Java Control: Repetition

CPSC 219: Introduction to Computer Science for Multidisciplinary Studies II
Fall 2023

Jonathan Hudson, Ph.D.
Instructor
Department of Computer Science
University of Calgary

Wednesday, 12 September 2023

UNIVERSITY OF CALGARY

Loop types

Post-test loops

- Checking the looping condition *after* executing the body of the loop.
- The loop body executes at least *one* time.

Pre-test loops

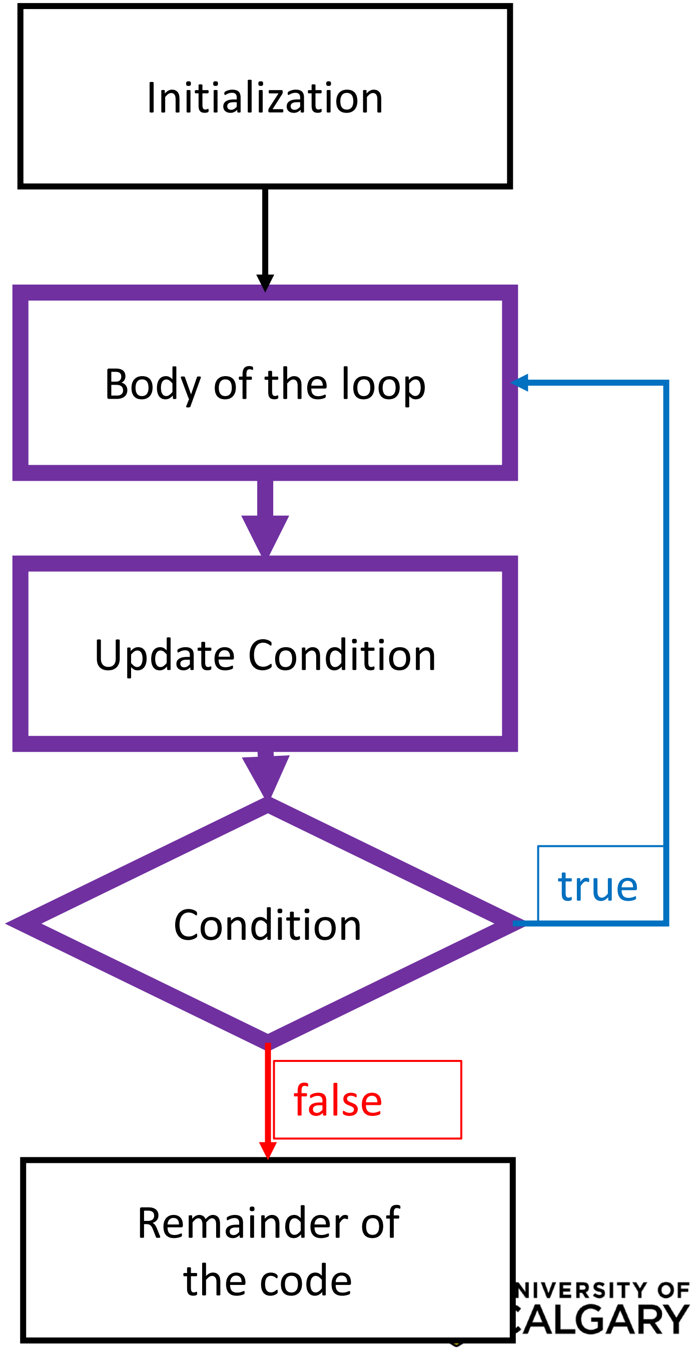
- Checking the looping condition *before* executing the body of the loop.
- The loop body executes *zero or more* times.

Post-Test Loop (do-while)

In Java! (not in Python)

Post-test loops

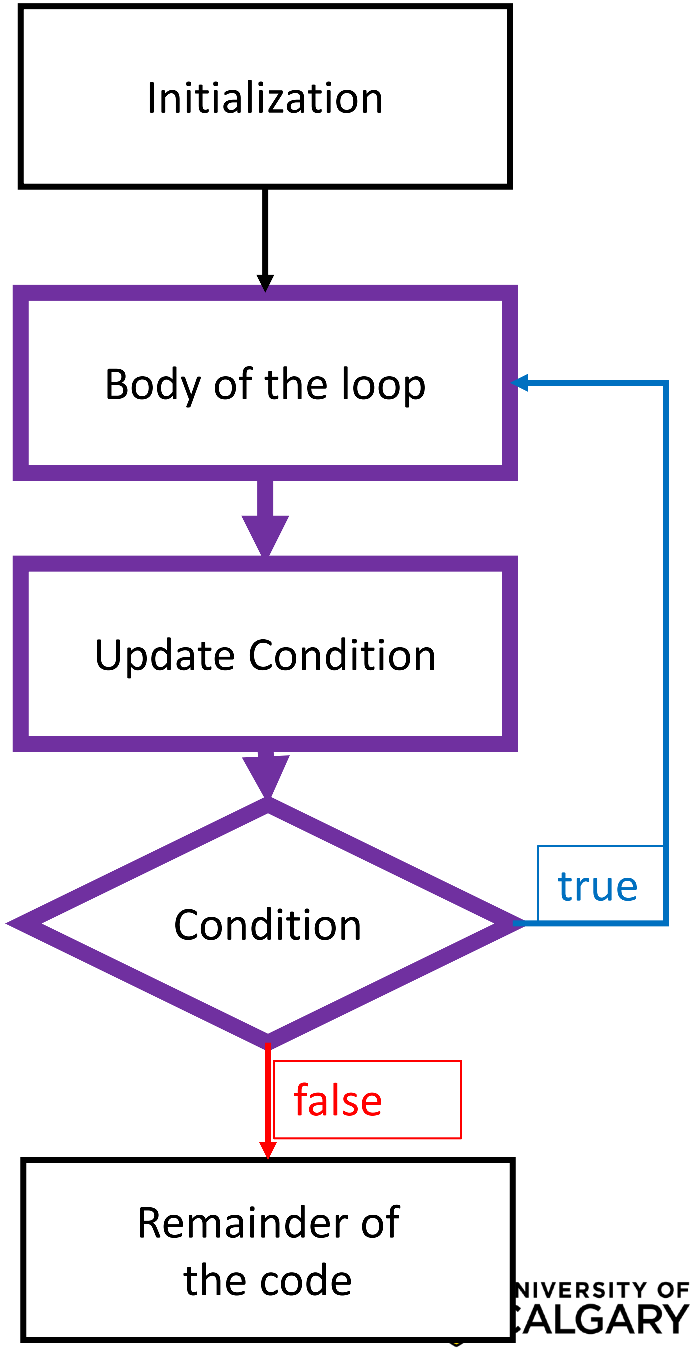
1. Initialize the loop control
2. Execute the body of the loop
   (the part to be repeated)
3. Update the loop control
4. Check the condition
   - **False** → stop the loop and go to the rest of program
   - **True** → repeat from step 2



