# Iteration:
# For and While Loops
## (and the increment statement)

IST 256

Application Programming for Information Systems

# Why do we need Loops?

- Loops allow us to repeat the instructions in a section of code many times
  - In order to compute some quantity that is defined by repetition
    - Examples: interest on a savings account, the factorial function
  - In order to apply the same computation to every item of data
    - Examples: a company wants to compute the salary for every employee in its database, …
    - We'll do these types of loops after we have data structures, such as arrays, to hold the data
  - Loops introduce another type of instruction sequencing

# Increment Statement

- A very common assignment statement is to add 1to a variable, e.g. to count how many times a loop is executed:

```
int count = 0; . . .
count = count + 1; . . .
```

- The increment statement uses an operation written as ++ to abbreviate adding 1 to a variable

  Equivalent:

```
int count = 0; . . .
count ++;
```

- There is also an operator consisting of two minus signs that can be used to subtract 1 from a variable

```
count --;
```

# Iteration – For Loop

- Sometimes we want to repeat an operation a number of times

  Sum of: 1 + 2 + 3 + 4 + 5 + 6 +…….. + 100

- Find a pattern of statements that can be repeated

  result = 0; result = result + 1; result = result + 2; . . .

- Put these in a For loop
  - The loop will repeat the pattern of instructions for the number of times needed

# For Loop

```
int result;
result = 0;

for (int i = 1; i <= 100; i++)
{
        result = result + i;
}

System.out.println("Sum is: " + result);
```

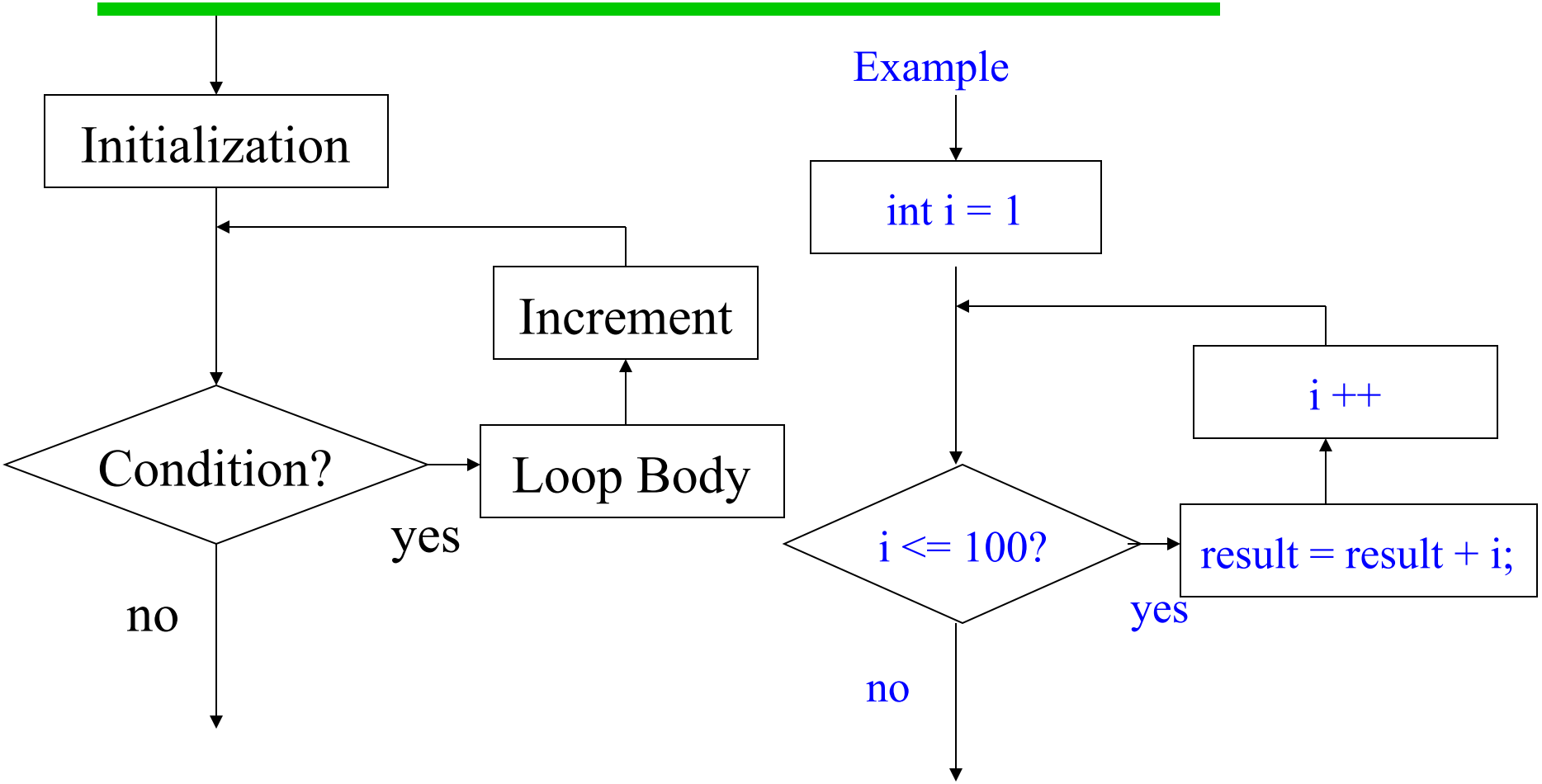
Initialization:
Start with 1

Condition:
Repeat as long as this is true

Increment:
Add 1 for each repetition

Loop Body:
Code is repeated

# How the For Loop is computed



# Iteration – While Loop

- But suppose that we don't know how many times to repeat the operation; we may want to achieve some goal:

  Sum of: 1 + 2 + 3 + 4 + 5 + 6 +…….. ??

