

# 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

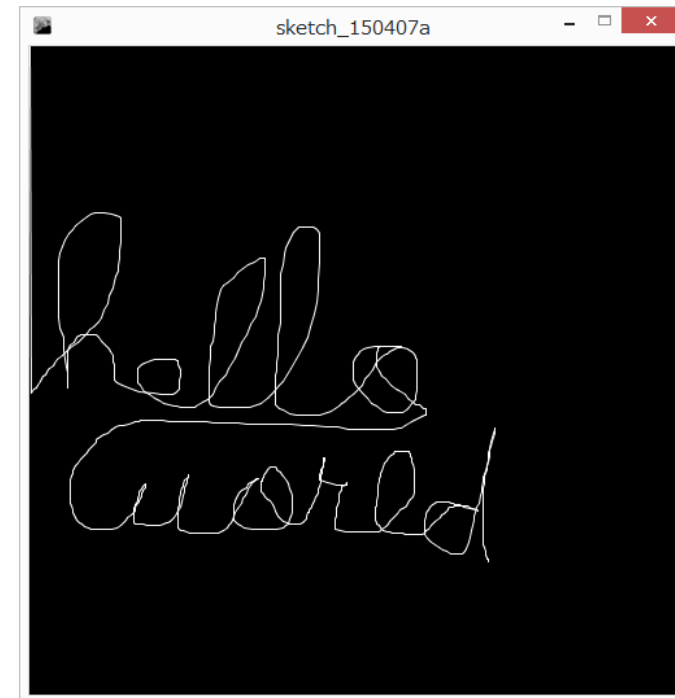
EITC-E2-582

(or by appointment, arrange by email)

# Update our drawing program

Only draw when the mouse is pressed

Erase the screen when the keyboard is pressed



# Draw a dashed line!!

- Have two drawing colors – A and B.
  - One is black one is white
  - Each time we draw, we swap A and B

```
int colorA =255;
```

```
int colorB = 0;
```

How do we swap the values in two variables???

# Swap version1

```
int colorA =255;
```

```
int colorB = 0;
```

```
// swap
```

```
colorA = colorB;    colorA set to 0
```

```
colorB = colorA;    color b set to 0
```

What happens?

# To swap a variable, you need a temporary holding variable

```
int colorA =255;
```

```
int colorB = 0;
```

```
int swapTmp;
```

```
// swap
```

```
swapTmp = colorA;    save colorA copy (255)
```

```
colorA = colorB;    colorA is now 0
```

```
colorB = tmp;    colorB set to colorA copy(255)
```

# Opposite test

We can do code IF something is TRUE

What about if something is not true?

Goal – if the mouse button is NOT pressed, then draw a black circle underneath it to simulate an eraser

# boolean operations!: negation

negation!! how can we get the opposite?

true ->false, false -> true

!boolean, the exclamation part negates

```
boolean isJimRich = false;
```

```
boolean isJimPoor = !isJimRich;
```

also known as “not”.



# Somewhat confusing...

if (!mousePressed)...

if the opposite of MousePressed is true

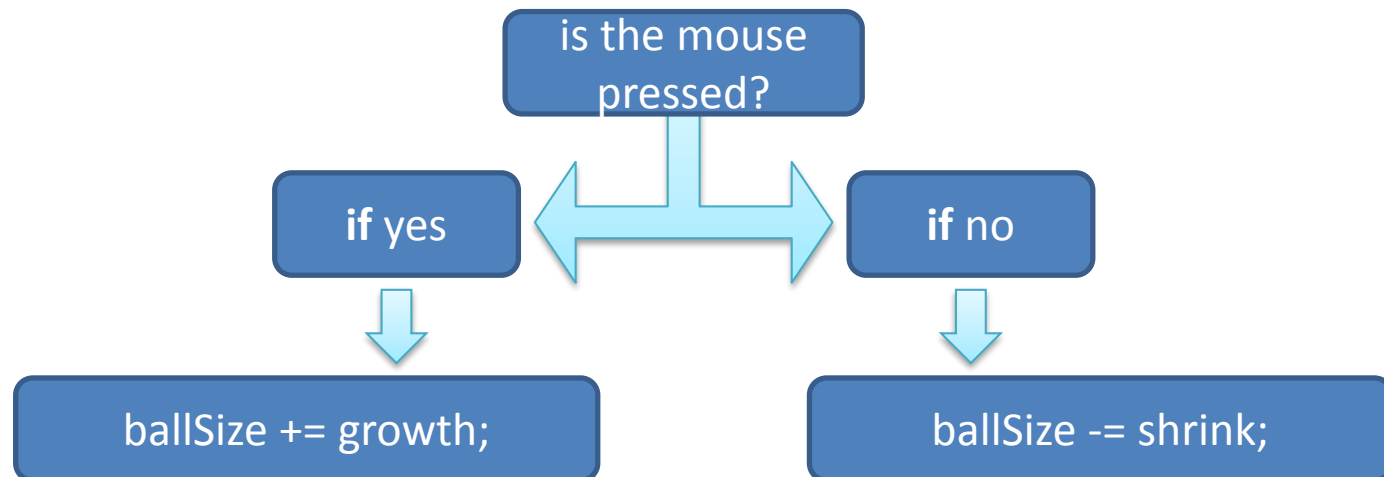
if mousePressed is false

Then draw a black circle!

