Iteration in Programming

for loops

Produced by: Mairead Meagher
Dr. Siobhán Drohan

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

Department of Computing and Mathematics
http://www.wit.ie/

Topics list

- There are three types of loop in programming:
  - While loops (covered in previous talk):
    - Counter controlled (n times)
    - Sentinel based (covered later in the course)
    - Flag based (covered later in the course)
  - For loops
  - Do While loops
- Comparative use of while and for loops
  - Lab03 - Challenge Exercise 1
  - Lab03 - Challenge Exercise 3

For loop pseudo-code

General form of a for loop

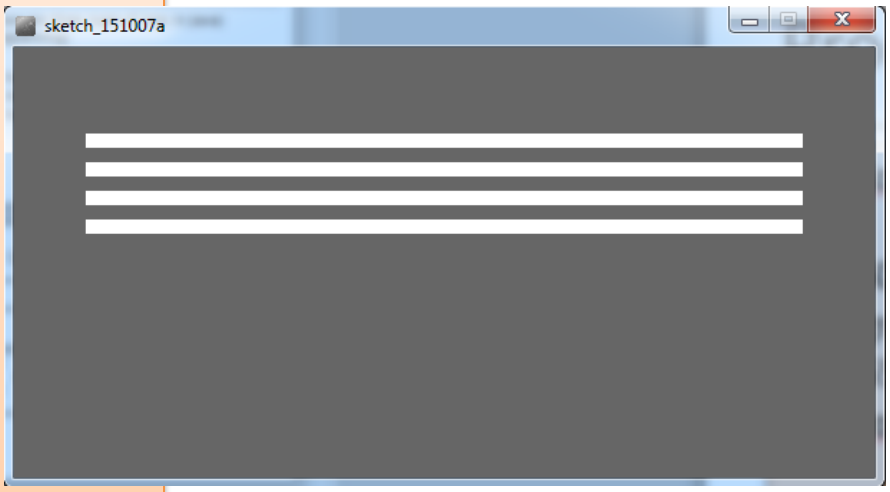
for(*initialization*; *boolean condition*; *post-body action*)
{
    statements to be repeated
}

Processing Example 4.4

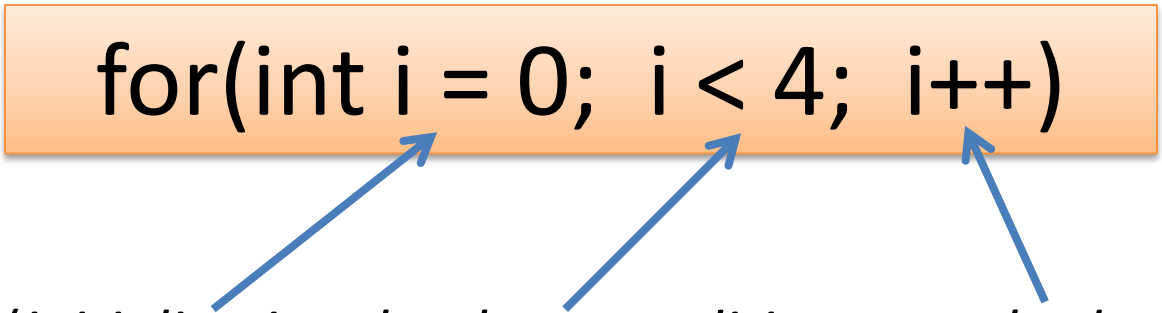
```
int yCoordinate = 60;

size(600, 300);
background(102);
fill(255);
noStroke();

for(int i = 0; i < 4; i++)
{
    rect(50, yCoordinate, 500, 10);
    yCoordinate = yCoordinate + 20;
}
```