do-while loops – (simple)

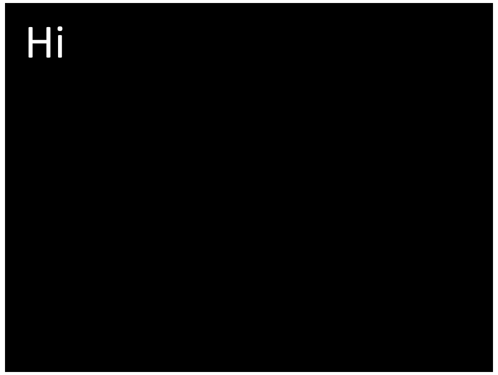
```
int x = 1;
do {
        System.out.println(“Hi”);
        x = x + 1;
} while (x <= 5);
```

```
Hi
Hi
Hi
Hi
Hi
```

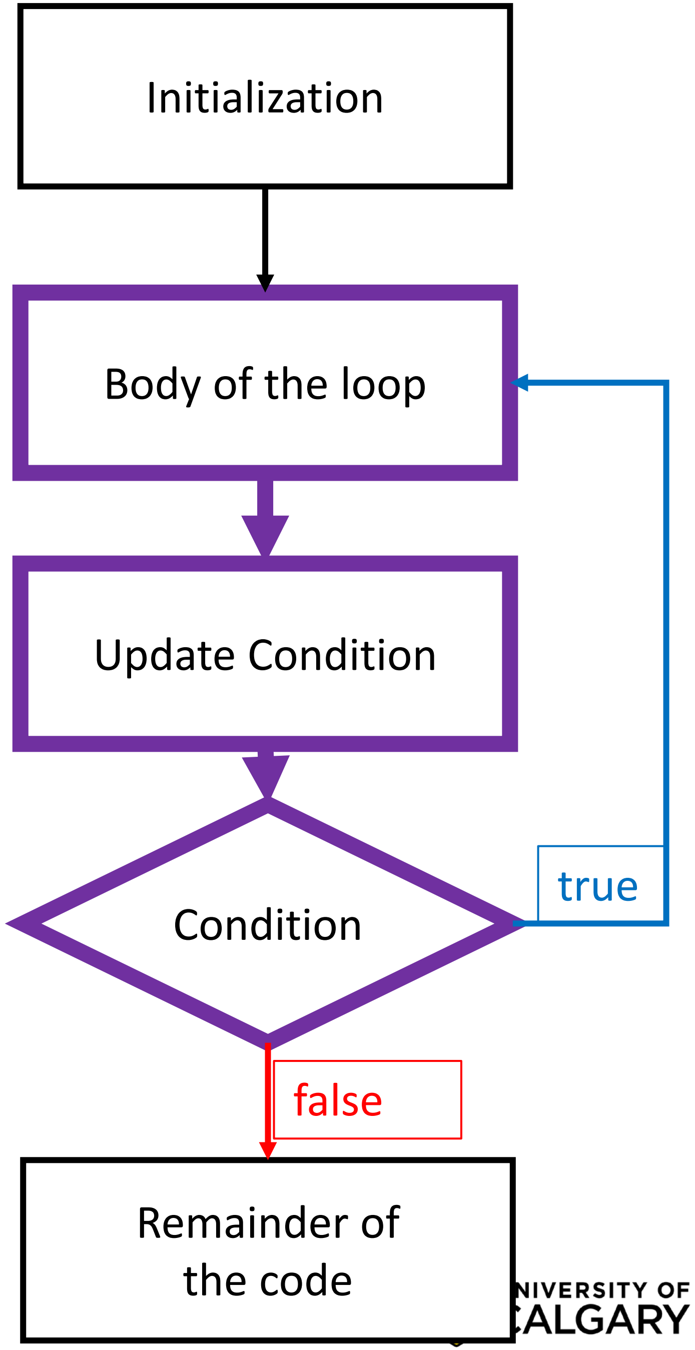


do-while loops – (always loops once)

```
int x = 1;
do {
        System.out.println("Hi");
        x = x + 1;
} while (x <= -999);
```



```
Hi
```



Pre-Test Loop (for/while)

In Java!

