for loops

COMP 102

Victoria University of Wellington

“for” loops

- A for loop is another way of repeating some code a number of times:

```
for (int i = 0  ;  i < 10;  i++ ) {
    UI.println(“I can count to “ + i + “!”);
}
```

Initialisation Condition Change

```
for (int x = 100  ;  x < 500;  x = x + 50 ) {
    UI.drawRect(x, 50, x + 20, 200);
    UI.drawEllipse(x – 20, 30, 40, 40);
}
```

for loops and while loops

- A for loop can be translated into a while loop:

```
for (int i = 0  ;  i < 10;  i++ ) {
    UI.println(“I can count to “ + I + “!”);
}
```

```
int i = 0;
while (i < 10) {
    UI.println(“I can count to “ + I + “!”);
    i++;
}
```

```
for (int x = 100  ;  x < 500;  x = x + 50 ) {
    UI.drawRect(x, 50, x + 20, 200);
    UI.drawEllipse(x – 20, 30, 40, 40);
}
```

```
int x = 100;
while (x < 500) {
    UI.drawRect(x, 50, x + 20, 200);
    UI.drawEllipse(x – 20, 30, 40, 40);
    x = x + 50;
}
```

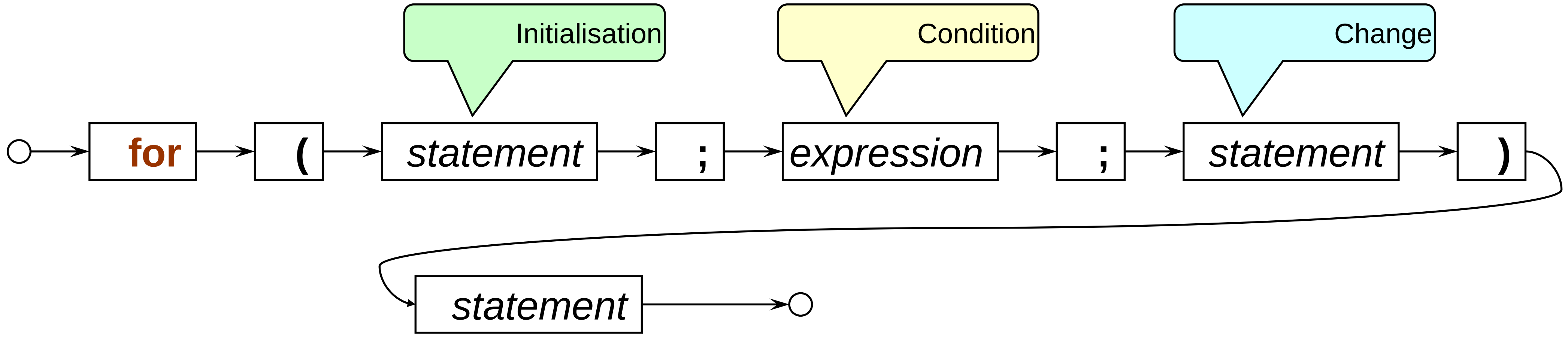
```
for (initialisation;  condition;  increment ) {
    body
}
```

```
initialisation
while (condition) {
    body
    increment
}
```

For loop

- For loop puts the
  - initialisation ← *<u>once</u>, <u>before</u> the loop body is run at all*
  - condition ← *tested each time, <u>before</u> loop body run*
  - increment ← *run each time, <u>after</u> loop body run*

  together, at the front of the loop



But the meaning is (almost) exactly the same as the while loop

(scope of variables in initialisation is different)

Using Numeric For: #1

- Print a table of numbers and their squares:

```
public void printTable(int max){
    UI.println("Table of integers and their squares");
    for (int num = 1;   num <= max;   num = num + 1 ) {
      UI.printf(“ %3d    %6d   %n”,  num,  (num*num));
    }
}
```

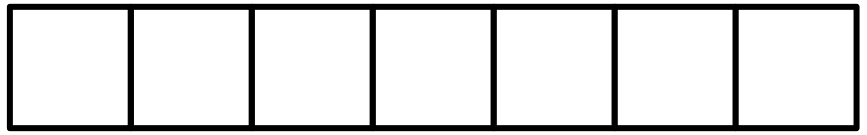
Using Numeric For: #2

Doesn't have to increment by 1 each time:

```
/**
 * Print each even number between start and end (inclusive)
 */
public void printEvenNumbers(int start, int end ){
    if (start%2==1 ) {    // make sure start is even
        start = start + 1;
    }
    for ( int num = start;   num <= end;  num = num + 2 )  {
        UI.println(num);
    }
}
```

Using Numeric For: #3

- Draw a row of squares:



```
/** Draws count squares in a horizontal row, starting at (left,top)  */
public void drawRowOfSquares (double left, double top, double size, int count){
    for  (int i = 0;  i < count;   i++  ) {
        double x = left + i * size;
        UI.drawRect(x, top, size, size);
    }
}
```

i++
is shorthand for
i = i + 1

Counting from 0 is often easier, especially for drawing stuff!

Count from 0 or 1?

Counted for loop: Can count from 0 or from 1

```
for (int n = 0; n < target; n++) {          OR          for (int n = 1; n <=max;  n++) {
    ⟨do actions ⟩                                           ⟨do actions ⟩
}                                                       }
```

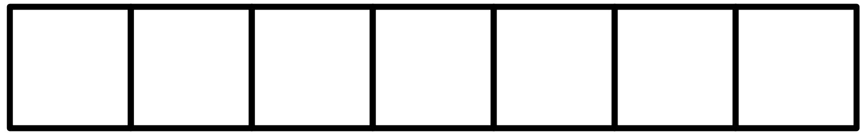
- If counting from 0,
  - n is the number of iterations that have been completed
  - Loop as long as n is less than target:
  - Good for drawing
  - Good for dealing with lists and arrays.

Off-by-one errors are common when you mix these two up.

- If counting from 1,
  - n is the iteration it is about to do
  - Loop as long as n is less than or equal to target:

Using Numeric For: #4

- Draw a row of squares:



```
/** Draws count squares in a horizontal row, starting at (left,top)  */
public void drawRowOfSquares (double left, double top, double size, int count){
    double right = left+count*SIZE;
    for  (double x = left;   x < right;   x = x + SIZE) {
        UI.drawRect(x, top, SIZE, SIZE);
    }
}
```

Note: this for statement is stepping through a sequence of doubles, rather than ints.