR8>

Other classic video games from the 70's:

https://www.youtube.com/watch?v=km8bc_oBRgk

Examples (interactive art exhibits)



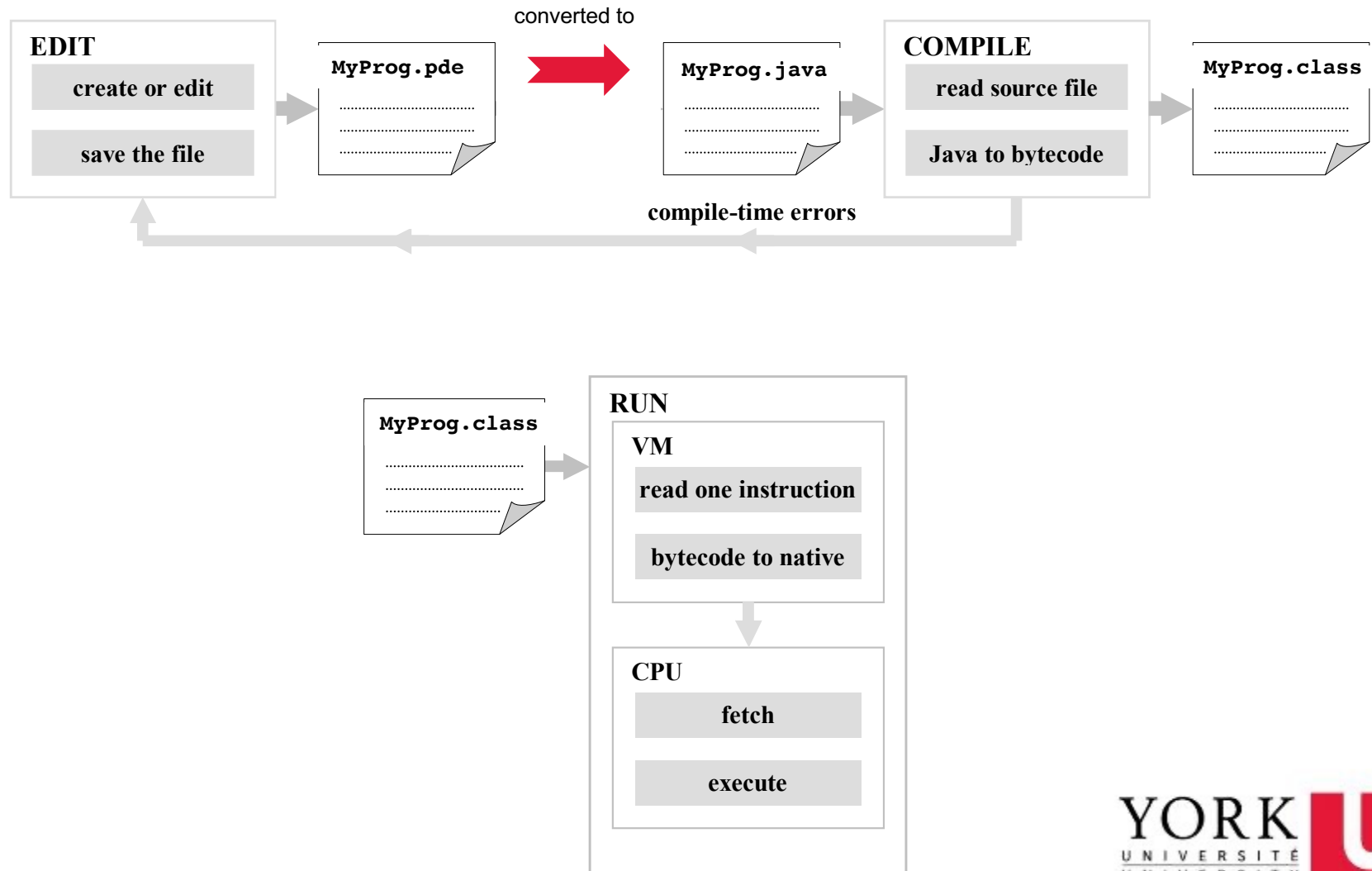
<https://www.youtube.com/watch?v=ISzlk6NeDLk>



A couple of notes about pure Java (vs. Processing)