Pre-test loops

1. Initialize loop control

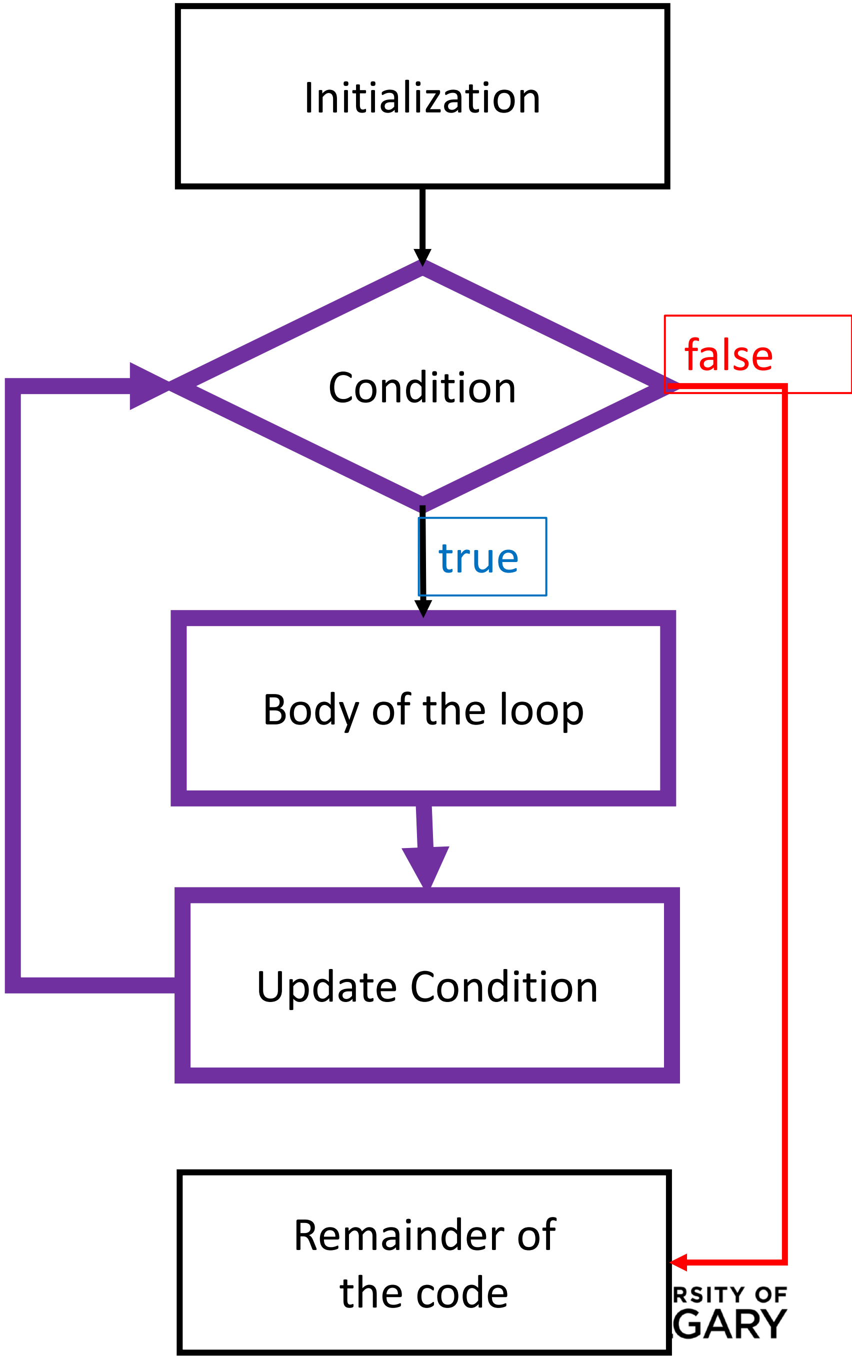
2. Check the condition

   False→ stop the loop, skip the body,
   go to the rest of program

   True → Execute the body of the loop

4. Update the loop control

5. Repeat from step 2

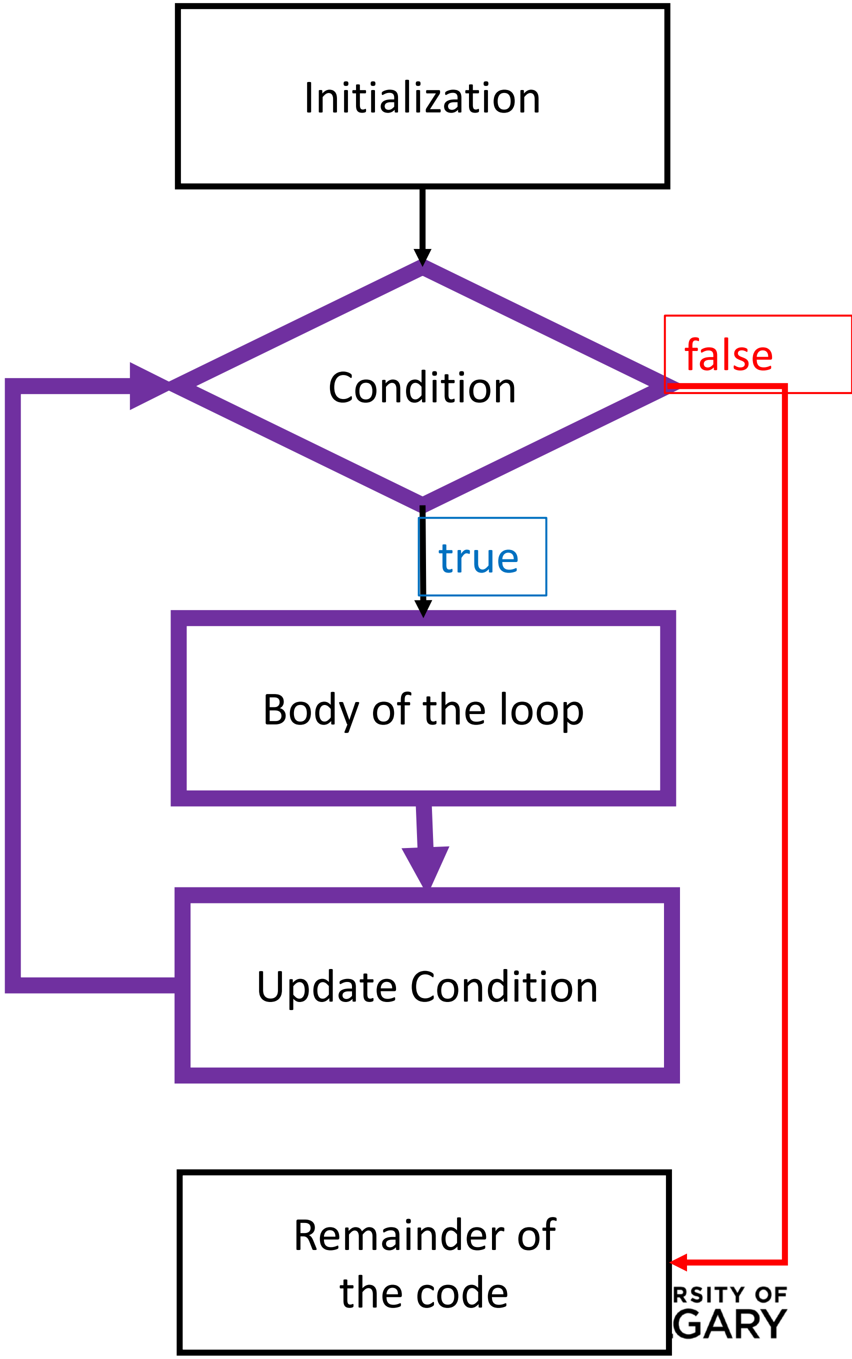


RSITY OF
GARY

while loop (simple)

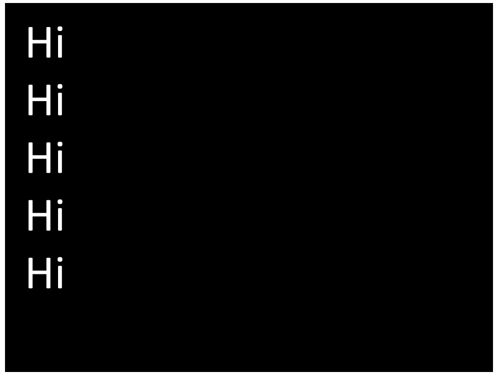
```
int x = 1;
while (x <= 5){
        System.out.println("Hi");
