

# COMP 1010- Summer 2015 (A01)

Jim (James) Young

[young@cs.umanitoba.ca](mailto:young@cs.umanitoba.ca)

[jimyoung.ca](http://jimyoung.ca)

# Hello!

James (Jim) Young

[young@cs.umanitoba.ca](mailto:young@cs.umanitoba.ca)

[jimyoung.ca](http://jimyoung.ca)

office hours T / Th: 17:00 – 18:00

EITC-E2-582

(or by appointment, arrange by email)

# computer science, to me....

## toys and fun!



XBOX Kinect

what I have done with computer  
science

dancing robots

cat toy!

robot control with toys!

robot-on-a-leash

and more!

**programming is just a tool**

my info...

office: EITC E2-582

office hours: T/Th 17:00 – 18:00  
or by appt.

email: [young@cs.umanitoba.ca](mailto:young@cs.umanitoba.ca)

[jimyoung.ca](http://jimyoung.ca)

EITC-E2-582

# today...

course description:

“Do I belong in this course?”

“What is computer programming?”

course logistics

overview, marking, website, etc.

introduction to computer programming

course description

# who can take this course?

no computer programming background required!

basic computer usage skills: typing, using new software, using the web..

if you have extensive programming background:

- a) you may be surprised at what you still learn
- b) there are ways of challenging the course



# what is this course about?

## 1) problem solving

how do you reach a goal given a set of tools?

learn the tools

practice using them to solve problems

## 2) computer programming

a specific set of tools that match computers

make computers solve problems for you

# what is this course *not* about?

learning how to use software

    microsoft word

    microsoft excel

learning how to make web pages

learning how to make a blog

how to use the internet

course logistics

the course website!!!!!!

(desire2learn)

<https://universityofmanitoba.desire2learn.com/>

COMP 1010

# D2L

assignments are submitted through D2L:  
MUST login to submit assignments  
(I will demo again at that time)

# D2L uses UMnetID

you need to make your accounts!

this is kind of crazy – all kinds of accounts!

D2L, aurora, jump, .....????

**claimid** ← you do this to create all your accounts

[umanitoba.ca/claimid](http://umanitoba.ca/claimid)

→ Create UMnetID

# things to see on D2L

ROASS (course outline)

grade breakdown

Course Schedule

Course notes

Labs

Forum

important announcements

# LABS!!

hands-on practice

TA there to help you

attendance mandatory (EASY MARKS)

ATTEND the section for which you are registered. you may not get credit otherwise!!

hint: exam questions are often taken from labs

hint: a little slow at them? they're posted ahead of time, work on them ahead of time



**LABS START NEXT WEEK**



# email policy

feel free to email me at any time

always include [comp1010] in the subject

however, **not** the first line of defense

help center!

office hours!

forum on D2L

Be polite:

“yo prof. was up im stuck on AS1 dude, dat  
hard \*\*\*”

deadlines!

are hard!

(almost) no exceptions!

hard drive crashed?

laptop stolen?

accidental deletion?



# USE BACKUP SOFTWARE

Dropbox, one drive, Wuala, Spideroak

automatically backs up as you work

you can access online if your laptop dies

synchronize between machines

# CHEATERS!!!!!!

permanent transcript  
specialized software

can be kicked from program

EVEN for signing your friends into the lab

EVEN if you PROVIDE assignment

ZERO tolerance, no excuses



# honesty declaration

there is a “check list” in D2L that you have to read and digitally sign off on. If you do not, you cannot submit assignments.

Midterm\*\*\*\*

Thursday, June 18, in class



# introduction to programming

# computers are fast!!

Cheap website:

\$500

2.9GHz

six cores



2.9GHz = 2.9 billion cycles

2,900,000,000 calculations per

six cores – do **six in parallel!**  
“we couldn’t afford faster computers,  
so we just made them *sound* faster.”

17,400,000,000 calculations per second

holy cow

# computers are stupid!

<http://www.youtube.com/watch?v=2H7NZ0GNIIE>

no common sense

no cultural sense!

doesn't speak english!

no ability of reason

can't "fill in the blanks" or "figure things out"

e.g., pick up the phone

pass the salt

copy my vacation photos

# computers are stupid!

computers only understand concrete, simple,  
unambiguous, logical instructions

has a specific, VERY limited, logical language

but, it can do billions of them per second!

“computers do exactly what yo  
Not what you want them to do



# computer programming:

Computer programming is the task translating something that you want to do into the computer's limited language.

