

# 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)

Seminar on Thursday

EITC-E2-304

```
void setup()
{
  size(500, 500);
  background(0);
}

void draw()
{
  int drawColor = 0;
  if (mouseX < 100)
  {
    drawColor = 50;
  } else {
    if (mouseX < 200)
    {
      drawColor = 100;
    } else {
      if (mouseX < 300)
      {
        drawColor = 150;
      } else {
        if (mouseX < 400)
        {
          drawColor = 200;
        } else {
          drawColor = 255;
        }
      }
    }
  }
}

if (mousePressed)
{
  stroke(drawColor);
  line(pmouseX, pmouseY, mouseX, mouseY);
}
}
```

# If – else – if is a very common pattern

New syntax!

```
if (condition)
```

```
{  
}
```

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

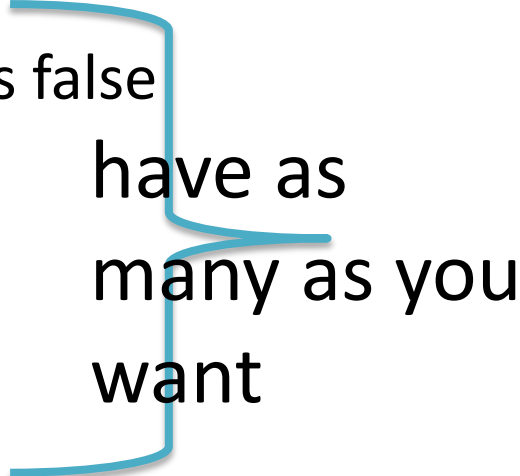
```
{  
}
```

```
else if (condition) // only if all above are false
```

```
{  
}
```

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

```
{  
}
```



have as  
many as you  
want

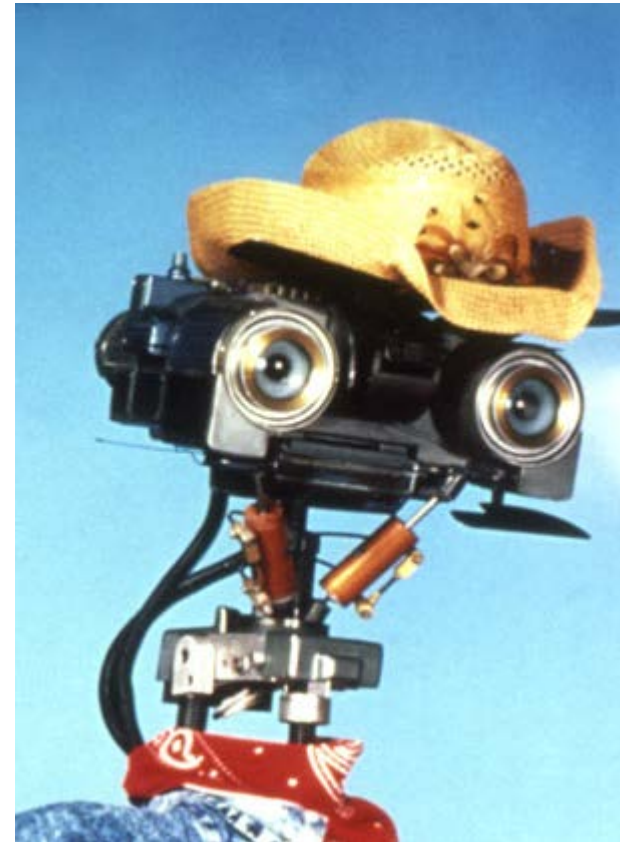
Update program

# short-circuiting!

**note:** if a condition is met, does not test any other conditions in the if-else-if chain

```
if (mouseX < 500) {  
    stroke(255);  
} else if (mouseX < 400) {  
    stroke(200);  
} else if (mouseX < 300) {  
    stroke(150);  
} else if (mouseX < 200) {  
    stroke(100);  
} else {  
    stroke(50);  
}
```

**this code has a bad bug!**



# blocks impose **SCOPE** rules

**scope** is the range within which a variable exists. Outside of that scope, you cannot access or work with that variable.



# SCOPE

variables created within one block can **ONLY** be accessed within that block. Each **code block** has its own **local scope**.



```
boolean hasUsedScopeMouthWash = true;  
if (hasUsedScopeMouthWash)
```

```
{  
    int freshness = 10;  
}
```

```
println(freshness);
```

this code will not compile! The **freshness** variable was created within the scope of the code block, and once that **scope** ends, the variable cannot be accessed...

how to fix this?