# Growing / shrinking ball

First make a program where a ball grows or shrinks

use the if statement to detect if the mouse is pressed, and act on that



we often do two opposite tests:

```
If (mousePressed)
```

```
{
```

```
  // do stuff
```

```
}
```

```
If (!mousePressed)
```

```
{
```

```
  // do stuff
```

```
}
```

# Two opposite tests!

it would be useful if we could say: **otherwise**

e.g., if the mouse is pressed, do something,  
otherwise, do something else

Processing has the **else** keyword to do this!

```
if (mousePressed)
```

```
{ //... }
```

```
else
```

```
{ //... }
```

} this code is ONLY run if the boolean test  
above is FALSE.

# if-then-else

```
if (boolean test) {  
  // then do all this stuff..  
  ...  
} else {  
  // otherwise do all this stuff.  
}
```

Update the program

Make a ball move left if the mouse is  
pressed,  
right otherwise

Up if keypressed, down otherwise

Don't clear the BG, and set a stroke color to  
leave a trail

# nesting ifs...

in computer programming, **nesting** is when you put one thing inside of another

Let's update our drawing program.

Previously: press a key clears the canvas to black

Let's make it so that if you press a key while the mouse is down, clear to white

# Previous code

```
if (mousePressed)
{
  stroke(255);
  line(pmouseX, pmouseY, mouseX, mouseY);
} else {
  stroke(0);
  fill(0);
  ellipse(mouseX, mouseY, 25, 25);
}
```

```
if (keyPressed)
{
  background(0);
}
```



# How:

Once we check if a key is pressed, make the clear color depend on whether or not the mouse is down.

Use another if statement **INSIDE** the first one

```
if (keyPressed)
{
  if (mousePressed)
  {
    background(255);
  } else {
    background(0);
  }
}
```

Inside the if keyPressed block, so this code is **ONLY RUN** if the keyPressed test is true.

Narrow tunnel vision!  
Step by step!

# relational operators: comparing data

`==` equals, true if the left and right operands are equal

`a == b` // gives a boolean result

```
int a = 3;
```

```
int b = 3;
```

```
boolean areEqual = (a == b); // brackets not necessary!!
```

Test if the mouse is on the diagonal  
and do something – draw a circle

Make Boolean variable that checks

Do if statement based on that

# relational operator summary

operator	example	true if...
!=	a != b	they are /not/ equals!
<	a < b	true if a is less than b
<=	a <= b	true if a is less than or equal to b
>	a > b	true if a is greater than b
>=	a >= b	.. a is greater than or equal to b

these all return the boolean type

```
boolean result = a < b;
```

# relational operator examples

`2 < 5 // true, 2 is less than 5`

`2 <= 5 // true, 2 is less than or equal to 5`

`2 > 5 // false, 2 is not greater than 5`

`2 == 2 // true, 2 is equal to 2`

`2 <= 2 // true, 2 is less than or equal to 2`

`2 >= 2 // true, 2 is greater than or equal to 2`

`2 != 2 // false, 2 is equal to 2`

`2 == (1+1) // true, 2 equals 1+1`

`3 == min(9, 3); // true!`

# relational operator examples

`2*10 <= max(-10,-2)` // false, 20 is not <= -2

`3 == 3*1` // true, 3 == 3!!

`3 != min(3, 1)` // true, 3 != 1!

# Variable not needed for if statement..

```
boolean onDiagonal = (mouseX==mouseY);  
if (onDiagonal)  
{  
    // ...  
}
```

```
if (mouseX == mouseY)  
{  
    // ...  
}
```



# Example with relational operators and ifs

Set color based on x  
value

0..99	color 50
100..199	color 100
200..299	color 150
300..399	color 200
400..500	color 255



Start over with simple example: draw if mouse pressed, clear on screen press

First test – add mouseX < 100

how to add the next one? 100..199?

re-think – not < 100, but < 200

use the else block

Next one? 200..299?

same approach

THIS IS GETTING MESSY!

# Careful points..

Hard to match brackets up – always make set of { and } when making a block to avoid problems

What about.

```
if(...)  
{  
} else {  
  // ..  
} else {  
  //  
}
```

# If – else – if is a very common pattern

New syntax!

```
if (condition)
```

```
{  
}
```

```
else if (condition) // only if above condition was false
```

```
{  
}
```

```
else if (condition) // you can have as many of as you like
```

```
{  
}
```

```
else // only run if ALL the above conditions are false
```

```
{  
}
```

Update program