



Programming Fundamentals 1

Produced by Mr. Dave Drohan (david.drohan@setu.ie)
Dr. Siobhán Drohan
Ms. Mairead Meagher

Department of Computing & Mathematics
South East Technological University
Waterford, Ireland

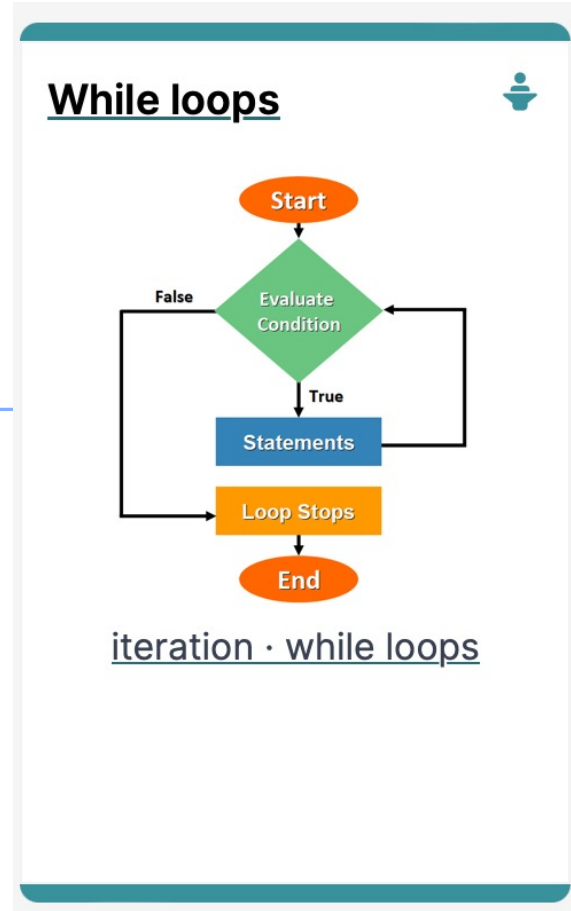
setu.ie





Introduction to Processing

Iteration in Programming – While Loops





Agenda

- Repetition in Programming – Intro to looping
- Use of loops (while loops)



Repetition in Programming – Intro to looping





Recap: Boolean conditions

- A boolean condition is an expression that evaluates to either true or false e.g.

`mouseX < 50`

- Boolean conditions can be used to control:
 - Selection i.e. if statements and
 - Iteration i.e. loops (we will look at these now).



Repetition in Programming

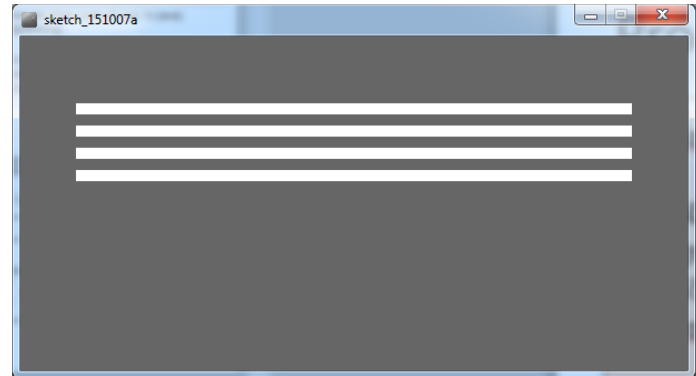
- Computers are very good at repetition.

- Example:
 - `calculate pay` for 1000 employees.
 - You should use the same `calculate pay` algorithm 1000 times.
 - You don't write the `calculate pay` algorithm 1000 times; *instead* you include it in a loop.



Form of loop

- Draw a rectangle 4 times that has a gap of 10 pixels between each one.