        x = x + 1;
}
```

```
Hi
Hi
Hi
Hi
Hi
```

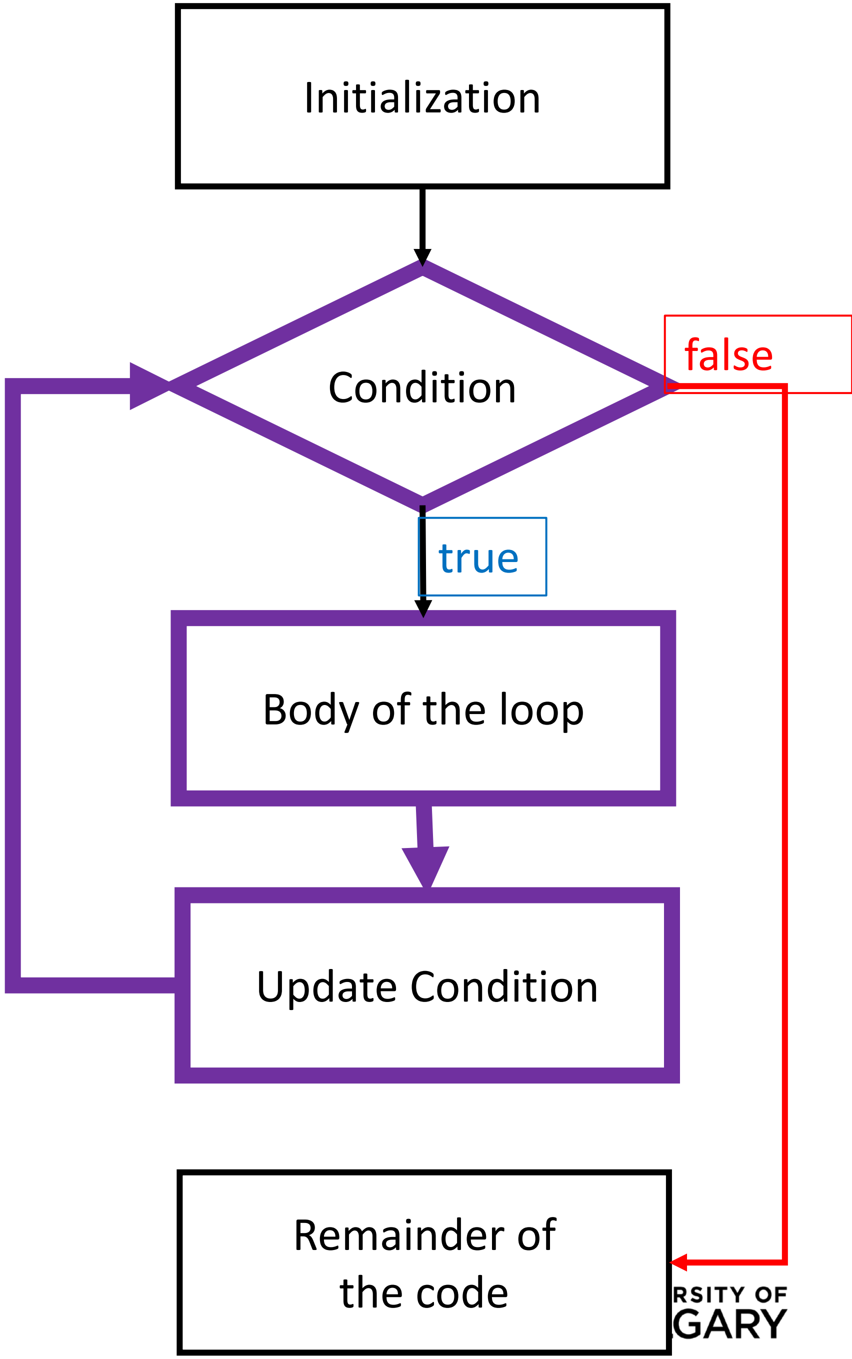


for loop (simple)

```
for (int x = 1; x <= 5; x = x +1){
        System.out.println(“Hi”);
}
```



```
Hi
Hi
Hi
Hi
Hi
```



Compare

For vs While

all for loops are while loops
all while loops are for loops*

```
int x = 1;
while (x <= 5){
        System.out.println(“Hi”);
        x = x + 1;
}

for (int x = 1; x <= 5; x = x +1){
        System.out.println(“Hi”);
}
```

