Iteration in Programming

for loops, do while loops

Produced by:

Dr. Siobhán Drohan,
Ms. Mairead Meagher,
Ms. Siobhán Roche.

Ollscoil Teicneolaíochta an Oirdheiscirt
South East Technological University

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

Loops

- There are three types of loop in programming:
  - while loops:
    - Counter controlled (n times) - covered in previous talk
    - Sentinel based (covered later in the course)
    - Flag based (covered later in the course)
  - for loops (this slide deck)
  - do While loops (this slide deck)

Topics

1. For loops

2. Comparison of while and for loops

3. do while loops (this slide deck)

4. Comparison of while and do while

For loop pseudo-code

General form of a for loop

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

Recap: Loop Example

```
int i = 1;
   while (i <= 4)
   {
     System.out.println("Hello World");
     i++;
   }
```

This was a slide from the previous talk. We used a while loop to repeatedly print "Hello World" to the console.

```
NoLoop.main({ });
Hello World
Hello World
Hello World
Hello World
```

For loop – Example 1

```
for (int i = 1; i<=4; i++) {
  System.out.println("Hello World");
 }
```

This code does the same as the previous slide, except that we use a different loop:

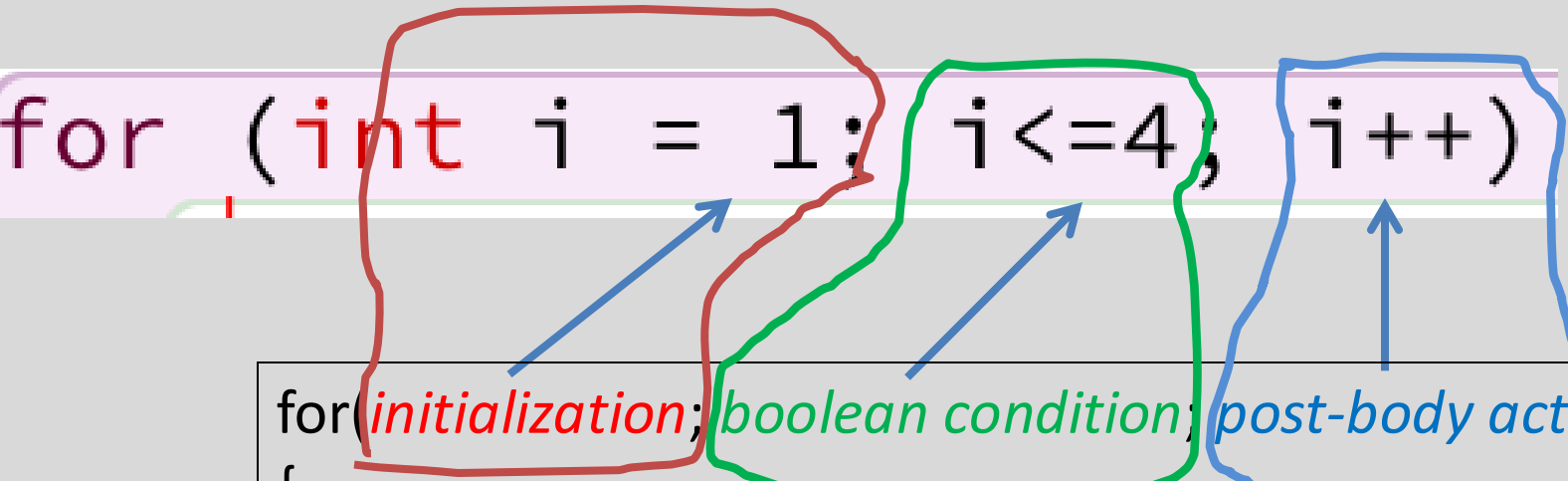
for

```
ForLoop.main({ });
Hello World
Hello World
Hello World
Hello World
```