For loop syntax



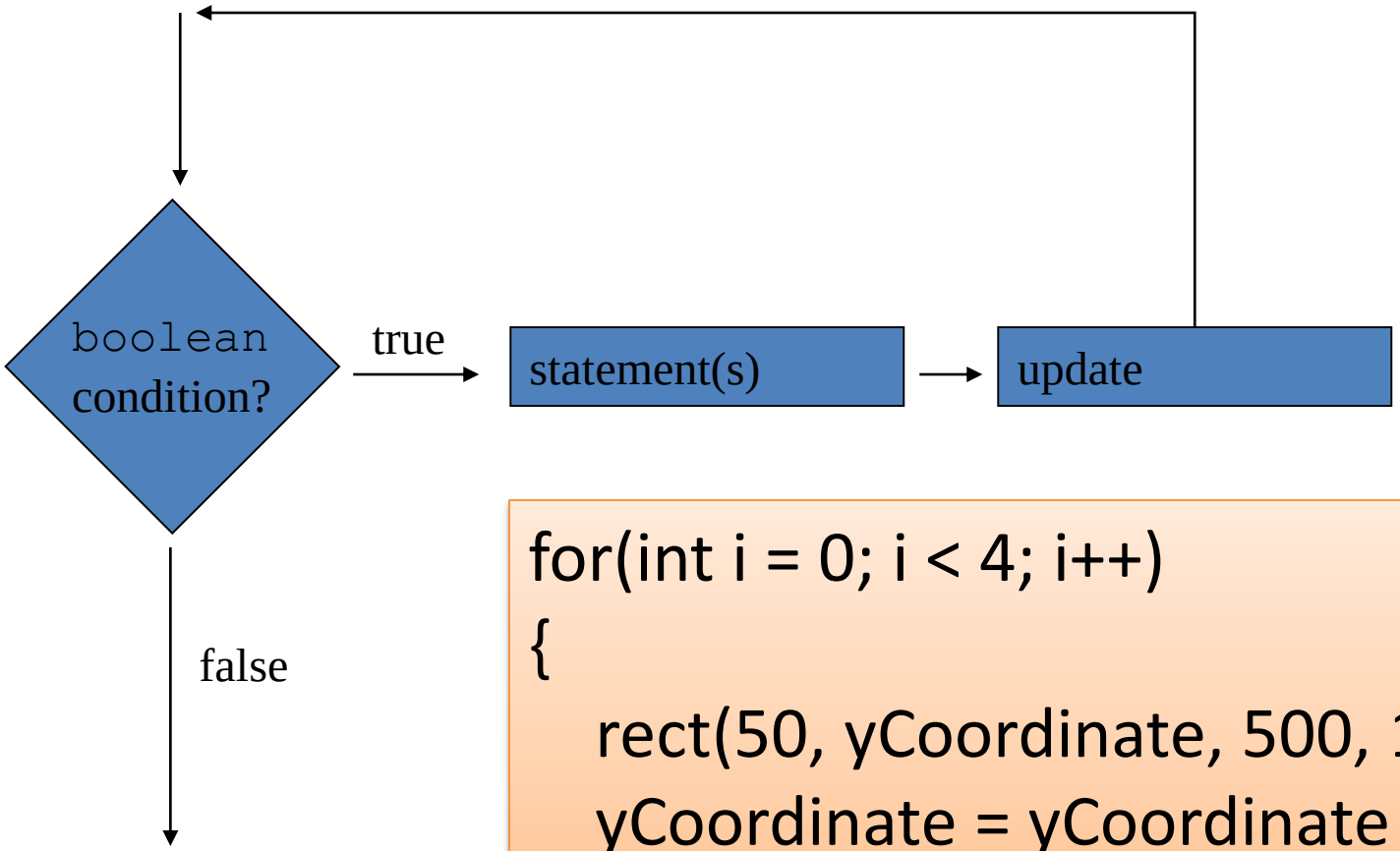
for(*initialization*; *boolean condition*; *post-body action*)
{
    statements to be repeated
}

For loop syntax

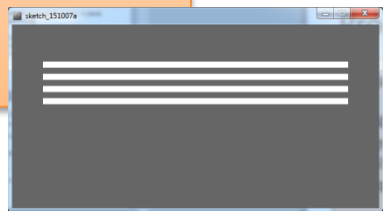
```
for(int i = 0;  i < 4;  i++)
```

Initialization	int i = 0	Initialise a loop control variable (LCV) e.g. i. It can include a variable declaration.
Boolean condition	i < 4	Is a valid boolean condition that typically tests the loop control variable (LCV).
Post-body action	i++	A change to the loop control variable (LCV). Contains an assignment statement.