Given a task or a job, you provide a “recipe” or exact set of instructions for the computer to complete that job.

# example – cooking recipe

## CORNBREAD

### Colvin Run Mill Corn Bread

1 cup cornmeal

1 cup flour

½ teaspoon salt

4 teaspoons baking powder

3 tablespoons sugar

1 egg

1 cup milk

¼ cup shortening (soft) or vegetable oil



a cooking recipe is a standard way to represent how to cook something so that (ideally, hah!) anyone can make a given dish. This is a **program** for cooking.

# example – music!

The image shows a musical score for a piece in 4/4 time, consisting of four staves. The first staff begins with an Am chord and contains a melodic line with a repeat sign. The second staff starts with a C chord and contains a melodic line with a repeat sign and two endings. The third and fourth staves contain accompaniment lines with chords F, C, and Am indicated above them.

this music score is a program for reproducing a piece of music. Gives /most/ of the details, the notes, the rhythm

# programmer-computer game

- a) make pairs!!!
- b) decide who is the programmer and who is the computer

task: write a set of instructions to reproduce a diagram:

no pictures! words only!

no metaphors or simile, no references, just basic objects and shapes!



programmer



computer



computers! put your heads down!!



computer

programmers, write your programs!



programmer



computers, execute the program!



computer

programming – not easy

and in this case, you had a smart HUMAN  
reading your program....

# introduction to Processing (and Java)

(and some quick history)

computers work on just a bunch of switches!

each switch can be either On or Off (no in the middle!)

shorthand, we say on is “1” and off is “0”

put 10 switches in a row: 0011011100

this language of 1s and 0s is called **binary**, and in the old days (<1950s), you would program a computer with a bunch of switches!

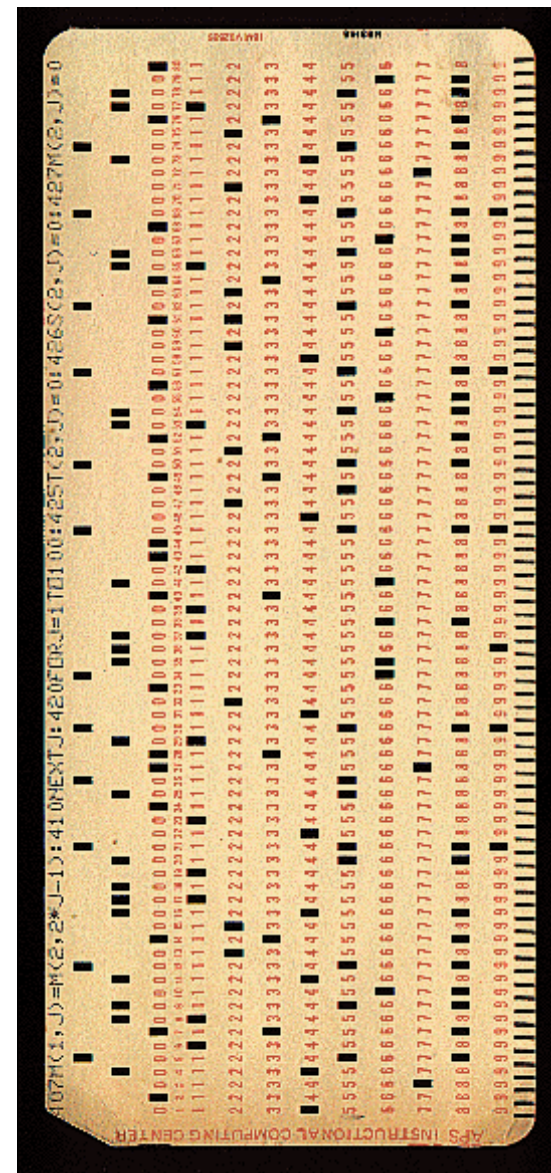
# early programming..



eniac computer, 1946

5 million hand-soldered joints,  
27tonnes, 150kWatts of power

5khz (5000 cycles / second),  
385 multiplications / second



punch cards: still binary  
used even up to mid 70s

# away from binary

assembly language: replace common binary sequences with commands for keyboard entry

e.g., (made up)