*For-each loops are different

all for loops are while loops
all while loops are for loops*

```
int x = 1; int y = 2;
while (x <= 5){
        System.out.println(“Hi”);
        x = x + 1; y = y + 10;
}

for (int x = 1, int y = 2; x <= 5; x = x + 1, y = y + 10){
        System.out.println(“Hi”);
}
```

*For-each loops are different

Loops in Java – Developing for/while

The following are equivalent loops:

```
int sum = 0;

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

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

Loops in Java – Developing for/while

The following are equivalent loops:

```
int sum = 0;

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

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

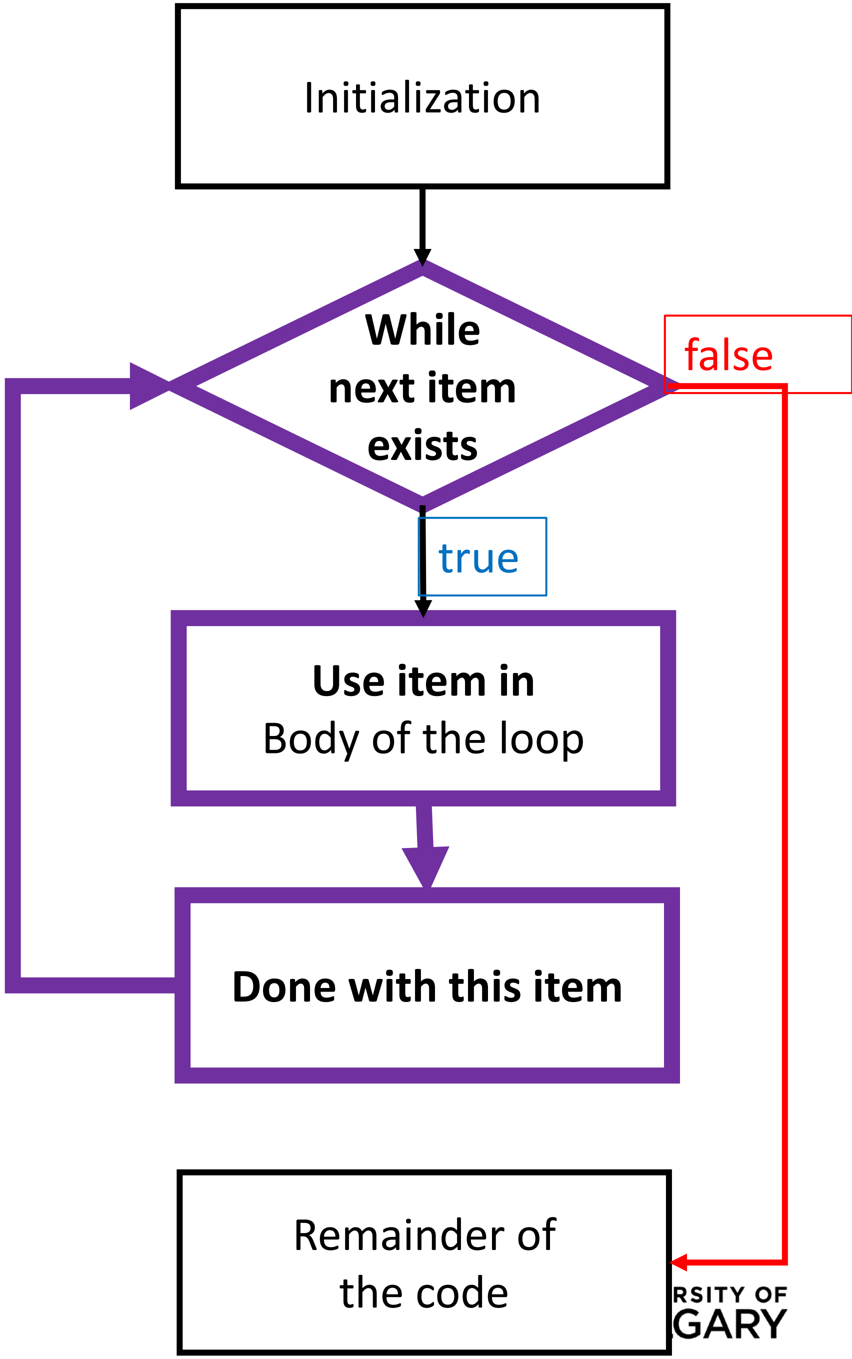
For-each Loop

for-each loops - invalid

```
String hello = "Hello, World!";
for (char x: hello){
    System.out.println(x);
}
```