# what about nested blocks?

```
if (booleanA)
```

```
{  
    int d = 12345;  
    if (booleanB)  
    {  
        println(d);  
    }  
}
```

the purple scope is **inside** the blue scope, so it can see the int variable d

# what about nested blocks?

if (booleanA)

```
{  
    int d = 12345;  
    if (booleanB)  
    {  
        int d2 = d*2;  
    }  
    println(d2);  
}
```

**ERROR**

d2 is created INSIDE the purple scope, so it cannot be accessed outside of that scope

# what about nested blocks??

```
if (goodWeather) {  
    int goodTemp = 25;  
    if (inWinnipeg) {  
        boolean coldAdjust = true;  
        goodTemp -= 5; OK!  
    } else if (inAtlanta) {  
        goodTemp += 5; OK!  
        boolean warmAdjust = true;  
        if (coldAdjust) ERROR  
            goodTemp += 5;  
        }  
    }  
}  
ERROR  
println(goodTemp);
```

# How to avoid scope issues..



## **Follow good coding practices:**

declare all your variables at the top of your draw block

use globals when needed

**why not make them all globals?**

**as your programs grow, very messy!**

# Boolean operations

How to do complex boolean tests..

If both x AND y are true then..

If either x OR y are true then..

# boolean operations!: and

**note:** the AND operation uses the symbols `&&` to determine if both of two booleans are true:

`boolValueA && boolValueB // true ONLY if both are true`

AND (&&) Truth Table			
A	B	Operation	Result
false	false	A && B	false
false	true	A && B	false
true	false	A && B	false
true	true	A && B	true!

```
boolean jimIsRich = false;
```

```
boolean iNeedMoney = true;
```

```
boolean shouldMugJim = jimIsRich && iNeedMoney;
```

Make a program to draw a circle at the mouse only if it is in the top left corner

If mouseX < 250

AND

If mouseY < 250

# boolean operations!: or

**note:** the OR operation uses the symbols `||` to determine if either of two booleans are true:

```
boolValueA || boolValueB; // true if A OR B is true
```

OR (&&) Truth Table			
A	B	Operation	Result
false	false	A    B	false
false	true	A    B	true
true	false	A    B	true
true	true	A    B	true

```
boolean jimIsRich = false;
```

```
boolean iNeedMoney = true;
```

```
boolean shouldMugJim = jimIsRich || iNeedMoney;
```



Make a program that draws only in the top 25% and the bottom 25%

If mouseY < 125 OR

If mouseY > 350

# Example: click a button

Create a small rectangle to be your button

Name all the parameters as variables

Test one case first – mouse to the right of the button left wall – change button color if true

Test all four walls – use &&

Only change when mouse clicked

# Example – rebounding ball

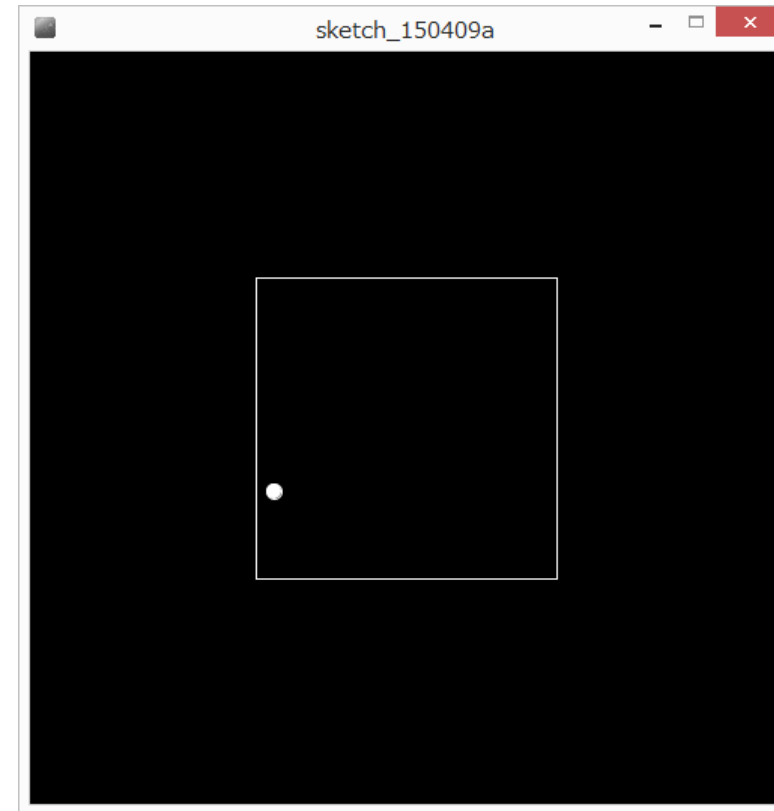
Make variables for a ball  
position, movement, size

Draw the ball

Move the ball (neg. movement?)

Make variables for a box and  
draw it

How to bounce the ball?



# Bounce the ball!

If hit left or right wall – reverse X direction

If hit top or bottom wall, reverse Y direction

How?

Multiply by -1

# Exercises:

The ball bounces when the center of the ball – not an edge – hits. Fix so that it bounces when an edge hits

Add a smaller box in the center (say, 10x10) that the ball also bounces off of and stays outside of

# boolean logic nesting

```
boolean result = ! ( c || !(a<b)); // this is legal
```

Assuming that this runs, what do you know about the data types of the variables a, b, and c?

a and b must be numerical because we are doing a less-than operator.

c must be boolean because it is an operand to the boolean OR operator.

what is the result if a=3, b=1, and c=true; ?