`for` Loop Flowchart




```
for(int i = 0; i < 4; i++)
{
    rect(50, yCoordinate, 500, 10);
    yCoordinate = yCoordinate + 20;
}
```



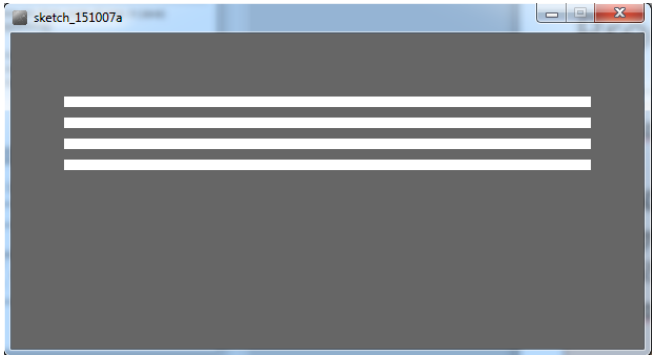
Returning to: Processing Example 4.4

```
int yCoordinate = 60;

size(600, 300);
background(102);
fill(255);
noStroke();

for(int i = 0; i < 4; i++)
{
    rect(50, yCoordinate, 500, 10);
    yCoordinate = yCoordinate + 20;
}
```

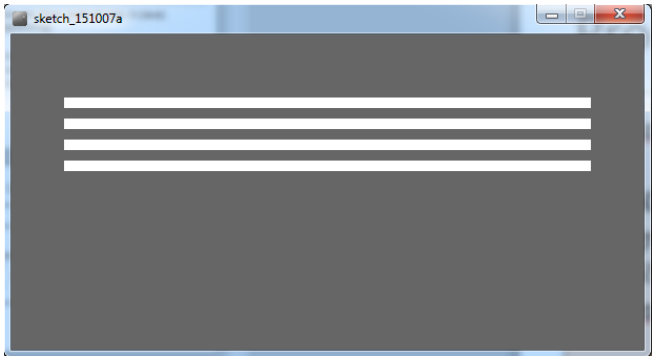
Do we need the yCoordinate variable? Can you think of a different approach using a for loop?



Updated: Processing Example 4.4

```
size(600, 300);
background(102);
fill(255);
noStroke();

for(int i = 60; i <= 120; i = i + 20)
{
    rect(50, i, 500, 10);
}
```



For loop: all parts are optional

```
for ( ; ; )
{
    // statements  here
}
```

This is an infinite loop…

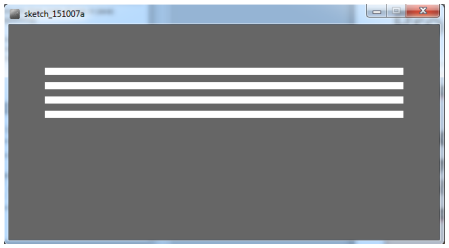
For loops can be nested

```
The value of i is: 0 and j is: 0
The value of i is: 0 and j is: 1
The value of i is: 0 and j is: 2
The value of i is: 0 and j is: 3
The value of i is: 1 and j is: 0
The value of i is: 1 and j is: 1
The value of i is: 1 and j is: 2
The value of i is: 1 and j is: 3
The value of i is: 2 and j is: 0
The value of i is: 2 and j is: 1
The value of i is: 2 and j is: 2
The value of i is: 2 and j is: 3
The value of i is: 3 and j is: 0
The value of i is: 3 and j is: 1
The value of i is: 3 and j is: 2
The value of i is: 3 and j is: 3
```

```
for (int i=0; i < 4; i++)
    for (int j=0; j < 4; j++)
        println("The value of i is: " + i + " and j is: " + j);
```