For loop syntax

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

For loop syntax



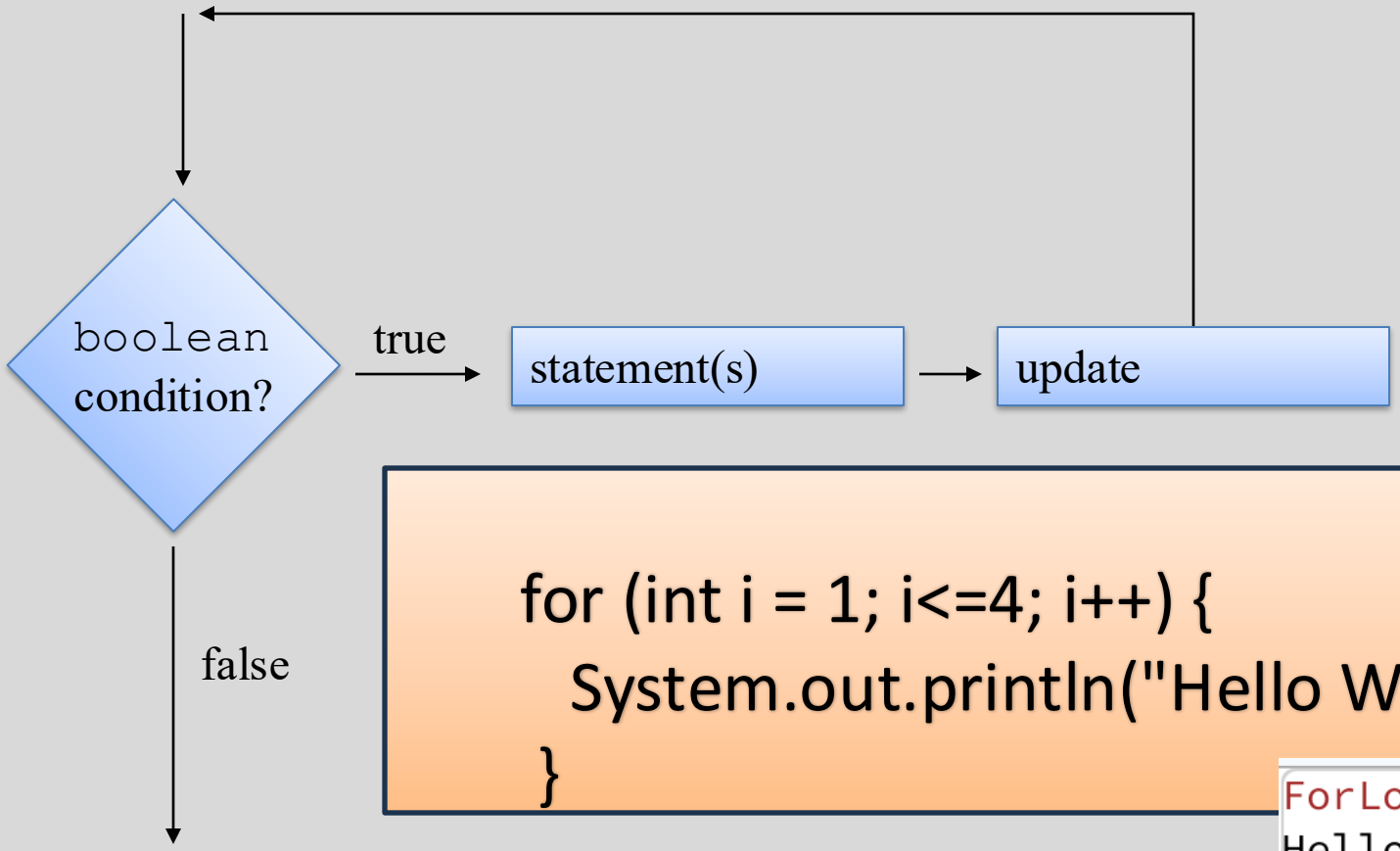

```
for(initialization; boolean condition; post-body action)
{
    statements to be repeated
}
```

For loop syntax

```
for (int i = 1; i<=4; i++)
```

initialization	int i = 1;	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 = 1; i<=4; i++) {
  System.out.println("Hello World");
 }
```

```
ForLoop.main({ });
Hello World
Hello World
Hello World
Hello World
```

For loop: all parts are optional

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

This is an infinite loop…

Loops can be nested

```
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);
        }
}
```

```
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
```

Topics

1. For loops

2. Comparison of while and for loops

3. do while loops (this slide deck)

4. Comparison of while and do while

```
NoLoop.main({ });
Hello World
Hello World
Hello World
Hello World
```

for versus while

for loop

```
for(int i = 1; i <= 4; i++)
{
    System.out.println("Hello World");
}
```

while loop

```
int i = 1;
while(i <= 4)
{
    System.out.println("Hello World");
    i++;
}
```

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

int i = 1 is the **initialisation**.

i <= 4 is the boolean condition i.e. the **test**

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

Topics

1. For loops

2. Comparison of while and for loops

3. do while loops (this slide deck)

4. Comparison of while and do while

Construction of **do** loop

Declare and initialise loop control variable (LCV)

```
do {
    "do the job to be repeated"
    "update the LCV"
} while (boolean condition based on LCV is true);
```

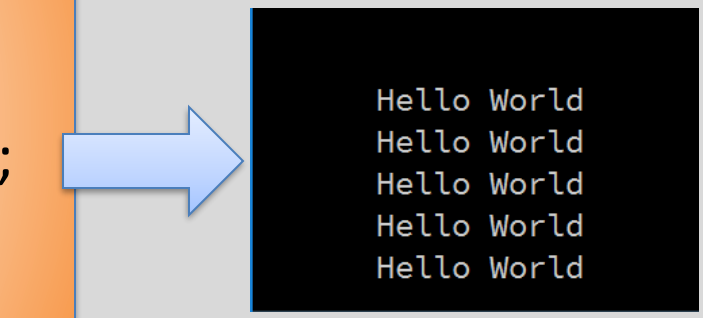
This structure should <u>always</u> be used

Simple **do** Loop

This basic do loop, produces this output.

```
int i = 1;

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



Topics

1. For loops

2. Comparison of while and for loops

3. do while loops (this slide deck)

4. Comparison of while and do while

Do versus While

- Use a **do** loop
  when the code that may be repeated
  must be run at least once

- Use a **while** loop
  when the code that may be repeated
  may never run

Questions?