

# COMP 1010- Summer 2015 (A01)

Jim (James) Young

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

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

# order of operations with the explicit cast!

```
int integerValue = (int)0.5*3.0;
```

Casts happen first!

the cast converts the 0.5 to an int first, = 0

second, the multiplication takes place: 0 \* 3.0

the result is a floating point.

To fix this?

```
int integerValue = (int)(0.5*3.0);
```

# What we learnt...

Data types have a fixed amount of memory, which dictates how much information they can store

Different datatypes store information differently, e.g., floating point versus integer

you can convert between data types (floating point or integer) and between memory sizes (e.g., long<->int, float<->double) using casts.

# Example: rose art, all at once

$X = \cos(k*t)*\cos(t)*scale+250;$

$Y = \cos(k*t)*\sin(t)*scale+250;$

k determines number of petals

As t goes from 0 to  $2..PI$ , x and y draw a shape

Previously, t increases each time we draw. We see the rose petal get drawn.

Instead, draw the whole petal using a for loop

k is the number of petals

Draw lines through the x,y points as t goes from 0..  
 $2PI$

# How to make $t$ go from $0..2\pi$ with a for loop?

$0..2\pi$  is a continuous range.

How often should you stop in that range?

Computers are not continuous, we need to specify this

Specify number of steps, and use that to setup the for loop.

# Example: rose art, all at once

For  $0 \leq i < \text{steps}$ :

$$t = i / (\text{steps} - 1) * 2\text{PI};$$

$$X = \cos(k * t) * \cos(t) * \text{scale} + 250;$$

$$Y = \cos(k * t) * \sin(t) * \text{scale} + 250;$$

Play with parameters

# While loops

# while loop

```
while (booleanCondition) { // is true
    repeatTheseCommands;
    repeatTheseCommands;
}
```

e.g., :

```
int i = 2;
```

```
while (i < 10) {
    i++;
}
```

everything between the brackets gets repeated again and again while the condition is true.

```
println(i); // is i 10 or 9?
```



# the **while** loop:

count to 1000

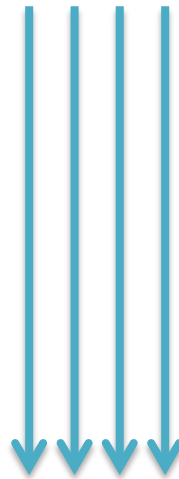
- 1) setup like an **if** statement
- 2) change the **if** to **while**

```
int i = 1;  
if (i <= 1000) {  
    while (i <= 1000) {  
        println(i);  
        i++;  
    }  
}
```

just like the **if**, put multiple statements  
in the **while** loop

```
while (booleanCondition)
```

```
{  
  statementA;  
  statementB;  
}
```



this entire block gets repeated while the  
booleanCondition is true

once inside the loop, the condition does not get checked again until the end

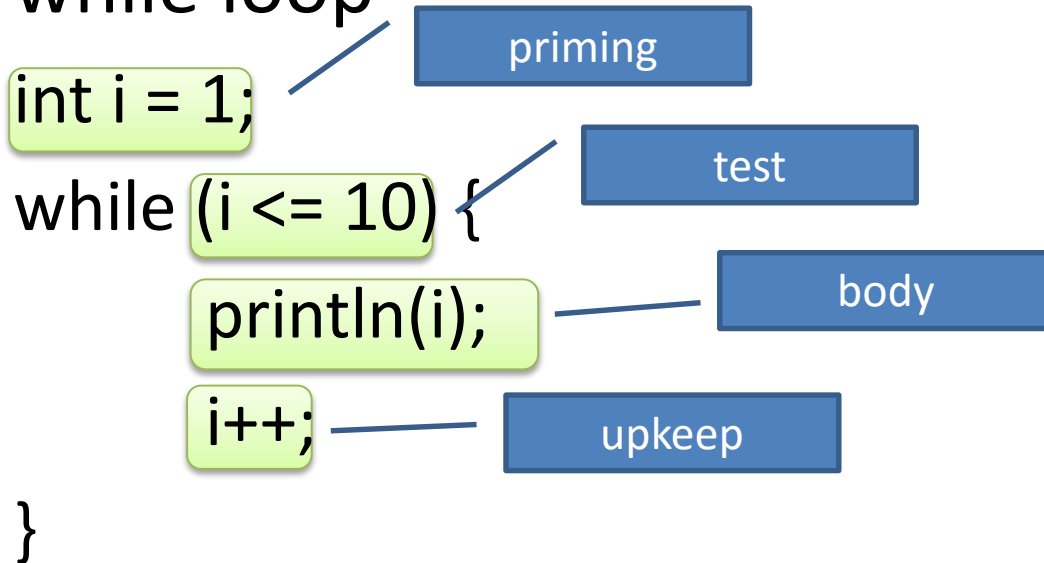
```
int i = 1;
```

```
while (i < 10) {  
    println(i);  
    i++;  
    println(0);  
}
```

the “and” gets print out even after the last #

# The while loop has four general parts

while loop



# for loop and while loop are similar...

for loop

```
for (i = 1; i <= 10; i++) {  
    println(i);  
}
```

while loop

```
int i = 1;  
while (i <= 10) {  
    println(i);  
    i++;  
}
```

# general usage of loops:

for loops : repeating something a certain number of times,  
e.g.,

- going over every student record and calculating letter grades

- going over a range of numbers for a calculation on each one

while loops: iteration count unsure, keep looping until a condition is met, e.g.,

- you are not sure ahead of time how many times to loop
- checking a network connection

choosing the wrong loop is not a big deal, you can easily fix it.

# Example: random ball path

If a ball starts at the mouse, and moves randomly, what path will it take to get off the screen?

- We can do this already
  - Each time we draw, move the ball, and draw a dot.
  - Don't clear -> we get a path.

BUT! What if we want to draw the whole path in a frame?

- how many random movements will it take to get off the screen?
- if we knew how many, we could do a for loop
- but we don't know how many -> while loop

# Random ball path: algorithm

Draw:

if the mouse is pressed:

    place a ball at the mouse position

    clear the screen

    while the ball is still on the screen

        do a random movement

        draw a point to represent the ball



# Implement the skeleton without the while...

Then setup the while loop

- boolean variable for on screen
- test the variable

this doesn't work!! Why??

We need the loop upkeep!

```
final float MAX_SPEED = 10;
final float BALL_SIZE = 5;

void setup()
{
  size(500, 500);
}

void draw()
{
  stroke(255);
  fill(255);

  if (mousePressed)
  {
    background(0);
    float ballX = mouseX;
    float ballY = mouseY;

    // is the ball on the screen?
    boolean onScreen = ballX >= 0 && ballX < width &&
      ballY >= 0 && ballY <= height;
    while (onScreen)
    {
      // move the ball
      float move = random(2*MAX_SPEED)-MAX_SPEED;
      ballX += move;
      move = random(2*MAX_SPEED)-MAX_SPEED;
      ballY += move;

      ellipse(ballX, ballY, BALL_SIZE, BALL_SIZE);
      onScreen = ballX >= 0 && ballX < width &&
        ballY >= 0 && ballY <= height;
    }
  }
}
```