3 Types of Loops:
(1) do-while} use when number of iterations is not known in advance.
(2) while:

```
{ skip
  { do ? <loop body> while (condition)?
    { while (condition): }
```

"Difference?" Do-while executes at least once.

ex:
```
int count = 10; // not 0!
do {
  System.out.println(count);
  count++;
} while (count < 5); // not 0!
```

"Output?" => 10 done
```
System.out.println("done");
```

do-while: executes at least once
while: may not execute.
(3) for loop: use when number of iterations is known in advance.

ex: input 10 integers from user

Scanner scan = new Scanner(System.in);
int num;

for(int i = 0; i < 10; i++) {
    Initialization (done once)  condition (repeated)  increment (repeated)
    (step 1)  12, 5, 8, ...  1  (4, 7, 10, ...)

    System.out.println("Enter number: ");  // body
    num = scan.nextInt();

    sum += num;  // must initialize to 0

}  // end of for loop

In-class: Modify above to print sum of 10 integers.

int sum = 0;  // must initialize to 0

for (i = 0; i < 10; i++) {
    sum += scan.nextInt();
}

System.out.println(sum);
Nested Loops:

1. for (int i = 0, i < 10, i++) {
   for (int j = 20; j >= 0, j--) {
      System.out.println("hi");
   }
}

   "How many times is "hi" printed? 10 * 21 = 210"

2. int i = 0;
   while (i < 10) {
   for (int j = i; j < i + 2; j++)
      System.out.println(i + " " + j);
      i++;
   } //not in for body

Final i, j values?

Make a table!
<table>
<thead>
<tr>
<th>Run</th>
<th>i+2</th>
<th>j</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0 0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>9 10</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>11</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**final values!**

*In-class code <Infinite loop>*

(Modify above)

```java
i = 0
while (i < 10) {
    for (int j = i; j < i + 2; j++)
        j++;
```

"
Arrays

array - list of values of the same type

ex: array of ints:

\[
\begin{bmatrix}
10 & 24 & 19 & 31 & 43 & 51 & 12
\end{bmatrix}
\]

Some terminology:

- length = 7
- indices: \( \rightarrow 0, 1, 2, 3, 4, 5, 6 \)

indices go from 0 to length - 1

- element: 24, 19, etc.

Creating an array:

ex: array of size 10 of ints

\[
\text{int }[ \text{ heights } ] = \text{ new int }[ 10 ];
\]

can access length via heights.length \( \Rightarrow 10 \)

This creates an empty array of size 10

\[
\begin{bmatrix}
0 & 1 & 2 & \ldots & 9
\end{bmatrix}
\]
To fill an array:
(* Fill one index *)
heights[2] = 72;

(* Fill with variable for index *)
int x = 5;
heights[x] = 61;

To access elements:

ex: int sum = heights[0] + heights[1]

Q: How to print the element in the middle?

System: println("middle element is: " + heights[heights.length/2]);

Typically, we fill an array with a loop.

Q: What kind of loop? Usually a for loop since we know number of iterations.

exs: int[] nums = new int[10]

(1) Fill nums with numbers from 1-10:
for (int i = 0; i < nums.length; i++)
    nums[i] = i+1;
2) Print the values:

```java
for (int i = 0; i < nums.length; i++)
    System.out.println(nums[i]);
```

3) Program to input 10 ints from user and print in reverse

```java
Scanner scan = new Scanner(System.in);

int arr = new int[10];
//get input
for (int i = 0; i < 10; i++)
    System.out.println("Enter number: ");
arr[i] = scan.nextInt();

//print in reverse
for (int i = length-1; i >= 0; i--)
    System.out.println(arr[i]);
```

Notice: ok to have "int i" in both for loops since they each have scope local to their for loop.
(2) How to modify previous program to get number of inputs from user?

// Before for loop:
System.out.println("How many inputs?");
int num = Scanner.nextLong();
int[] arr = new int[num];

(3) Get max of numbers in arr?

int max = arr[0];
for (int i = 1; i < arr.length; i++)
if (arr[i] > max)
max = arr[i]

Initialize List: Set initial elements of array.

int[] scores = {87, 98, 65, 43, 100};
char[] vowels = {'a', 'e', 'i', 'o', 'u'};

Arrays, like Strings, are objects and can cause exceptions—error due to invalid use of an object.

<Sample code> (Exceptions)

for (int i = 1; i <= arr.length; i++)
System.out.println(arr[i]);