Topics list

- There are three types of loop in programming:
  - While loops (covered in previous talk):
    - Counter controlled (n times)
    - Sentinel based (covered later in the course)
    - Flag based (covered later in the course)
  - For loops
  - Do While loops
- Comparative use of while and for loops
  - Lab03 - Challenge Exercise 1
  - Lab03 - Challenge Exercise 3



for versus while

Processing Example 4.4 (for loop)

```
for(int i = 0; i < 4; i++)
{
   rect(50, yCoordinate, 500, 10);
   yCoordinate += 20;
}
```

Processing Example 4.5 (while loop)

```
int i = 0;
while(i < 4)
{
   rect(50, yCoordinate, 500, 10);
   yCoordinate += 20;
   i++;
}
```

Variable **i** is the Loop Control Variable (LCV). It must be initialised, tested and changed.

int i = 0 is the initialisation.

i < 4 is the boolean Condition i.e. the test

i++ is the post-body action i.e. the change.

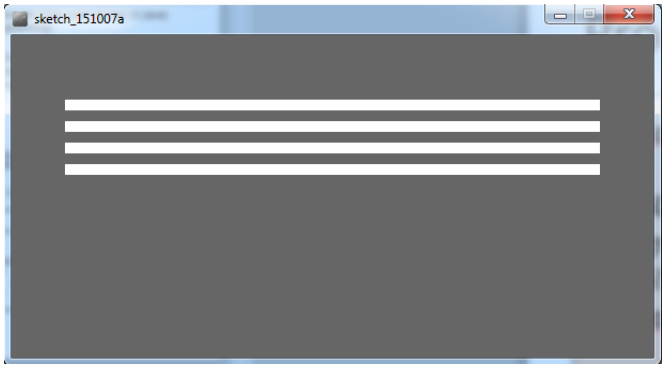
Processing Example 4.5

```
int yCoordinate = 60;

size(600, 300);
background(102);
fill(255);
noStroke();

int i = 0;
while(i < 4)
{
    rect(50, yCoordinate, 500, 10);
    yCoordinate += 20;
    i++;
}
```

As we did with the for loop, can we remove the **yCoordinate** variable?

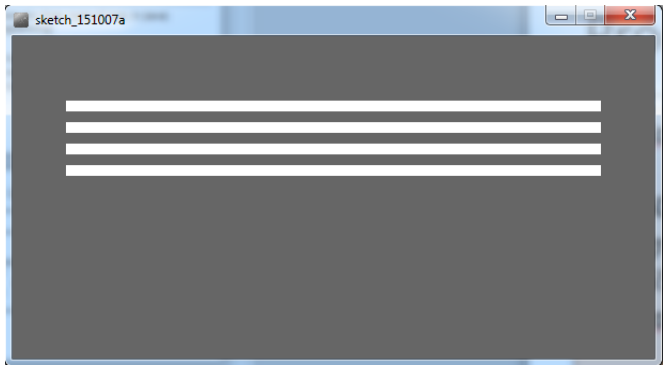


Updated: Processing Example 4.5

```
size(600, 300);
background(102);
fill(255);
noStroke();

int i = 60;
while(i  <= 120)
{
    rect(50, i, 500, 10);
    i += 20;
}
```

Solution with no **yCoordinate** variable.

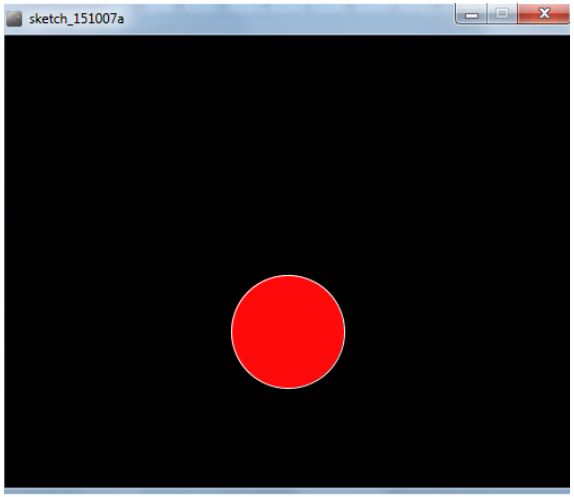


Topics list

- There are three types of loop in programming:
  - While loops (covered in previous talk):
    - Counter controlled (n times)
    - Sentinel based (covered later in the course)
    - Flag based (covered later in the course)
  - For loops
  - Do While loops
- Comparative use of while and for loops
  - Lab03 - Challenge Exercise 1
  - Lab03 - Challenge Exercise 3

