Loops

#6

CS 0007
Introduction to
Computer Programming

Luís Oliveira

Summer 2020

Loops

- Repeat an action… until a stop condition is met:
  - There must be a condition that stops execution…
  - Otherwise the code will never end
- There are different loops in Java
  - While loops
  - Do-while loops
  - For loops

While loops

- While loops check exit condition at the top

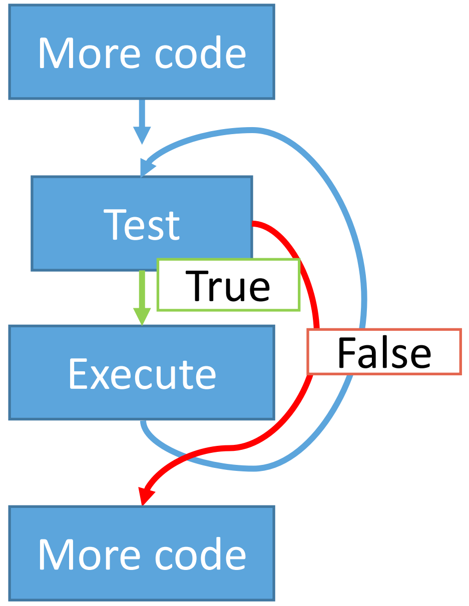
```
boolean condition;
while(condition) {
    runCode();
}
```

1. Test the condition
    1. If the condition if false, leave the loop
    2. If the condition is true, execute the body and go back to the top

Example

- The loop will run WHILE the number is not negative
  - So the loop will exit when the opposite is true: the number is negative

```
int number = 10;
while(number>=0) {
    System.out.println(number);
    number-=1;
}
```



- It always goes back to the top to test
  - When it finishes the loop body!

Do…while loops

- Do…while loops check exit condition at the bottom
  - They always execute one!

```
boolean condition;
do {
    runCode();
} while(condition);
```

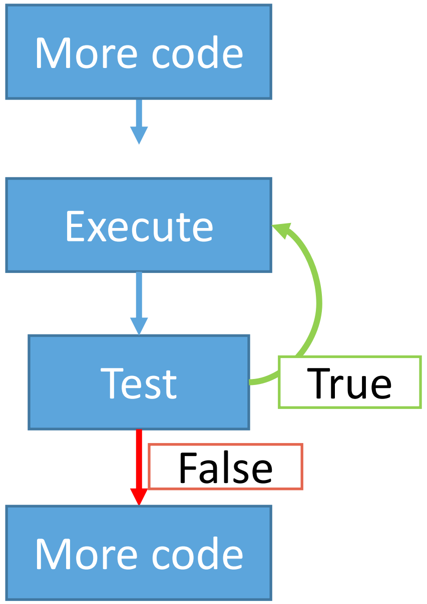
Don't forget the semicolon

1. Execute the body
2. Test the condition
   1. If the condition if false, leave the loop
   2. If the condition is true, go back to the top

Example

- **The loop will run WHILE the number is not negative**
  - So the loop will exit when the opposite is true: the number is negative

```java
int number = 10;
do {
    System.out.println(number);
    number-=1;
} while(number>=0);
```



- **It always goes back to the top to test**
  - When it finishes the loop body!

For loops

- Just a disguised while loop (a specific type of while)

```
int i = 0;
while(i<10) {
    runCode();
    i++;
}
```

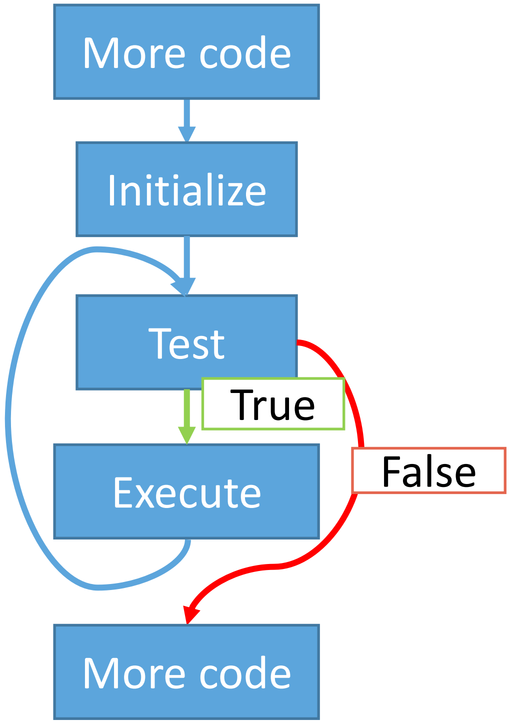
```
for(int i = 0; i<10; i++) {
    runCode();
}
```

1. Initialize loop variable
2. Test
   1. If the condition if false, leave the loop
   2. If the condition is true, execute the body and go back to the top
3. Increment the loop variable, and go back to the top

Example

- The loop will run WHILE “i” is smaller than 10
  - So it will exit when the opposite is true: the number is larger or equal to 10

```
for(int i = 0; i<10; i++) {
    System.out.println(number);
}
```



- It always goes back to the top to test
  - When it finishes the loop body!