  How many times to add up to 1000?

- The same pattern of statements can be repeated until we achieve a goal by using a While loop
  - The while loop will repeat the pattern of statements as long as the "continuing" condition is met (in this example, we have not yet reached the goal of summing to 1000)

# While Loop

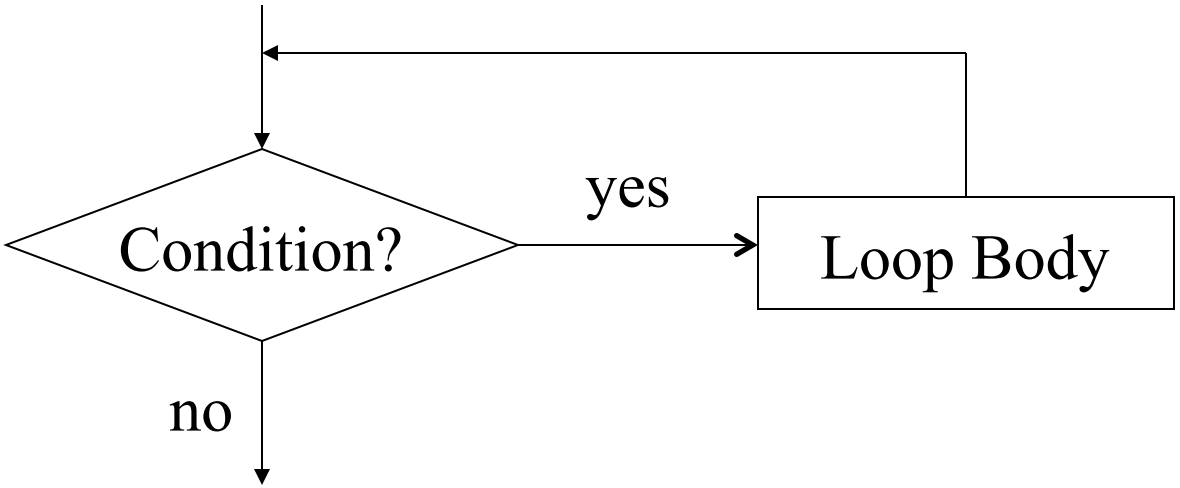
```
int result = 0;
int count = 1;

while (result < 1000)
{
        result = result + count;
        count = count + 1;
}

System.out.println("number of times: " + count);
```

Condition:
Repeat as long as this is true

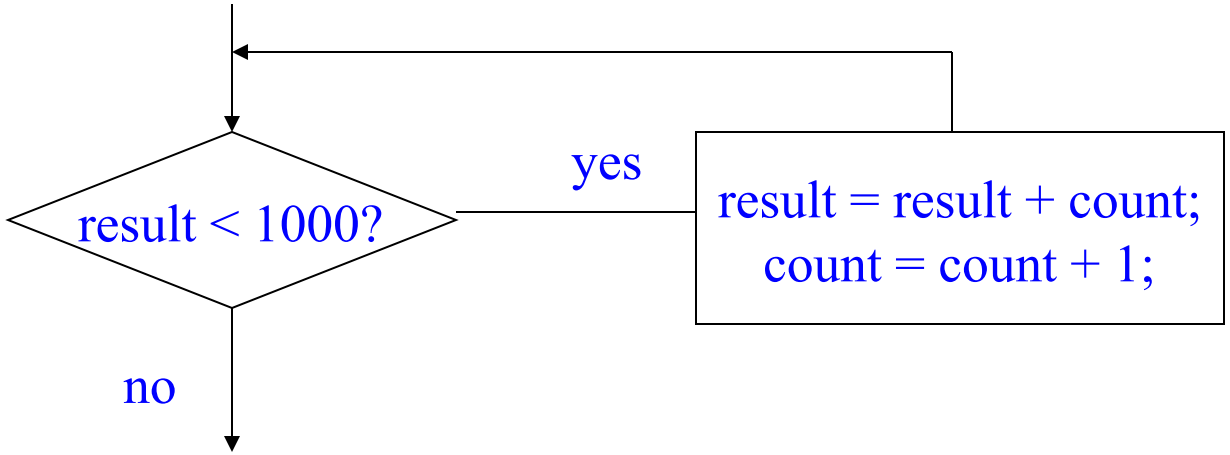
Loop Body:
Code is repeated

Note:
Count variable for iterations is not automatically incremented

# How the While Loop is computed




Example



# Comparison of For and While

- Both loops repeat a set of statements
- The *for* loop has an index variable that is automatically incremented
  - In the *while* loop, if you need a count variable, you must put the increment statement in the loop body
- In the *for* loop, you know ahead of time how many times the loop will be repeated
  - This is not required for the while loop
- In the *while* loop, the body of the loop must be written so that the condition will eventually become true
  - If not, the while loop can run on forever and never get done

```
while ( x < 10 )
        {    x = 5 ;   }  // the value of x never becomes >= 10
```