00111101 perhaps means to add the following numbers, so replace with the word “add”

this introduced a level of abstraction that made programming easier. This is “higher level” than pure binary programming



```
ORG ROM+$0000 BEGIN MONITOR
START LDS #STACK
```

```
*****
```

```
* FUNCTION: INITA - Initialize ACIA
* INPUT: none
* OUTPUT: none
* CALLS: none
* DESTROYS: acc A
```

```
RESETA EQU %00010011
CTLREG EQU %00010001
```

```
INITA LDA A #RESETA RESET ACIA
      STA A ACIA
      LDA A #CTLREG SET 8 BITS AND 2 STOP
      STA A ACIA

      JMP SIGNON GO TO START OF MONITOR
```

```
*****
```

```
* FUNCTION: INCH - Input character
* INPUT: none
* OUTPUT: char in acc A
* DESTROYS: acc A
* CALLS: none
* DESCRIPTION: Gets 1 character from terminal
```

```
INCH LDA A ACIA GET STATUS
      ASR A SHIFT RDRF FLAG INTO CARRY
      BCC INCH RECIEVE NOT READY
      LDA A ACIA+1 GET CHAR
      AND A #$7F MASK PARITY
      JMP OUTCH ECHO & RTS
```

# High level languages

most programming languages aim to be **human-readable** (hah) so you don't have to work in binary or assembly

java and processing are such “high-level” languages

# introduction to Java

Java is a programming language  
a standard way to give  
instructions to a computer



Other languages:

Perl

Python

C, C++

C#

# A basic Java program:

```
public class D1HelloWorld
{
    public static void main (String [] args)
    {
        System.out.println ("Hi out there!");
    }
}
```

Puts the text “Hi out there!” to the screen

# A basic Processing program

```
line(0,0,10,10);
```

draws a line to the screen

# introduction to processing

download it!

<https://processing.org/download/>

run it during class!!

You will download and use it in your first lab.

# Processing Development Environment

- Project name
- Version
- JAVA processor
- Run and stop buttons. Shortcuts.
- Tabs (not used for us)
- Text editor.
- Message bar
- console

# processing sketch

Programs are called Sketches!

- Sketchbook
- Change sketchbook location
  - (BACKUP!!)

