Scope of variables, Compound Assignment Statements, Printing

The scope of a local variable is the block it is declared in.

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

- Use of println(), text() in Processing
- Variable Scope
- Compound Assignment Statements

println() and text() in Processing

- To print a message to the console in Processing, use print() or println().
- Both take a String as input, (more later)..for now

```
println(“Hello World”);
println(“Hello ” + “World”);
println(“Hell” + “o World”);
```

All will produce the same output.

println() contd.

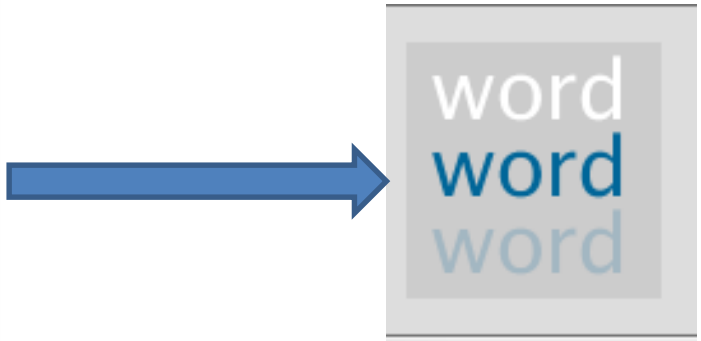
- We can introduce variables in the print statement also:

```
int myAge = 20;
println(“I am ” + myAge + “years of age”);
```

text() in processing

- text() can be used to draw text to the screen

```
textSize(32);
text("word", 10, 30);
fill(0, 102, 153);
text("word", 10, 60);
fill(0, 102, 153, 51);
text("word", 10, 90)
```



Text to be written (also in String format)

x, y co-ordinates on screen

Topics list

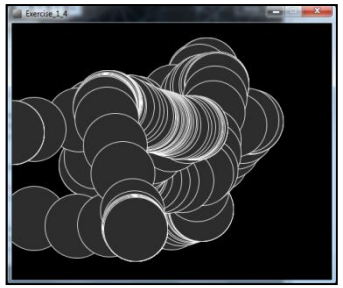
- Use of println(), text() in Processing
- Variable Scope
- Compound Assignment Statements

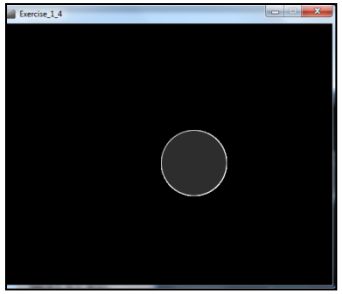
Recap: Processing Example 3.8

```
void setup() {
  size(500,400);
  background(0);
  stroke(255);
  fill(45,45,45);
}
```

```
void draw() {

  if (mousePressed) {
    background(0);
  }
  ellipse(mouseX, mouseY, 100, 100);
}
```

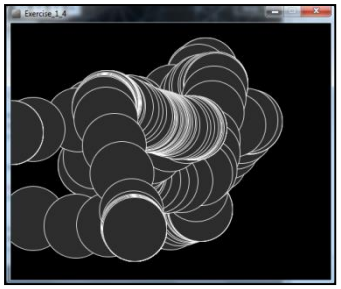


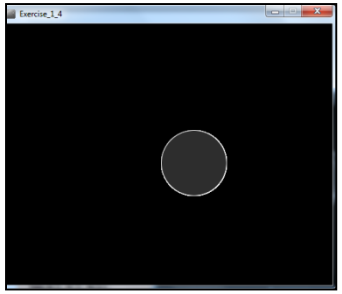


https://processing.org/tutorials/interactivity/

Processing Example 4.1

```
void setup() {
  size(500,400);
  background(0);
  stroke(255);
  fill(45,45,45);
}
```





```
void draw() {
  int diameter = 100;
  if (mousePressed) {
    background(0);
  }
  ellipse(mouseX, mouseY, diameter, diameter);
}
```

