The Command Tree drawing algorithm (i.e. method) will now proceed a little faster and several steps will be shown on each page.

Usually there will be three:

(1) STUB
(2) UNDERLINE
(3) DRAW
This also introduces another kind of if conjunction known as the if-else conjunction. It has THREE legs instead of two. Its tree branch looks like the example below:
y = 5;
if (z==4) x = 2;
if (w>x) y=y+1;
else w=7;
else x=9;

NOTE: It may later turn out that we should have used an if-else node instead of an if node. NEVER make this decision UNTIL you ABSOLUTELY have to.
y = 5;
if (z==4) x = 2;
if (w>x) y=y+1;
else w=7;
else x=9;

if (x<3) {
    y = 5;
    if (z==4) x = 2;
    if (w>x) y=y+1;
    else w=7;
} else x=9;
\begin{verbatim}
if ( x<3 ){
    y = 5;
    if (z==4)  x = 2;
    if (w>x)  y=y+1;
    else w=7;
} else x=9;
\end{verbatim}
if (x<3) {
    y = 5;
    if (z==4)  x = 2;
    if (w>x)  y=y+1;
    else w=7;
} else x=9;

```
if
```

```
Seq
```

```
x<3
```

```
y=5;
```

```
(1) STUB
```

```
(2) UNDERLINE
```

```
(3) DRAW
```

Fourth S-U-D cycle
Monday, February 06, 2012
1:15 PM
\[ y = 5; \]
\[
\text{if (} z == 4 \text{)} x = 2; \\
\text{if (} w > x \text{)} y = y + 1; \\
\text{else } w = 7; \\
\text{else } x = 9; \\
\]

(2) UNDERLINE

NOTE: You ONLY underline the if CONJUNCTION
The parenthesized expression is done in the next cycle.
if (x<3) {
    y = 5;
    if (z==4) x = 2;
    if (w>x) y = y+1;
    else w = 7;
} else x = 9;

Again - our choice of an if node instead of if-else may turn out incorrect -
BUT WAIT AND SEE.
```java
if ( x<3 ) {
    y = 5;
    if (z==4)  x = 2;
    if (w>x)  y=y+1;
    else w=7;
} else x=9;
```

**NOTE**: that the `x=2;` leaf **REQUIRES** a semicolon ( ; ) but that the `z==4` leaf **MUST NOT** have one.
if ( x<3 ) {
    y = 5;
    if (z==4) x = 2;
    if (w>x) y=y+1;
    else w=7;
} else x=9;

(3) To see which - LOOK at the very next thing that would be underlined.

(4) ONLY an else will justify a third leg. In this case there IS NO else and so there IS NO third leg.

(1) There is a temptation to add a THIRD leg to the lower if node.

(2) Sometimes this is okay and sometimes it is BAD.

(5) BAD BAD BAD - do NOT add a third leg
if (x<3) {
    y = 5;
    if (z==4) x = 2;
    if (w>x) y = y+1;
    else w = 7;
} else x = 9;

Since there was NO else next in the java, DO NOT:
(1) underline the next conjunction (the if)
(2) nor draw a STUB
BUT INSTEAD - just check off the if and move back up the tree to the Seq node above it.
if (x<3) {
    y = 5;
    if (z==4) x = 2;
    if (w>x) y = y+1;
    else w = 7;
} else x = 9;

(2) UNDERLINE
if (x<3) {
  y = 5;
  if (z==4) x = 2;
  if (w>x) y = y+1;
  else w = 7;
} else x = 9;

(2) UNDERLINE

if (x<3) {  
  y = 5;
  if (z==4) x = 2;
  if (w>x) y = y+1;
  else w = 7;
} else x = 9;

(1) STUB

(3) DRAW
if (x<3) {
    y = 5;
    if (z==4) x = 2;
    if (w>x) y = y+1;
    else w = 7;
} else x = 9;

if (x<3) {  
    y = 5;
    if (z==4) x = 2;
    if (w>x) y = y+1;
    else w = 7;
} else x = 9;

if (w>x) y = y+1;

(1) STUB

(2) UNDERLINE

(3) DRAW
if (x<3) {
    y = 5;
    if (z==4) x = 2;
    if (w>x) y = y+1;
} else w = 7;
} else x = 9;

THIS TIME the next underlined item IS an else and so you get to underline it.
And also
CHANGE the if node to be an if-else

And so in the next cycle, there WILL BE a third leg.
if (x<3) {
    y = 5;
    if (z==4) x = 2;
    if (w>x) y = y+1;
    else w = 7;
} else x = 9;
if (x<3) {
    y = 5;
    if (z==4) x = 2;
    if (w>x) y=y+1;
    else w=7;
} else x=9;

NOTE: An if-else node can ONLY have exactly THREE legs - so NO MORE legs can be added to this if-else.
if (x<3) {
    y = 5;
    if (z==4) x = 2;
    if (w>x) y=y+1;
    else w=7;
} else x=9;

The underlined close brace causes the Seq to be complete (no more legs).
if (x<3) {
    y = 5;
    if (z==4) x = 2;
    if (w>x) y=y+1;
    else w=7;
} else x=9;

Underlines else changes the if node into an if-else node.

Which will have a third leg in the NEXT S-U-D cycle.
```c
if ( x<3 ) {
    y = 5;
    if (z==4) x = 2;
    if (w>x) y=y+1;
    else w=7;
} else x=9;
```
if (x<3) {
    y = 5;
    if (z==4) x = 2;
    if (w>x) y=y+1;
    else w=7;
} else x=9;

Third leg finishes the if-else node - so check it off.

Now the entire Java Command has been underlined AND ALL of the tree nodes have been checked off - all is WELL and all is FINISHED.
Number the circled nodes starting with #1 at the top. Each child node MUST have a number LARGER than its parent. DO NOT SKIP any numbers.
Note that the boolean expressions:

```
x<3
z==4
w>x
```

Are NOT commands and DO NOT have a semicolon.

The # indicates the child is a numbered internal node.

Don't leave out the semicolons.

<table>
<thead>
<tr>
<th>Mentor Table</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>if-else</td>
</tr>
<tr>
<td>#2</td>
<td>seq</td>
</tr>
<tr>
<td>#3</td>
<td>if</td>
</tr>
<tr>
<td>#4</td>
<td>if-else</td>
</tr>
</tbody>
</table>

The Mentor Table shows the relationships between the if-else, seq, and if nodes in the diagram, with each node's condition and actions. The diagram visually represents the execution flow of the boolean expressions and the corresponding actions.