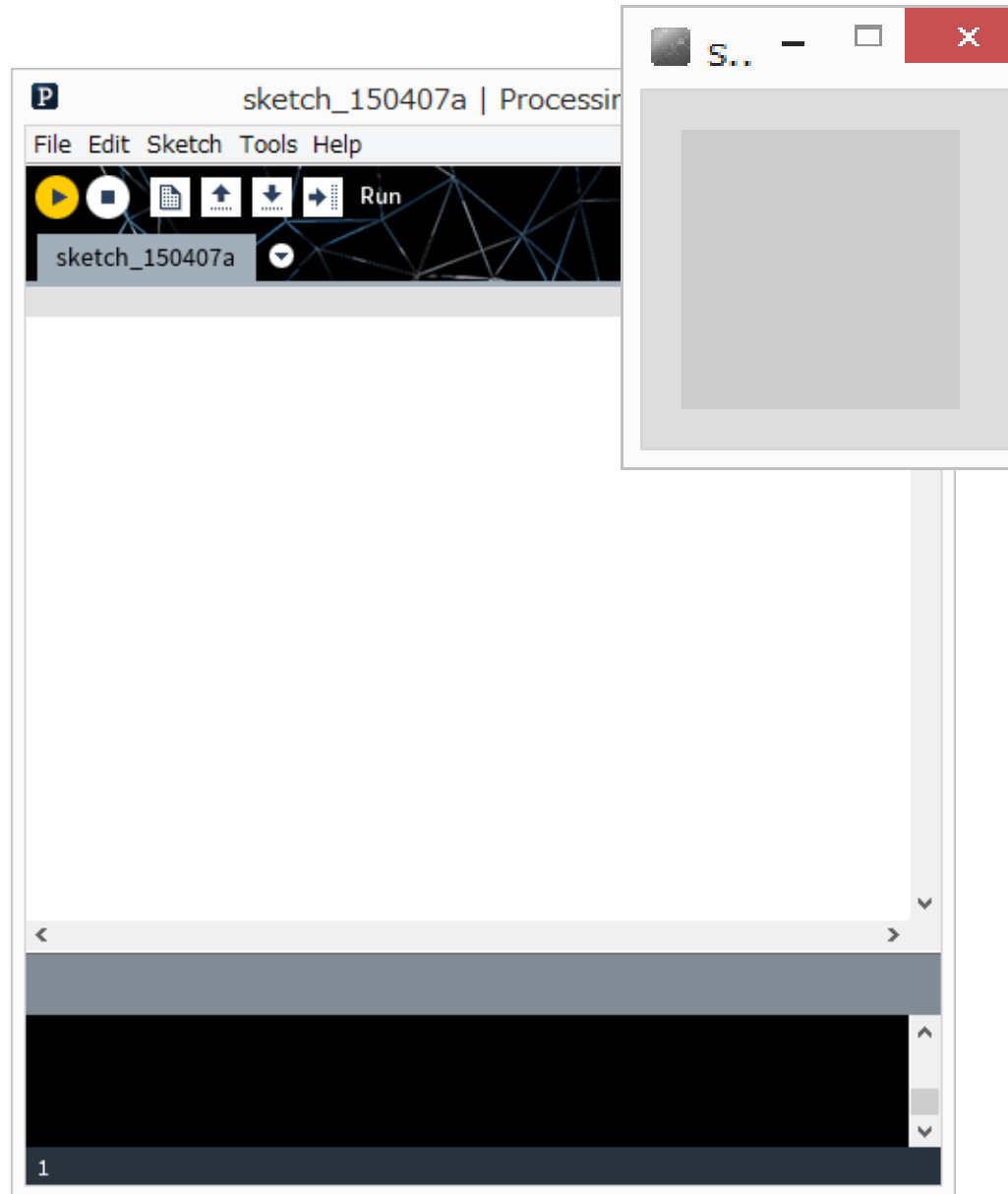


# The canvas!!!

play button

stop button

grey inside and  
outside

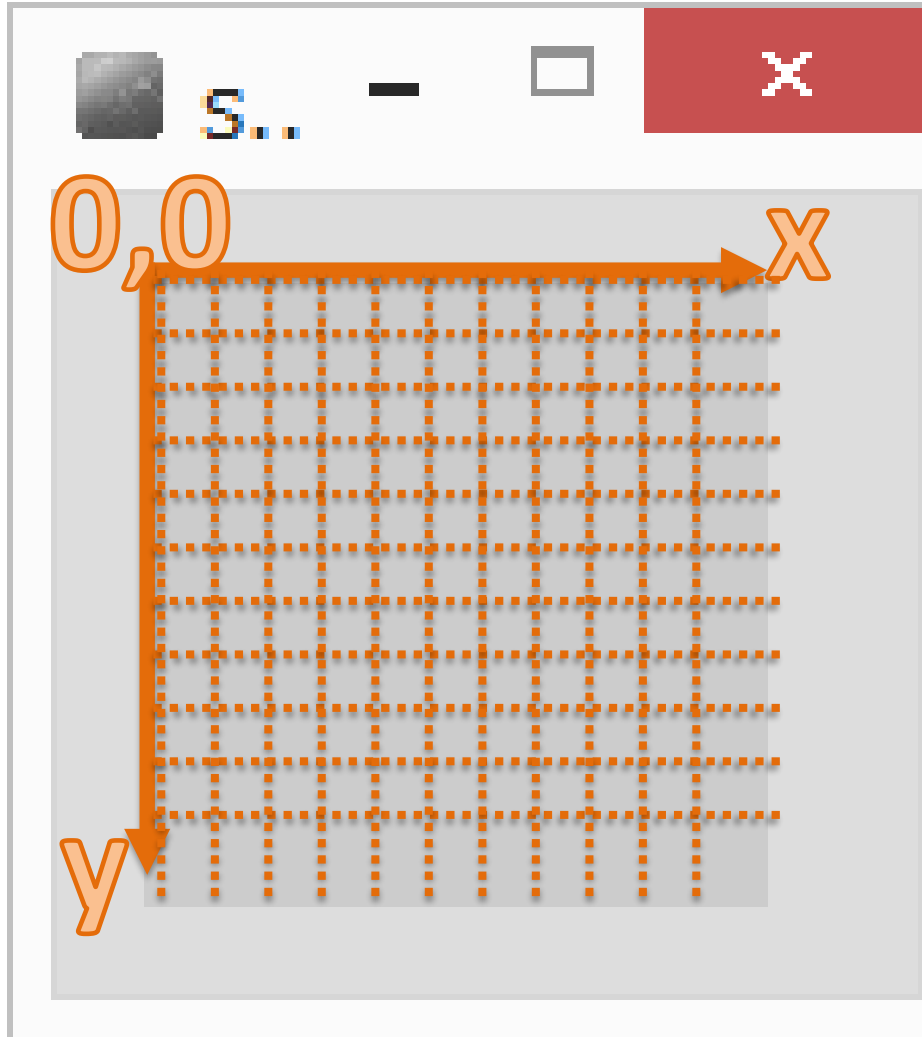


# The Canvas co-ordinate system

Euclidean coordinates

0,0 at top left

Default window is 100 by 100 dots (pixels)



# OK! Your first Processing Command draw a line!!!

Processing has some basic rules that we have to follow to give it commands. These rules are called **syntax**.