- In Java, every file is usually associated with a single class
 - It is possible to have classes within classes (for special circumstances, but generally only one class per file, and the class has to have the same name as the file)
- In Java, a project can have multiple files (i.e. multiple classes), but there must exist a class with a “main” method
 - This “main” method is always the entry (start) point of the program when run
 - If tracing a program, we always start our tracing from there

defining & executing a Processing sketch



Source code

DynamicSketch.pde



```
1 void setup() {  
2   // statements to run once at start  
3   print("starting now..");  
4   size(250,250);  
5 }  
6  
7  
8 void draw() {  
9   // statements to run each time  
10  // the screen refreshes  
11  background(255,255,255);  
12  ellipse(mouseX,mouseY,100,50);  
13 }  
14  
15
```

starting now..

Processing ~ Java (lite)

DynamicSketch.java

```
import processing.core.PApplet;  
  
public class DynamicSketch extends PApplet {  
  
    public void settings() {  
        print("starting now..");  
        size(250,250);  
    }  
  
    public void draw() {  
        background(255,255,255);  
        ellipse(mouseX,mouseY,100,50);  
    }  
  
    public static void main(String[] args) {  
        String[] processingArgs = {"HelloSketch"};  
        DynamicSketch mySketch = new DynamicSketch();  
        PApplet.runSketch(processingArgs, mySketch);  
    }  
}
```

Java (full) – note the extra scaffolding!

DynamicSketch.class
(Byte Code)

Converted Processing Program

- Example from labtest recoveries (EECS1710 last semester) – anonymized...
 - Demo (example from labtest last semester – recovered *.java file after conversion from processing *.pde)

Example shows much more scaffolding in a "pure" java program

Course Details

Course Title: EECS 1720: Building Interactive Systems

Term: Winter 2023

Lectures: Tuesday, Thursday: 2.30pm – 4:00pm (hybrid delivery)

Location:

Tuesday (VH-A Vari Hall Lecture Theatre A) – in person

Thursday (ZOOM session – no lecture in LAS-B)

Zoom link:

EECS1720 - W2023 - LECT

<https://yorku.zoom.us/j/98286340595>

Meeting ID: 982 8634 0595

Labs: Wednesday: 12.30pm-2.00pm
(supervised labs begin Week 2, lab0 familiarity session will be run in Week 1)

Term Dates: Classes: Jan 10, 2023 – Apr 9, 2023
Study Day: Apr 10, 2023
Exam Period: starts April 11)

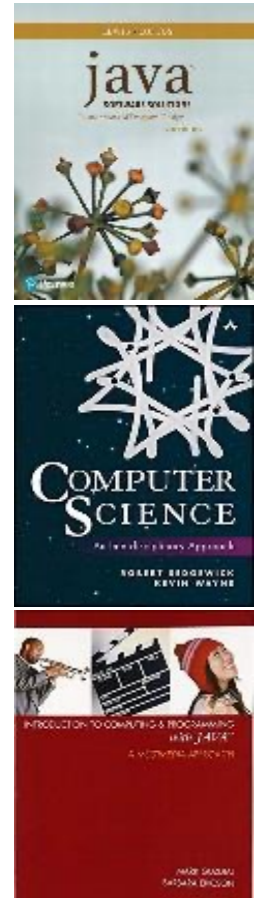
Course Website: Hosted on eClass (<https://elcass.yorku.ca/>)

Course Text

- **NO OFFICIAL TEXTBOOK !!**

- Useful Reference Texts

- **Head First Java: A Brain-Friendly Guide**, Kathy Sierra, Bert bates. ISBN-13: 978-0596009205.
 - This book is different from any other books that is used for academic purpose. It does not have a formal structure that you normally see in other books, which makes it less boring to continue reading the book. I'm sure you do not fall sleep, reading this book. Please beware that not all the concepts that is taught, can be found in this book.
- **Introduction to Programming in Java**: Robert Sedgewick and Kevin Wayne. Second Edition. ISBN-13: 978-0672337840.
 - This is an excellent book, with lots of good examples.
- **Introduction to Computing & Programming with JAVA – A Multimedia Approach**. M. Guzdial and B. Eriscon,, ISBN 0-13-149698-0.
 - This is an excellent book that focuses on digital media concepts with Java.
- **Absolute Java**: Walter Savitch. Sixth Edition. ISBN-13: 978-0134041674.
 - This is an almost comprehensive book for this course however chapter 7 onward is what is useful for this course.
- **Computer Science: An Interdisciplinary Approach**: Robert Sedgewick and Kevin Wayne. ISBN-13: 978-0134076423.
 - This book does is mostly about the science of computing. If you are perfectly comfortable with the materials taught in this course and you want to know more, this book is a good reference for you.