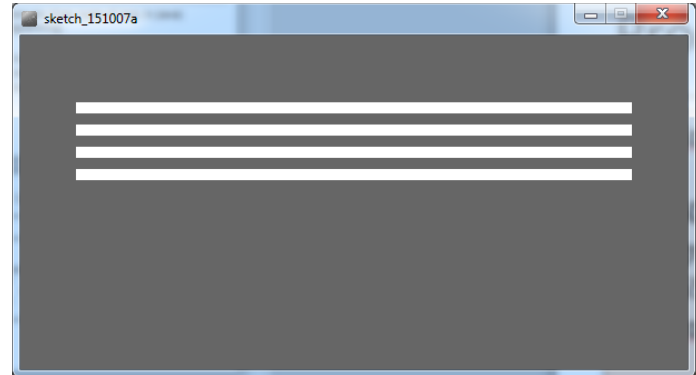


Form of loop

□ Draw a rectangle 4 times that has a gap of 10 pixels between each one.

- Without loop:

```
rect(50, 60, 500, 10);  
rect(50, 80, 500, 10);  
rect(50, 100, 500, 10);  
rect(50, 120, 500, 10);
```

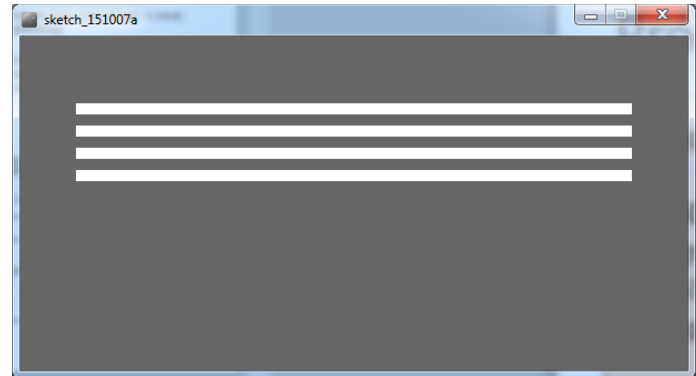




Form of loop

□ Draw a rectangle 4 times that has a gap of 10 pixels between each one.

- With a loop:
 - ◆ do this 4 times (adding 20 onto the yCoordinate variable each time).



```
rect(50, yCoordinate, 500, 10);
```



Form of loop

□ Draw a rectangle 4 times that has a gap of 10 pixels between each one.

- With a loop:
 - ◆ do this 4 times (adding 20 onto the yCoordinate variable each time).

We will learn a little more about loops and then we will write the code to solve this problem.

```
rect(50, yCoordinate, 500, 10);
```



Use of loops (while loops)





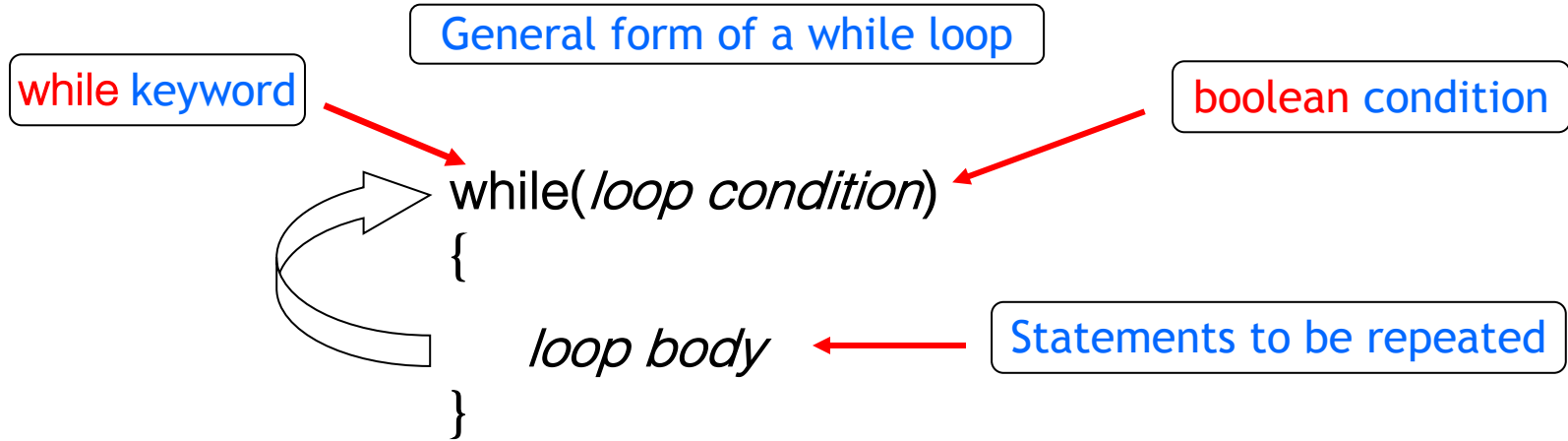
Loops in Programming

□ There are three types of loop in (Java) programming:

- **while** loops
- **for** loops
- **do while** loops



while loop pseudo code



Pseudo-code expression of the actions of a while loop

while we wish to continue, do the things in the **loop body**

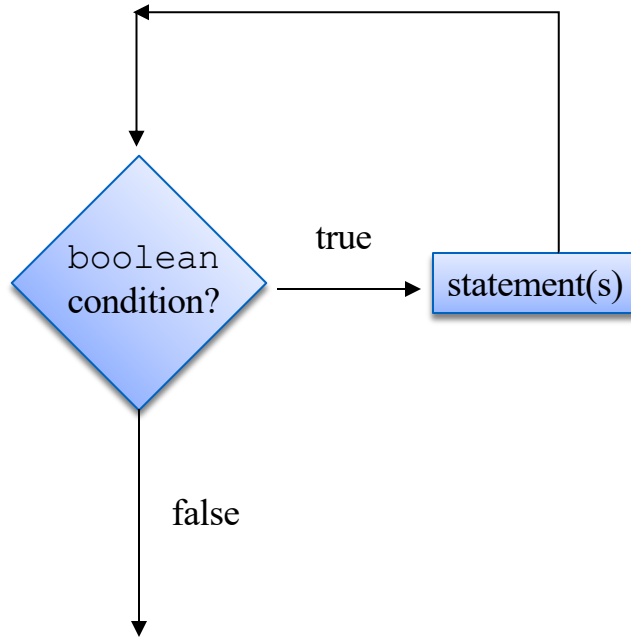


Construction of while loop

```
Declare and initialise loop control variable (LCV)
while(boolean condition based on LCV is true)
{
    "do the job to be repeated"
    "update the LCV"
}
```

This structure should always be used

while loop Flowchart



```
int yCoordinate = 60;
```

```
int i = 0; //i is the LCV
```

```
while(i < 4)
```

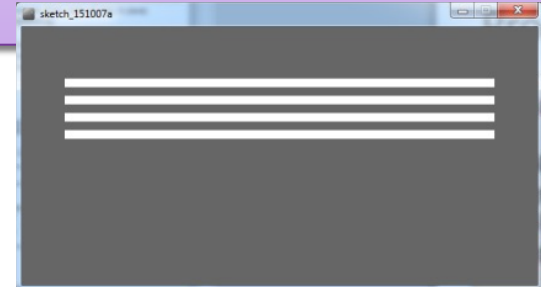
```
{
```

```
    rect(50, yCoordinate, 500, 10);
```

```
    yCoordinate += 20;
```

```
    i++;
```

```
}
```





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++;
}
```





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++;
}
```

Q: Could we remove the `yCoordinate` variable and rework the code to still produce the four lines using the while loop?





Processing Example 4.6

```
size(600, 300);  
background(102);  
fill(255);  
noStroke();  
  
int i = 60;  
while(i <= 120)  
{  
    rect(50, i, 500, 10);  
    i += 20;  
}
```

A: Yes. Here is the solution with *no* **yCoordinate** variable.



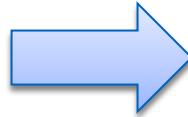


Some Study Exercises

This basic while loop, produces this output.

```
int i = 1;

while (i <=5)
{
    println("Hello World");
    i++;
}
```



```
Hello World
Hello World
Hello World
Hello World
Hello World
```



Some Study Exercises

1. Change the code so that “Hello World” is printed out 10 times.
2. Change the code so that the numbers from 1 to 10 (inclusive) are printed out, one line at a time.
3. Change the code so that the numbers from 10 to 1 are printed out.

Questions?





References

- Reas, C. & Fry, B. (2014) Processing – A Programming Handbook for Visual Designers and Artists, 2nd Edition, MIT Press, London.

