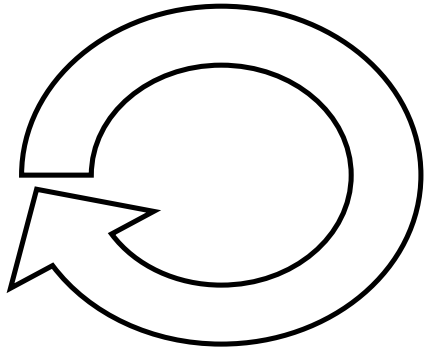


Looping Structures



CompSci 4

Iteration

10.1

The Plan

While not everyone understands:

1. Motivate loops
2. For loops
3. While loops
4. Do-while loops
5. Equivalence
6. Application of Simulated Collision
7. Practice Problems

CompSci 4

Iteration

10.2

Motivation

Why loop?

**Sometimes you need to do things again, and again, and
again, and again, and again, and again, and again,
again, and again, and again, and again, and again, and
again, and again, and again, and again, and again, and
again, and again, and again, and again, and again, and
again, and again, and again, and again, and again, and
again...and finally you get tired of typing.**

CompSci 4

Iteration

10.3

Motivation

Okay, so that's not all. You also loop in order to:

- ❖ **Group repeatedly executed code for uniformity**
- ❖ **Make the number of repetitions easily changeable**
- ❖ **Repeat events which the number of executions is known only dynamically**
- ❖ **Combine with selection statements to make more complex algorithms**

CompSci 4

Iteration

10.4

while Loop

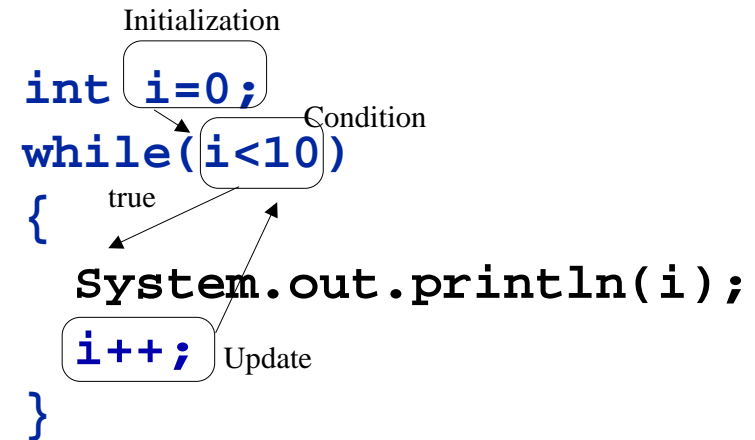
```
int i=0;
while(i<10)
{
    System.out.println(i);
    i++;
}
```

CompSci 4

Iteration

10.5

while Loop



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

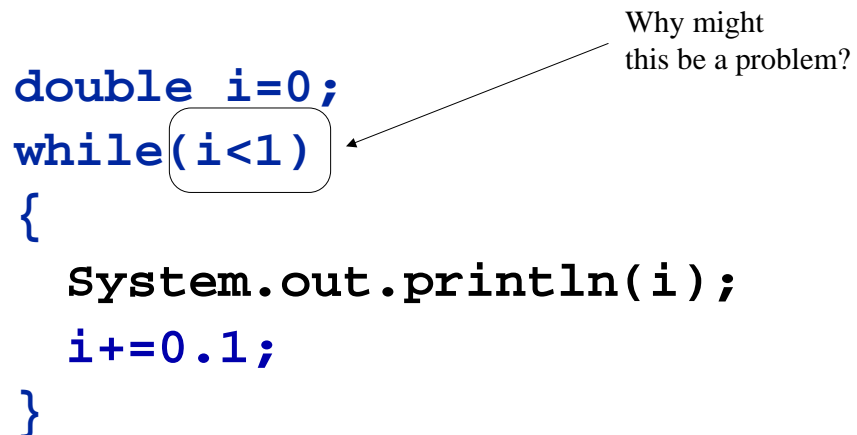
The diagram illustrates the execution flow of a while loop. It starts with 'Initialization' pointing to `i=0`. Then, the 'Condition' `i<10` is checked. If 'true', the loop body `System.out.println(i);` is executed, followed by the 'Update' `i++`, which then loops back to the condition check.