Lab03 - Challenge Exercise 1

Draw a continuously bouncing ball. For the moment, the xCoordiate should remain the same value i.e. the ball only bounces vertically, so it is only the yCoordiate that needs to change.



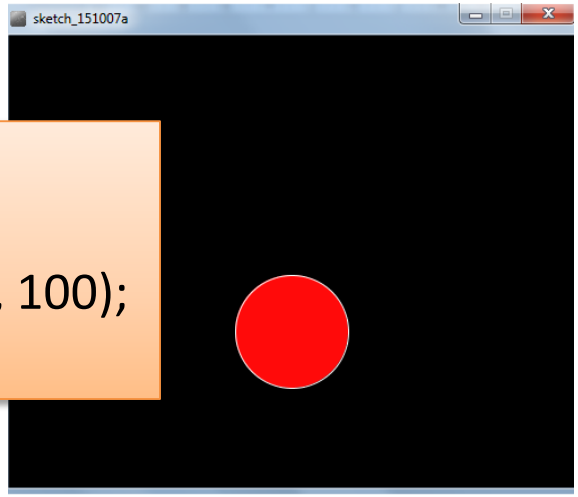
- Assume the display window is 500 x 400
- Assume the ball is 100 in diameter.
- Assume the static xCoordinate is 250.
- Assume the background is called in the draw() method.
- Assume the starting yCoordinate is 300.

Lab03 - Challenge Exercise 1

```
float yCoordinate = 300;

void setup() {
  size(500,400);
  fill(255, 10, 10);
  stroke(255);
}
```

```
void draw() {
   background(0);
   ellipse(250, yCoordinate, 100, 100);
}
```



- Assume the display window is 500 x 400
- Assume the ball is 100 in diameter.
- Assume the static xCoordinate is 250.
- Assume the background is called in the draw() method.
- Assume the starting yCoordinate is 300.

Lab03 - Challenge Exercise 1

```
float yCoordinate = 300;
boolean bounceUp = false;

void setup() {
  size(500,400);
  fill(255, 10, 10);
  stroke(255);
}
```

```
void draw() {
   background(0);
   ellipse(250, yCoordinate, 100, 100);

   if (bounceUp)
     // code to bounce the ball up

   if (!bounceUp)
      // code when ball is falling
}
```

- We need to track whether the ball is bouncing up or falling.
- To do this, we will use a boolean variable. It will be:
  - false if the ball is falling and
  - true if the ball is bouncing up

```
float yCoordinate = 300;
boolean bounceUp = false;

void setup() {
  size(500,400);
  fill(255, 10, 10);
  stroke(255);
}
```

```
void draw() {
   background(0);
   ellipse(250, yCoordinate, 100, 100);

   //ball is bouncing up
   if (bounceUp){
     if (yCoordinate > 100)
        yCoordinate = yCoordinate - 1;
     else
        bounceUp = false;
   }

   //ball is falling down
   if (!bounceUp){
     if (yCoordinate <= 350)
        yCoordinate = yCoordinate + 1;
     else
        bounceUp = true;
   }
}
```

Topics list

- There are three types of loop in programming:
  - While loops (covered in previous talk):
    - Counter controlled (n times)
    - Sentinel based (covered later in the course)
    - Flag based (covered later in the course)
  - For loops
  - Do While loops
- Comparative use of while and for loops
  - Lab03 - Challenge Exercise 1
  - Lab03 - Challenge Exercise 3

Lab03 - Challenge Exercise 3

- In a new sketch, draw a vertical line that is the height of your display window.
- This vertical line should start in the left most position of your display window and move right, pixel by pixel, until it reaches the right hand side of your display window.
- Upon reaching the right hand side, the vertical line should reverse direction and return, pixel by pixel, to the left hand side of the display window.
- As your vertical line is continually traversing the display window, your grayscale background should be varying very slightly in colour.