https://processing.org/tutorials/interactivity/

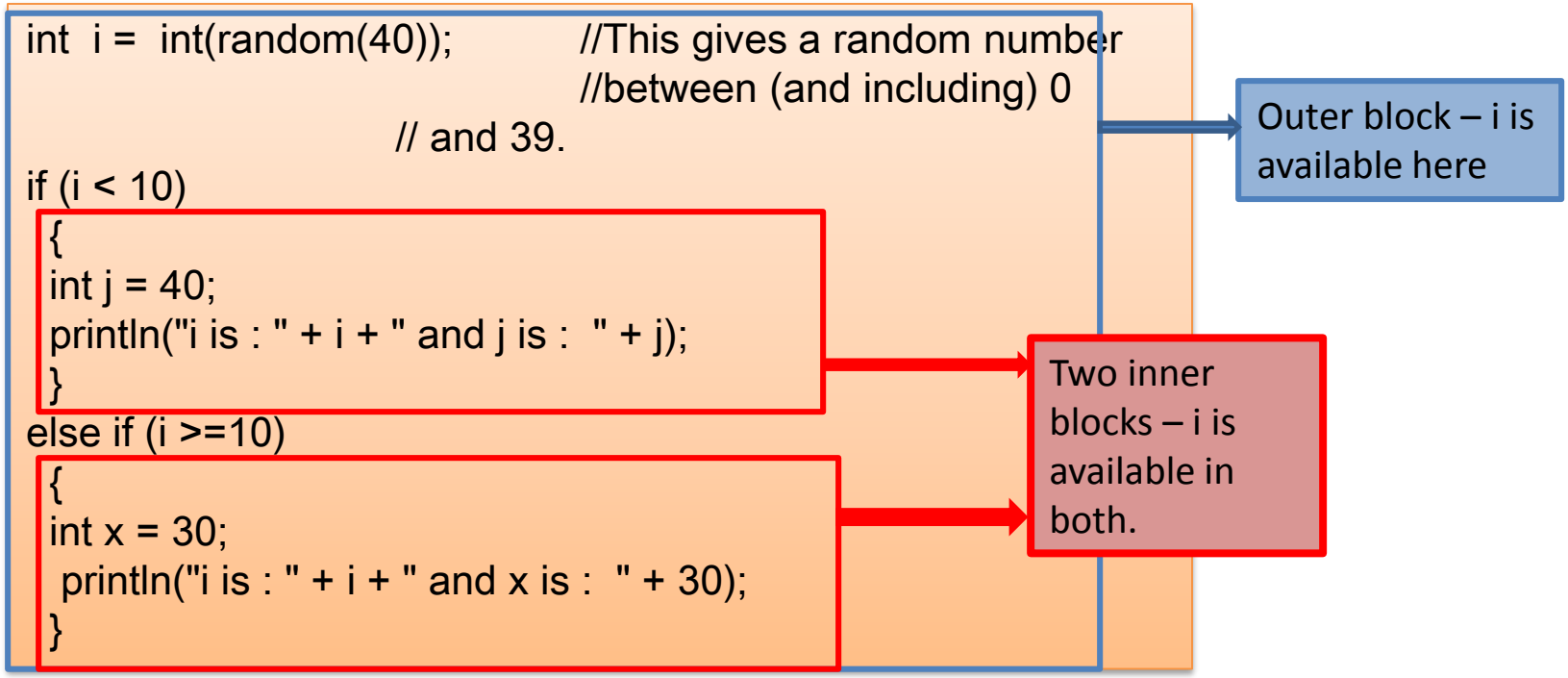
Local Scope – diameter variable

- The diameter variable is declared in the draw() function i.e. it is a local variable.
- It is only "alive" while the draw() function is running.
- Each time the draw() function:
  - finishes running, the diameter variable is destroyed.
  - is called, the diameter variable is re-created.

```
void draw() {
  int diameter = 100;
  if (mousePressed) {
    background(0);
  }
  ellipse(mouseX, mouseY, diameter, diameter);
}
```

Local variables – scope rules!

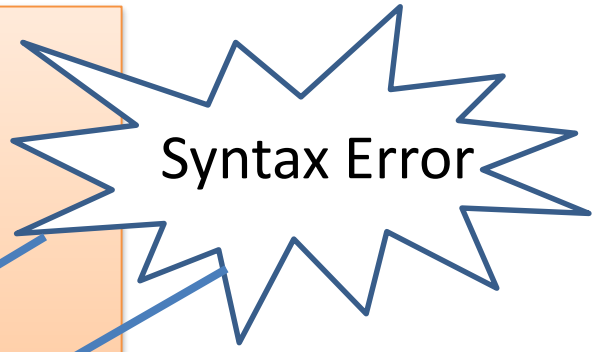
- The scope of a local variable is the block it is declared in. A block is delimited by the curly braces {}.
- A program can have many nested blocks.
- A variable must be declared before it is used.



Local variables – scope rules .. Contd.