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

Course Evaluation

Item	Weight (% of final grade)	Due Date
Lab 0	0	(See course calendar & syllabus for dates)
Labs 1-3	15	
Labs 4-6	15	
Assignment 1	5	
Assignment 2	5	
Programming Test 1	15 (10% prog, 5% theory)	
Programming Test 2	15 (10% prog, 5% theory)	
EndTerm	30 (theory)	Last week lab session
TOTAL	100	

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

Topics:

- Classes & Objects
- Exceptions
- Aggregation & Composition
- Classes & Inheritance

- User Interfaces (UIs)
 - UI Elements & User Interface Builders

- Event driven programming
 - Handling UI Interactions
 - Other Events

- Interactive systems
 - Basics of Threads
 - Animation & Interaction

Labs

- Week 1 – (work through on your own)
 - **Ensure you have an EECS account (and can log in!!!)**
 - Register if NOT!
 - Lab 0 (not graded)
 - Video series on github
 - Check videos on echo360 (preliminaries) for Eclipse Installation and accessing RemoteLab
 - Walkthrough in the lab (video will be posted)
 - Basic info on getting up and running with Eclipse, compiling/running from terminal, & importing a java project
 - Basic practice exercises (get familiar with
- Week 2 – (next week => lab1)

6 labs in total (Lab 0 not assessed)

Labs 1-6 will be formally submitted (30%)

Labs 1-3 (15%) are paired with assignment 1 (5%) and labtest 1 (15%)

Labs 4-6 (15%) are paired with assignment 2 (5%) and labtest 2 (15%)

Assignments

- Like a lab, but question is a little more design oriented (open-ended)
- You will have a couple of weeks to finish these
- Submission dates will be available on calendar & in syllabus (on eclass)

Labtests

- Similar to EECS1710
 - i.e. edit/complete source code and submit
 - Additionally, there will be some text file short answers
 - 10% question (1 or 2 questions to complete) – programming
 - 5% simple short answer question(s) - theory
 - Complete in Eclipse Editor (use submit command)
 - A little more controlled than via Processing editor
 - NEED TO MAKE SURE YOU ARE ON TOP OF SUBMIT PROCESS!!! (see lab0)

Important Dates



EVENT	FALL (TERM F)	YEAR (TERM Y)	WINTER (TERM W)
Classes start	Sept. 7	Sept. 7	Jan. 9
Last date to announce components of final grades	Sept. 20	Sept. 20	Jan. 23
Fall Reading Week ¹	Oct. 8-14	Oct. 8-14	
Last date to submit Fall term work	Dec. 6	Dec. 6	
Fall classes end	Dec. 6	Dec. 6	
Fall Study Day ²	Dec. 7	Dec. 7	
Fall examinations ³	Dec. 8-23	Dec. 8-23	
Winter Reading Week ¹		Feb. 18-24	Feb. 18-24
Last date to submit Winter term work		April 8	April 8
Winter classes end ⁴		April 10	April 10
Winter Study Days ²		April 11	April 11
Winter examinations ⁵		April 12-27	April 12-27
Notes		Virtual Friday required due to Good Friday: Monday, April 10	Virtual Friday required due to Good Friday: Monday, April 10

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

Important Dates



✓ 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 10	March 18 - April 10

* Add and Drop Deadline Information

There are deadlines for adding and dropping courses, both academic and financial. Since, for the most part, the dates are **different**, be sure to read the information carefully so that you understand the differences between the sessional dates below and the [Refund Tables](#).

You are strongly advised to pay close attention to the "Last date to enrol without permission of course instructor" deadlines. These deadlines represent the last date students have unrestricted access to the registration and enrolment system.

After that date, you must contact the professor/department offering the course to arrange permission.

You can drop courses using the registration and enrolment system up until the last date to drop a course without receiving a grade (drop deadline).