Lab03 - Challenge Exercise 3

- Assume a window size 300x400.
- Background is initially set to 120.
- Stroke weight is 4

```
float background = 120;

void setup()
{
    size(300,400);
    background(background);
    strokeWeight(4);
}
```

Lab03 - Challenge Exercise 3

- Draw a vertical line that is the height of your display window.
- Call background to clear the previously drawn line.

```
float background = 120;

void setup()
{
   size(300,400);
   background(background);
   strokeWeight(4);
}
```

```
void draw()
{
  background(background);
  line (xCoordinate, 0, xCoordinate, height);
}
```

Lab03 - Challenge Exercise 3

- Draw a vertical line that is the height of your display window.
- Call background to clear the previously drawn line.

```
float background = 120;
float xCoordinate = 0.0;

void setup(){
   size(300,400);
   background(background);
   strokeWeight(4);
}
```

```
void draw()
{
  background(background);
  line (xCoordinate, 0, xCoordinate, height);
}
```

Lab03 - Challenge Exercise 3

This vertical line should start in the left most position of your display window and move right, pixel by pixel, until it reaches the right hand side of your display window.

```
float background = 120;
float xCoordinate = 0.0;

void setup(){
   size(300,400);
   background(background);
   strokeWeight(4);
}
```

```
void draw(){
  xCoordinate = xCoordinate + 1;
  background(background);
  line (xCoordinate, 0, xCoordinate, height);
}
```

Lab03 - Challenge Exercise 3

As your vertical line is continually traversing the display window, your grayscale background should be varying very slightly in colour.

```
float background = 120;
float xCoordinate = 0.0;

void setup(){
   size(300,400);
   background(background);
   strokeWeight(4);
}
```

```
void draw(){
  xCoordinate = xCoordinate + 1;
  background = background + 0.5;
  background(background);
  line (xCoordinate, 0, xCoordinate, height);
}
```

Lab03 - Challenge Exercise 3

- Upon reaching the right hand side, the vertical line should **reverse direction** and return, pixel by pixel, to the left hand side of the display window.

- We need to keep track of the direction that the line should be moving i.e. is it going left-to-right, or has it reversed direction and gone from right-to-left?

- We will use a boolean variable to do this:
  - boolean reverseDirection will be initially set to false, indicating a left-to-right direction.
  - A value of true will indicate a right-to-left direction.

Lab03 – Challenge Exercise 3

```
float background = 120;
float xCoordinate = 0.0;
boolean reverseDirection = false;

void setup(){
   size(300,400);
   background(background);
   strokeWeight(4);
}
```

```
void draw()
{
 if (!reverseDirection){
    background = background + 0.5;
    xCoordinate = xCoordinate + 1;
  }
  else{
    background = background - 0.5;
    xCoordinate = xCoordinate - 1;
  }

  background(background);
  line (xCoordinate, 0, xCoordinate, height);
}
```

Lab03 - Challenge Exercise 3

- But, we have no code written that will set the flag to true e.g.

  boolean reverseDirection = true;

- Under what circumstances should the flag be set to true?

- And when should it be set back to false?

```
void draw(){
 if (xCoordinate == width)
    reverseDirection = true;
  if (xCoordinate == 0)
    reverseDirection = false;

if (!reverseDirection){
    background = background + 0.5;
    xCoordinate = xCoordinate + 1;
  }
  else{
    background = background - 0.5;
    xCoordinate = xCoordinate - 1;
  }

  background(background);
  line (xCoordinate, 0, xCoordinate, height);
}
```

```
float background = 120;
float xCoordinate = 0.0;
boolean reverseDirection = false;

void setup(){
   size(300,400);
   background(background);
   strokeWeight(4);
}
```

Questions?

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

Department of Computing and Mathematics
http://www.wit.ie/