CompSci 4

Iteration

10.6

while Loop



```
double i=0;
while(i<1)
{
    System.out.println(i);
    i+=0.1;
}
```

The diagram shows a while loop with a condition `i<1` that is highlighted and pointed to by an arrow from the text 'Why might this be a problem?'. This highlights a potential issue with floating-point precision in loop conditions.

CompSci 4

Iteration

10.7

for Loop

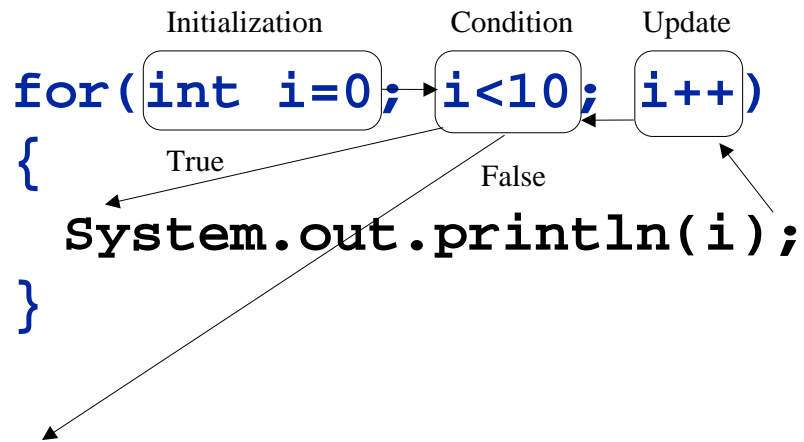
```
for(int i=0; i<10; i++)
{
    System.out.println(i);
}
```