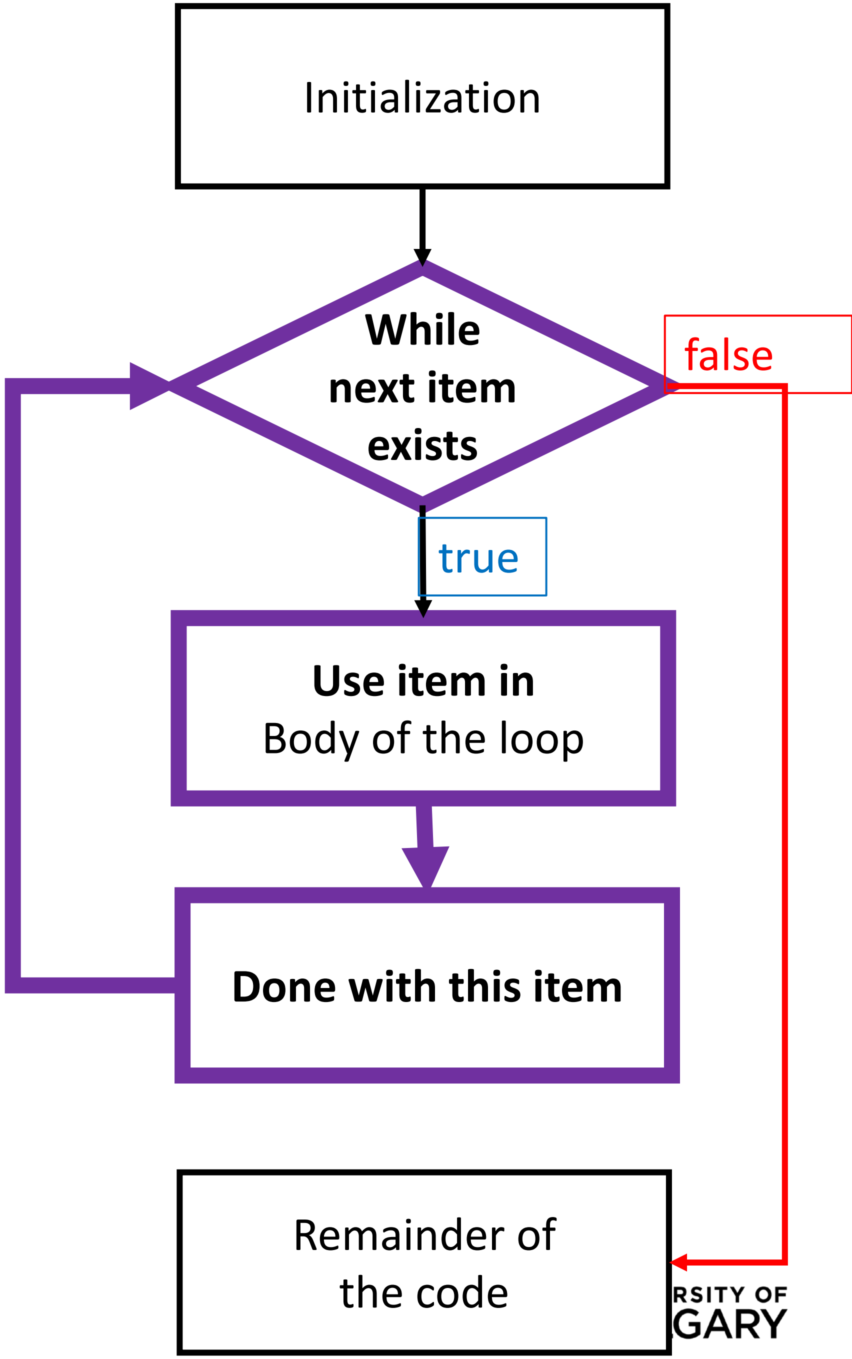
Not always as simple as Python (this is invalid syntax!)

- For-each on String requires more syntax
- For-each on other types will exist (like ArrayLists)



for-each loops - valid

```
String hello = "Hello, World!";
//Change String into for-each loop-able type
for (char x: hello.toCharArray()){
    System.out.println(x);
}
// By index
for (int i =0; i < hello.length(); i++){
    System.out.println(hello.charAt(i));
}
```

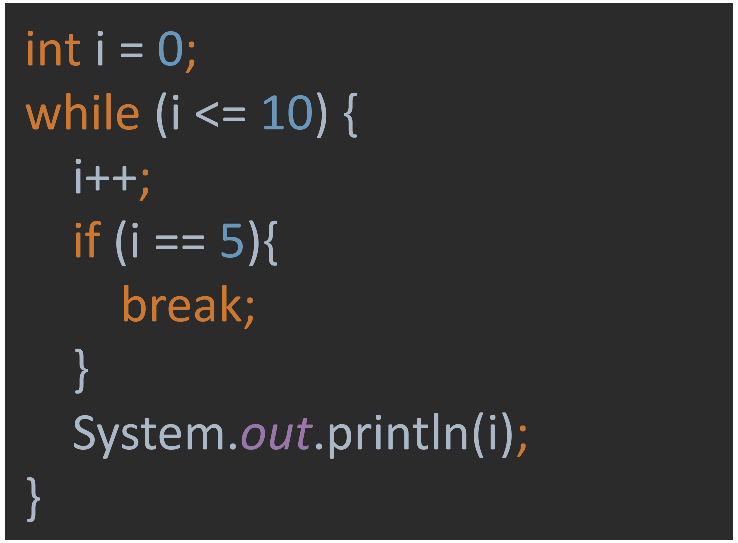


Break/Continue

Break and Continue

- Allow a loop iteration to end prematurely
- `break`
  - Entire loop ends immediately
  - Execution continues at the first statement after the loop body
- `continue`
  - Current iteration ends immediately
  - Execution returns to the top of the loop
    - In a for loop, the next item in the list is used

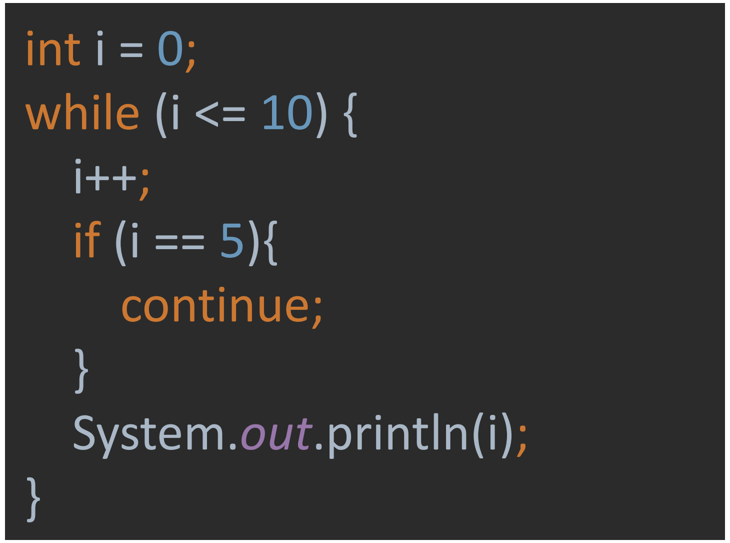
Break and Continue



```
int i = 0;
while (i <= 10) {
   i++;
   if (i == 5){
      break;
   }
   System.out.println(i);
}
```