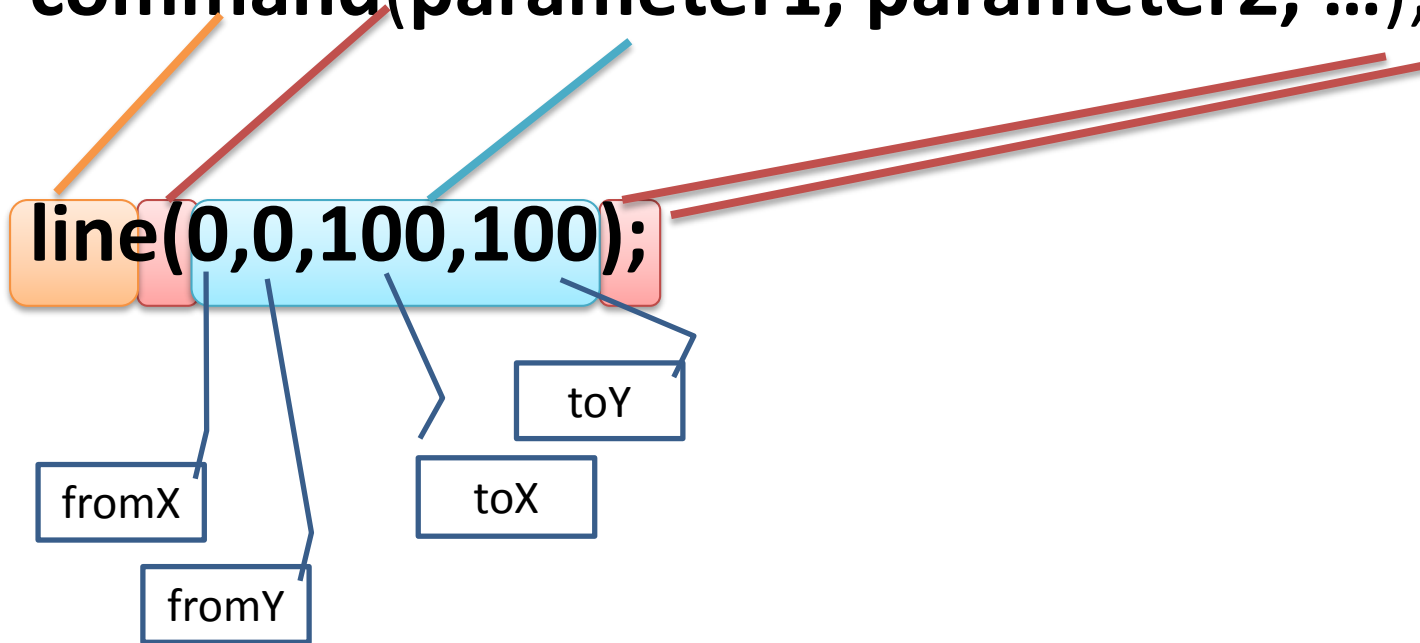
Remember how stupid computers are. We need to follow syntax or the computer won't understand.

# syntax of a command statement:

We need to specify the command (e.g, draw a line)

We need to pass information to the command (e.g., where to draw the line)

**command(parameter1, parameter2, ...);**



**line(0,0,100,100);**

How do you know what the required parameters are and what they mean???

Processing Help -> Reference

# **syntax issues...**

**command(parameter);**

Everyone seems to remember the parenthesis ()

But **don't forget the semi colon!**

The semi-colon means: end of command

Also – you can have spaces around the brackets and commas – doesn't matter.

What happens if you forget semi colon?

# Processing is **case sensitive!**

anything you type in processing is **case sensitive**

this means that **upper case** is different than  
**lower case**

line 

Line 

LiNe 

(computers are stupid)

## *compile time errors*

When Syntax Errors happen, Processing cannot convert your program into computer code. It happens when Processing tries to “compile” the program into binary. This means your program is never run.



# some more processing commands

Check these out in the reference

size()

ellipse()

... at home

point()

rect()

triangle()

What is the coordinate of the bottom right corner?

```
size(500,500);
```

```
line(0,0,499,499);
```