You may **withdraw from a course** using the registration and enrolment system after the drop deadline until the last day of class for the term associated with the course. When you withdraw from a course, the course remains on your transcript without a grade and is notated as 'W'. The withdrawal will not affect your grade point average or count towards the credits required for your degree.

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 include¹:

- 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.

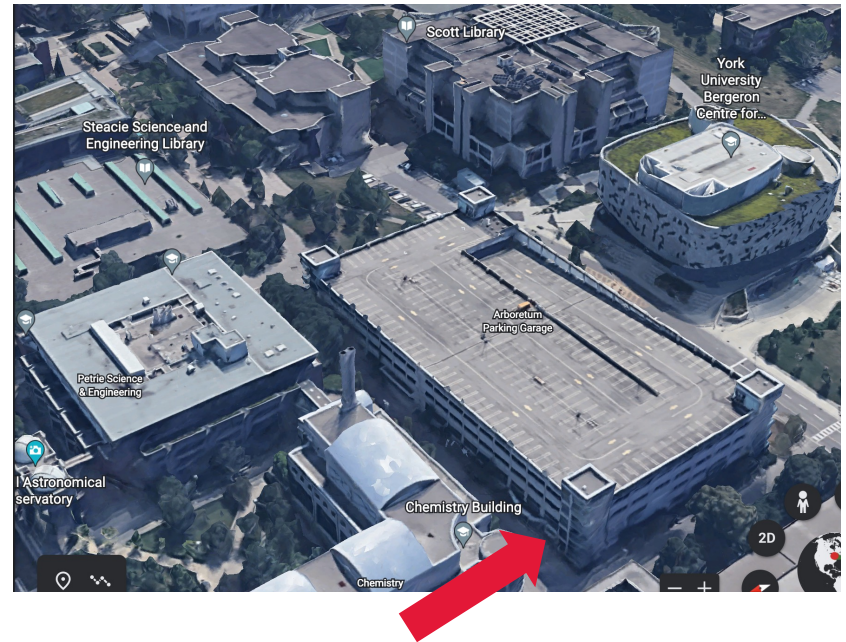
Important: You must use York's standards when submitting your work even if you were taught to document your sources differently in the past.

TUTORIAL http://www.yorku.ca/tutorial/academic_integrity/plagdef.html

POLICIES <http://secretariat-policies.info.yorku.ca/policies/academic-honesty-senate-policy-on/>

Note on lab locations:

- All labs will be run out of WSC 105, 106, 108.
- If you have LAS 1002/1002B on your schedule, please do not go to these labs (come to WSC instead for all labs). Enter by walking past Tim Hortons, down steps and to right
 - <https://maps.info.yorku.ca/2010/06/william-small-centre/>



- Students with accommodations will be using LAS1002/1002B for labtests only

Software/Tools



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



- (at home) > virtual box + image of linux (CentOS)

tutorial posted on
course website

Resources

- Tutorials (will be linked/posted on course website)
- Java API
<http://docs.oracle.com/javase/8/docs/api/>
- Java Development Kit (JDK) (ver 8+)
<http://www.oracle.com/technetwork/java/javase/downloads/index.html>
- Eclipse
<https://eclipse.org/downloads/>
- Oracle's Java Tutorials
<http://docs.oracle.com/javase/tutorial/java/>

Your EECS System Account

- You require an EECS account to complete this course.
 - marked course material
 - course announcements
- Activate your account:
 - <http://www.eecs.yorku.ca/activ8>
 - This account can be used to access **any** of the EECS labs, such as WSC 105,106,108 ; LAS1002/2B, 1004, 1006 and eecs-remotelab
 - Linux and Windows (machines can boot into either OS)
- disk quota, web space, print quota

Checklist – for week 1

- 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.
- Ensure you can access eClass site (email me if not)
- Watch videos and work through (Lab 0)
 - Try out eeecs-remotelab (or try install of eclipse/java on your computer)
 - Try out web-submit (and terminal submit from remotelab or in lab)
- Additional resources
 - There will be links to additional tutorial(s) & videos posted on course website (in Useful Resources Section)
 - E.g. for setting up your home computer/laptop environment
 - Using github (optional – video on setting this up provide in lab0)
 - Other tutorials/reference materials relating to course topics