1, 2, 3, 4

ends loop at 5 before print



```
int i = 0;
while (i <= 10) {
   i++;
   if (i == 5){
      continue;
   }
   System.out.println(i);
}
```

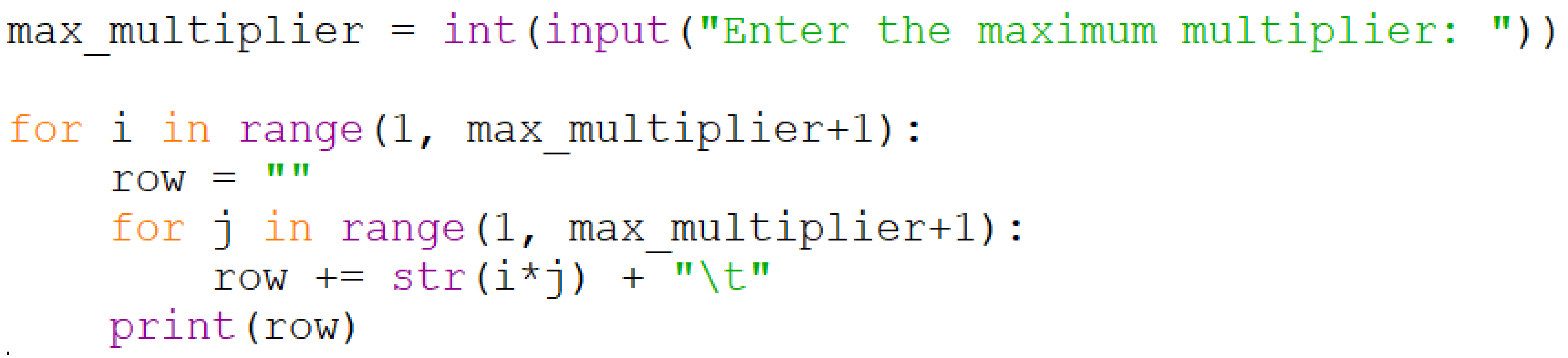
1, 2, 3, 4, 6, 7, 8, 9, 10, 11

skips rest of body before print at 5

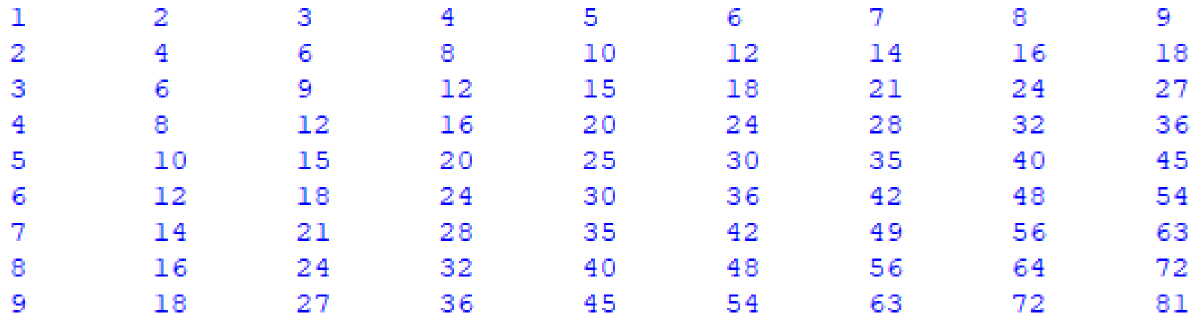
Loop Examples

Multiplication Table (Python)

Produce a multiplication table from **1** to some **value** inputted by user:



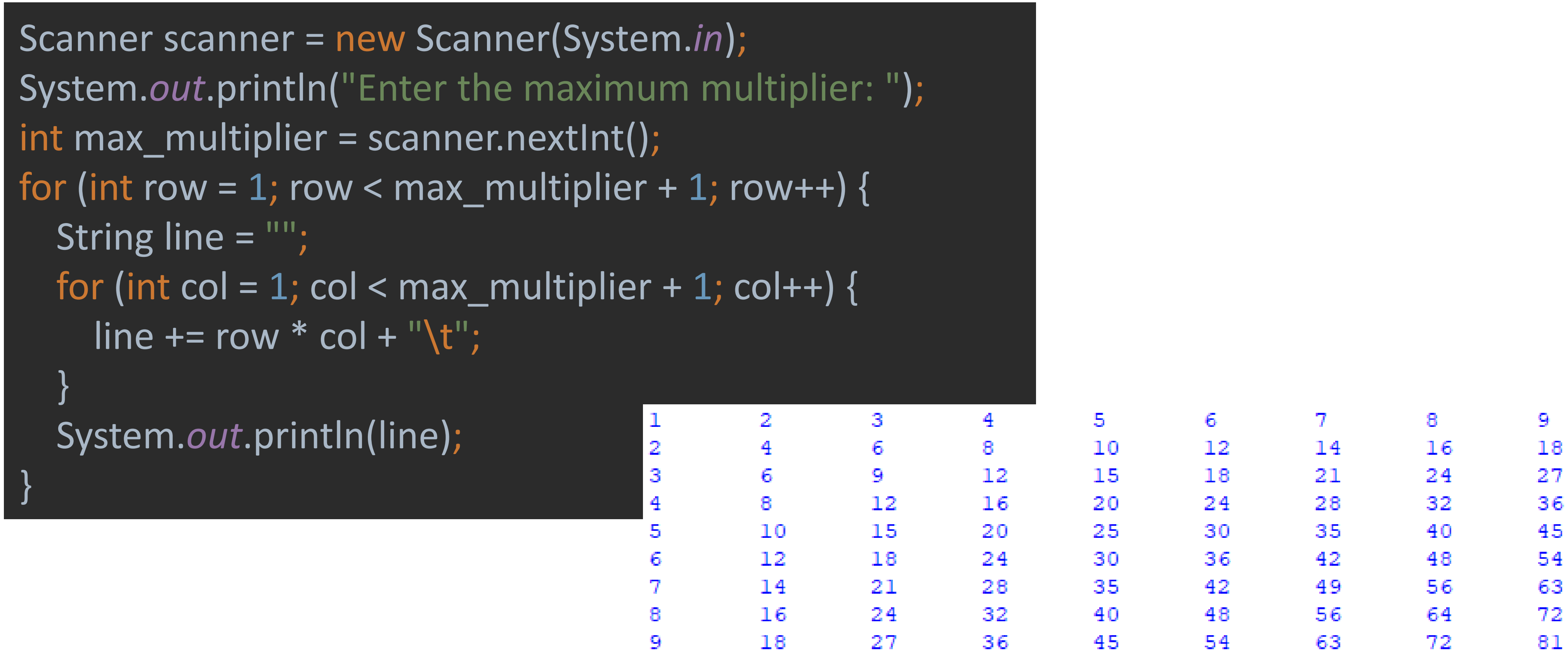
```
max_multiplier = int(input("Enter the maximum multiplier: "))

for i in range(1, max_multiplier+1):
    row = ""
    for j in range(1, max_multiplier+1):
        row += str(i*j) + "\t"
    print(row)
```



```
1       2       3       4       5       6       7       8       9
2       4       6       8       10      12      14      16      18
3       6       9       12      15      18      21      24      27
4       8       12      16      20      24      28      32      36
5       10      15      20      25      30      35      40      45
6       12      18      24      30      36      42      48      54
7       14      21      28      35      42      49      56      63
8       16      24      32      40      48      56      64      72
9       18      27      36      45      54      63      72      81
```

