COMP 1010- Summer 2015 (A01)

Jim (James) Young

young@cs.umanitoba.ca

jimyoung.ca

Don’t use == for floats!!

Floating point numbers are approximations. How they are stored inside the computer means that base 10 numbers cannot be perfectly stored

There is an accumulated error as you do operations

If you have a pocket of change, and you..

count nickles, then dimes, then quarters, then loonies, an you get $15.25

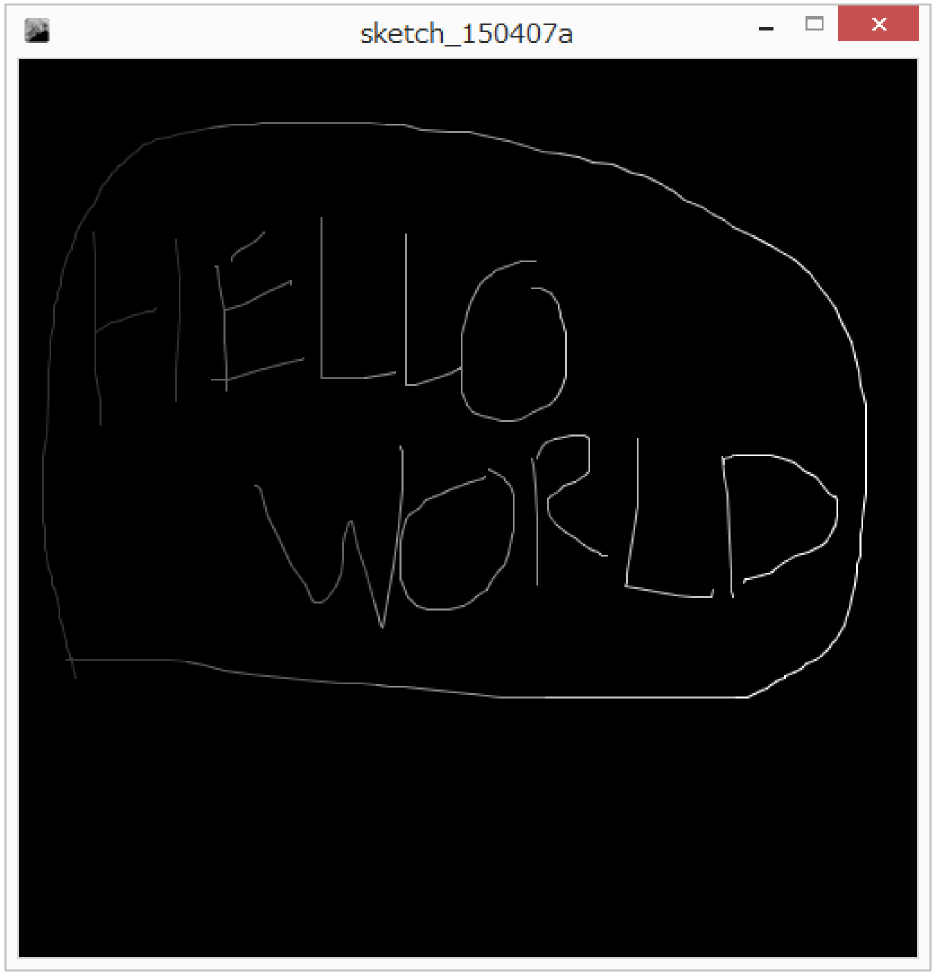
same change: count loonies, then quarters, then dimes, then nickles, you may get $15.22!

The order matters! -> don’t use ==

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 with simple example: draw if mouse pressed, clear on key 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!

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

Careful points..

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

What about.

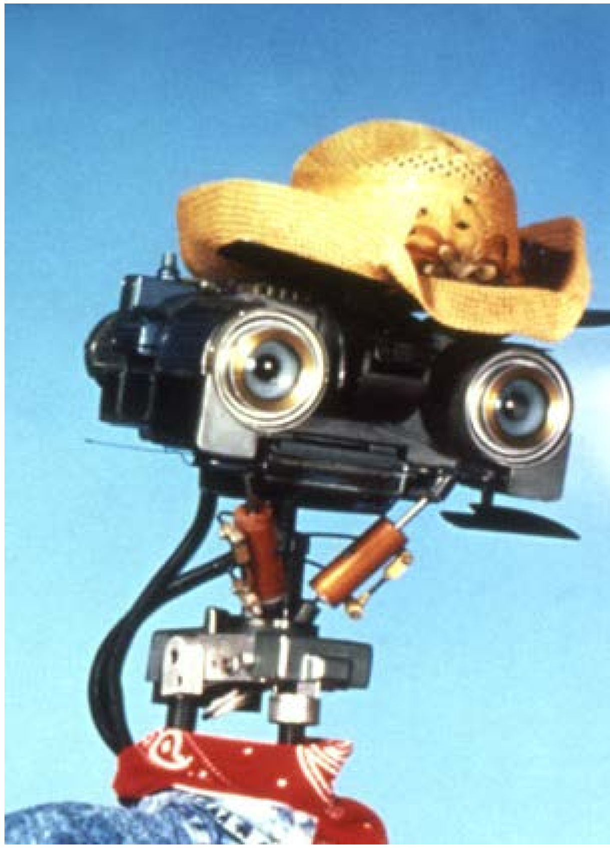
```
if(…)
{
} else {
 // ..
} else {
 //
}
```

short-circuiting!

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

this code has a bad bug!

```
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);
}
```



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;
        }
}
println(goodTemp);                    ERROR
```

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 quadrant

If mouseX < width/2

AND

If mouseY < height/2

boolean operations!: or

note: the OR operation uses the symbols || to determine of 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 < height/2 OR

If mouseY > 3*height/4