Conditionals (if/else):

1) Input an integer from user and output valid if positive.

- comparative operators: \(<\>\leq\geq\=\neq\leftarrow\)
- logical operators: \(!\) \(\land\) \(\lor\) \(\Rightarrow\) \(\leftarrow\) \(\rightarrow\)

```java
if (condition) {} // if block
<body>

else {} // else block
<body>
```

Boolean valid:
```java
Scanner scan = new Scanner(System.in); // needs import
int num = scan.nextInt();
```

if (num > 0) \(\Rightarrow\) Needed if more than 1 statement.
```java
valid = true;
System.out.println("valid");
```

else // else-block
```java
valid = false
```

2) Output valid if num < 20 or > 50

2 ways
if (num < 20) // if block
    valid = true
    System.out.println("valid");

else if (num > 50) // else-if block
    valid = true;
    System.out.println("valid");

else
    valid = false;

--- 2nd way ---

if (num < 20 || num > 50)
    valid = true
else
    valid = false.

Blocks of code limit accessibility of variables.

Scope - area within a program where a variable can be accessed. Rule - variable can be accessed only within innermost block of its declaration (boolean valid).
Ex. if \((\text{num} < 20)\) \(\not\Rightarrow\) necessary

```
boolean valid2 = true;
```

System.\textbf{println} (valid2);

Will it compile? No.

\textbf{valid2} declared here

\textbf{println() } not within valid2's block

\textbf{Sample Code}

\textbf{Loops:}

Looping done more easily with increments:

\begin{align*}
i++ & : i = i + 1 \\
i-- & : i = i - 1 \\
i+2 & : i = i + 2 \\
i-5 & : i = i - 5 \\
i* & : i = i \times k
\end{align*}
3 Types of Loops.

1) do while use when number of iterations
2) while is not known in advance

do: while:

do while (condition):
  \textbf{<loop body>}

while (condition):
  \textbf{<loop body>}

"Difference?" Do - loop executes at least once"

ex:

int count = 10
while (count < 5):
  System.out.println(count)
  count++

while (count < 5):
  System.out.println("done")

"Output?" => 10 done
(3) for loop: use when number of iterations is known in advance.

Ex: input 10 integers from user

```java
Scanner scan = new Scanner(System.in);
int num,
for (int i = 0; i < 10; i++) {
    System.out.print("Enter int:");
    num = scan.nextInt();
}
```

Break this up:

- **Steps 1**: initialization
- **Steps 2, 5, 8**: condition
- **Steps 4, 7, 10**: increment
- (done once)
- (repeated)

```java
System ...
num = ...
```

```
System.out.println(sum);
```

Question: how to modify above to print sum of integers?

```java
Scanner ...
int sum = 0; // must initialize to 0!
for (...
    System ...
    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");

2. while (i < 10)
   for (int j = i; j < i+2; j++)
       System.out.println(i + " x " + j);

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

Final i, j values?

<table>
<thead>
<tr>
<th>i</th>
<th>j</th>
<th>i+2</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
<Sample Code> 9 9

Sample Code