- The lifetime of a local variable is the time of execution of the block it is declared in.
- Trying to access a local variable outside its scope will trigger a syntax error e.g.:

```
void draw()
{
    if (mousePressed)
    {
        int diameter = 100;
        background(0);
     }
     ellipse(mouseX, mouseY, diameter, diameter);
}
```

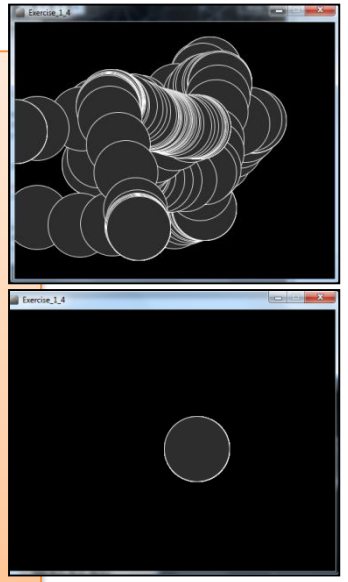


Processing Example 4.2

```
void setup() {
  size(500,400);
  background(0);
  stroke(255);
  fill(45,45,45);
}
```

We now want to reduce the diameter size by 10 each time the mouse is pressed. Is this correct?

```
void draw() {
  int diameter = 100;
  if (mousePressed) {
    diameter = diameter – 10;
    background(0);
  }
  ellipse(mouseX, mouseY, diameter, diameter);
}
```



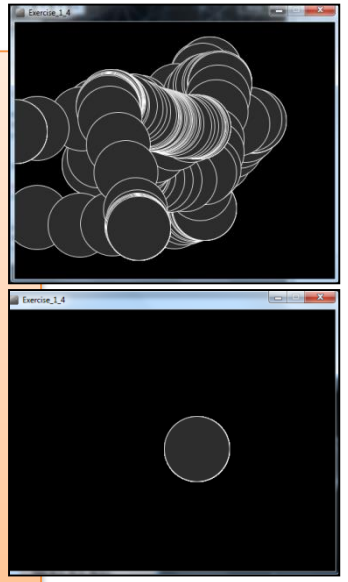
https://processing.org/tutorials/interactivity/

Processing Example 4.2

```
void setup() {
  size(500,400);
  background(0);
  stroke(255);
  fill(45,45,45);
}
```

We have a bug in our logic.
As the diameter variable is re-created each time draw() is called, its value will be reset to 100 and will lose our decrement of 10.

```
void draw() {
  int diameter = 100;
  if (mousePressed) {
    diameter = diameter – 10;
    background(0);
  }
  ellipse(mouseX, mouseY, diameter, diameter);
}
```



https://processing.org/tutorials/interactivity/

Global variables – scope rules!

- The scope of the diameter variable is too narrow; as soon as draw() finishes running, the local variable is destroyed and we loose all data.

- We need a diameter variable that lives for the lifetime is sketch i.e. a global variable.

Processing Example 4.3

```
int diameter = 100;

void setup() {
  size(500,400);
  background(0);
  stroke(255);
  fill(45,45,45);
}

void draw() {
  if (mousePressed) {
    diameter = diameter - 10;
    background(0);
  }
  ellipse(mouseX, mouseY, diameter, diameter);
}
```

We still have a bug in our logic.

The diameter variable is decreased each time we press the mouse. Correct!

However, what happens when we reach zero?

Processing Example 4.3

```
int diameter = 100;

void setup() {
  size(500,400);
  background(0);
  stroke(255);
  fill(45,45,45);
}

void draw() {
  if ((mousePressed) && (diameter > 20)){
    diameter = diameter - 10;
    background(0);
  }
  ellipse(mouseX, mouseY, diameter, diameter);
}
```

In ellipse, the width and height are absolute values (negative sign is dropped).

To handle this logic bug, we need to stop reducing by 10 when we reach a certain value, say 20.

Processing Example 4.3

```
int diameter = 100;

void setup() {
  size(500,400);
  background(0);
  stroke(255);
  fill(45,45,45);
  frameRate(20);
}

void draw() {
  if ((mousePressed) && (diameter > 20)){
    diameter = diameter - 10;
    background(0);
  }
  ellipse(mouseX, mouseY, diameter, diameter);
}
```

Did you notice that it seems the reduction is larger than 10 when we press the mouse?

Why? The default frame rate is 60 refreshes of the screen per second i.e. draw() is called 60 times per second.

You can change the frame rate by calling the frameRate() function.

Topics list

- Use of println(), text() in Processing
- Variable Scope
- Compound Assignment Statements

Compound Assignment Statements

	Full statement	Shortcut
Mathematical shortcuts	x = x + a;	x += a;
	x = x - a;	x -= a;
	x = x * a;	x *= a;
	x = x/a;	x /=a;
Increment shortcut	x = x+1;	x++;
Decrement shortcut	x = x - 1;	x--;

Questions?

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

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