CompSci 4

Iteration

10.8

for Loop

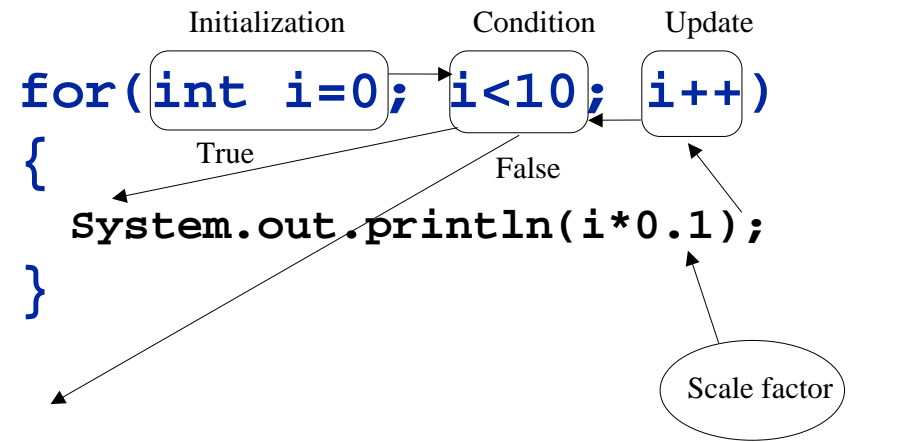


CompSci 4

Iteration

10.9

for Loop

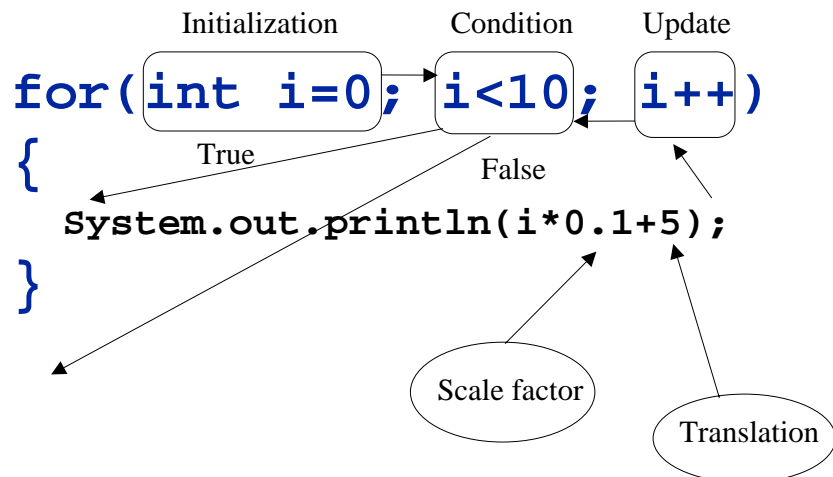


CompSci 4

Iteration

10.10

for Loop



CompSci 4

Iteration

10.11

Equivalence of Loops

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

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

CompSci 4

Iteration

10.12

do-while Loop

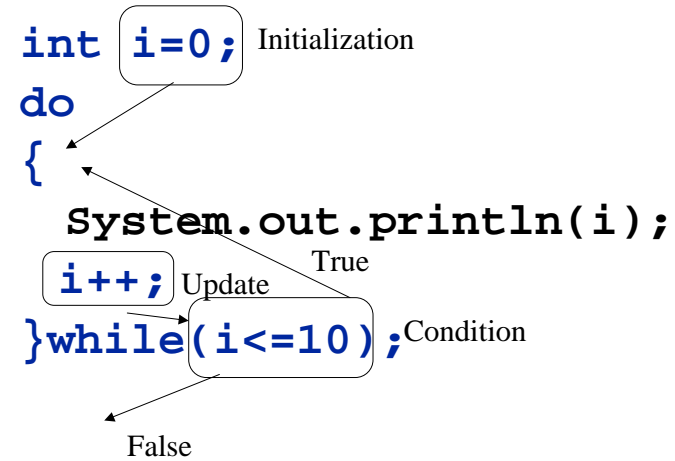
```
int i=0;
do
{
    System.out.println(i);
    i++;
}while(i<=10);
```

CompSci 4

Iteration

10.13

do-while Loop



CompSci 4

Iteration

10.14

do-while Loop

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

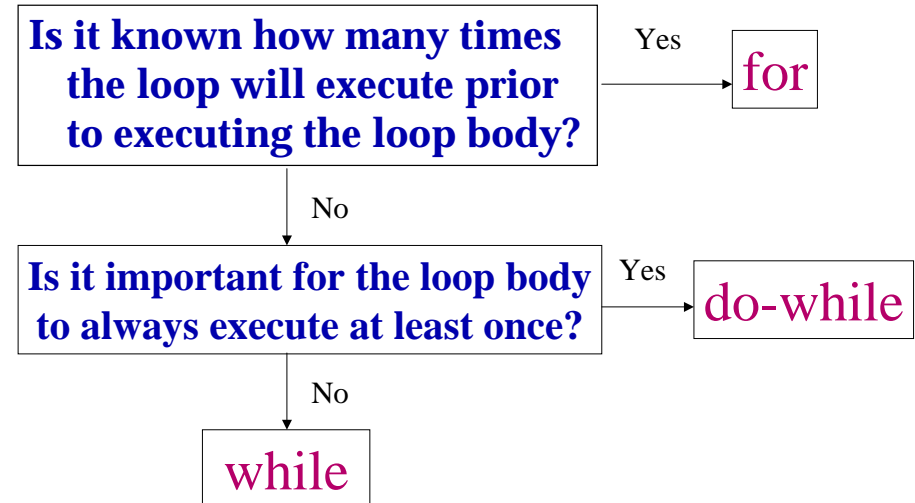
Notice this semicolon
was not here in the while loop!

CompSci 4

Iteration

10.15

When to use which loop?



CompSci 4

Iteration

10.16

When to use which loop?

Real answer:

Use which ever structure is most convenient, because all loop structures can be represented as any other loop structure.

Why are there multiple loop structures then?

Simple answer – for the programmer's convenience.

Note: Java 5.0 offers another form of the `for` loop

We will cover this at a later point

Practice Problems

- ❖ Write a loop to print out from 10 to 100 inclusive counting by 10s
- ❖ Write a loop that starts with an arbitrary double x and divides it by 2 repeatedly until it is less than 1. Output the number of times the loop executed. What is being computed?
- ❖ Write a loop that sums the first x integers where x is a positive integer. Print out the results.
- ❖ Write a loop that takes an integer x starting with value 1 and doubles x so long as x is positive. Bonus question: why doesn't this loop infinitely? Super Bonus question: why does it loop infinitely when x is a double?