Multiplication Table (Java)

Produce a multiplication table from **1** to some **value** inputted by user:



```
Scanner scanner = new Scanner(System.in);
System.out.println("Enter the maximum multiplier: ");
int max_multiplier = scanner.nextInt();
for (int row = 1; row < max_multiplier + 1; row++) {
    String line = "";
    for (int col = 1; col < max_multiplier + 1; col++) {
        line += row * col + "\t";
    }
    System.out.println(line);
}
```

```
1       2       3       4       5       6       7       8       9
2       4       6       8       10      12      14      16      18
3       6       9       12      15      18      21      24      27
4       8       12      16      20      24      28      32      36
5       10      15      20      25      30      35      40      45
6       12      18      24      30      36      42      48      54
7       14      21      28      35      42      49      56      63
8       16      24      32      40      48      56      64      72
9       18      27      36      45      54      63      72      81
```

Christmas tree

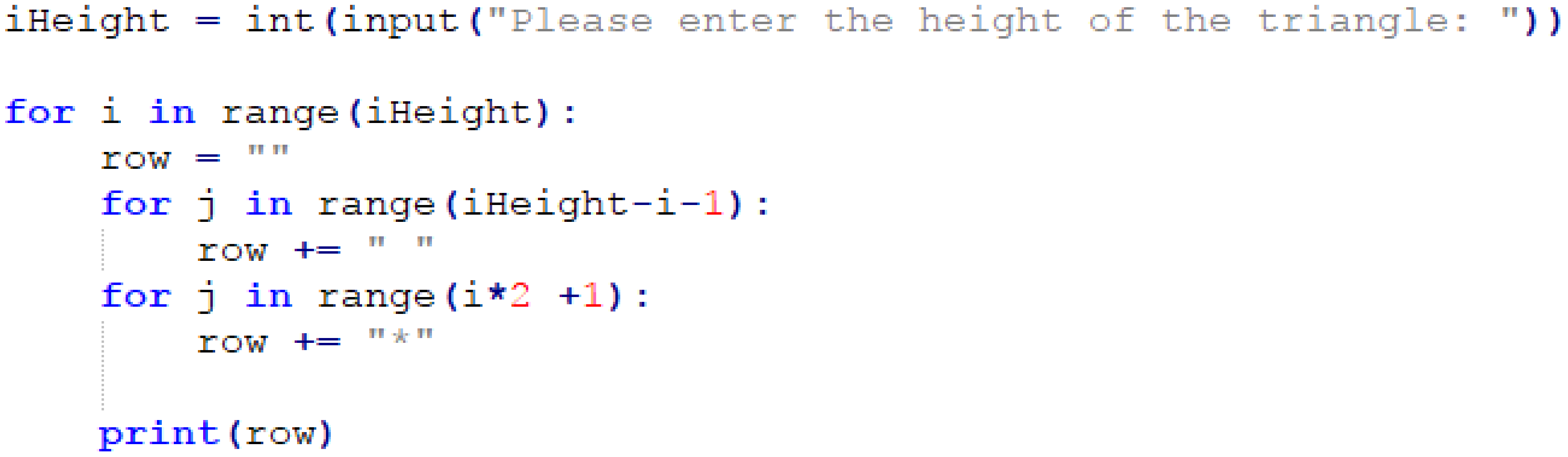
- Write a program that will print a triangle of a height provided by the user. For example:
  - If the height is 3, the triangle will look like:

```
  *
 ***
*****
```

  - If the height is 10, the triangle will look like:

```
         *
        ***
       *****
      *******
     *********
    ***********
   *************
  ***************
 *****************
*******************
```

Christmas Tree Solution (Python)



```
iHeight = int(input("Please enter the height of the triangle: "))

for i in range(iHeight):
    row = ""
    for j in range(iHeight-i-1):
        row += " "
    for j in range(i*2 +1):
        row += "*"

    print(row)
```

Christmas Tree Solution (Java)

```java
Scanner scanner = new Scanner(System.in);
System.out.println("Enter height of tree: ");
int height = scanner.nextInt();
for (int h = 0; h < height; h++) {
    int spaces = height - h - 1;
    int stars = 2 * h + 1;
    String line = "";
    for (int i = 0; i < spaces; i++) {
        line += " ";
    }
    for (int i = 0; i < stars; i++) {
        line += "*";
    }
    System.out.println(line);
}
```

Onward to … Functions.

Jonathan Hudson
jwhudson@ucalgary.ca
https://pages.cpsc.ucalgary.ca/~jwhudson/

UNIVERSITY OF CALGARY