Basic Control Structures

- A **sequence** is a series of statements that execute one after another.
- A **selection** (branch) is used to determine which of two or more different statements to execute depending on a certain condition.
- A **loop** (repetition) is used to repeat statements while certain conditions are true.
- A **subprogram** is a smaller part of another program solving a certain problem; a collection of subprograms solves the original problem.
Sequence

♦ A series of statements that execute one after another
Sequence

- A series of statements that execute one after another
Selection

- Determine which of two or more different statements to execute depending on a certain condition

Condition → If Condition is True, execute Statement 1; if False, execute Statement 2

Only Statement 1 or Statement 2 is executed, but not both

IF (Condition) THEN (Statement 1) ELSE (Statement 2)
Selection

✦ Determine which of two or more different statements to execute depending on a certain condition

Figure 1-5: An example of the selection structure

1. repeat 20 times:
   walk
2. if the balloon is red, do this:
   drop the balloon in the red box
   otherwise, do this:
   drop the balloon in the yellow box
3. turn
4. repeat 20 times:
   walk
5. turn
Loop

- Repeat statements while certain conditions are true

Condition

True

False

Statement 1

Statement 1 is executed only if while Condition is True

WHILE (Condition) DO (Statement 1)
Loop

- Repeat statements while certain conditions are true

Figure 1-3: Another example of the repetition structure

algorithm

1. repeat until you are directly in front of the chair:
   walk
2. turn
3. sit
Subprogram

- A smaller part of another program solving a certain problem

A meaningful collection of “Sequence”, “Selection”, “Loop”, and “Subprogram”

A collection of subprograms solves the original problem
Evaluating Expressions

✦ How expressions are evaluated?

\[
\text{mybox} = 5
\]

✦ How mathematical equations are evaluated?

\[
\text{mybox} = a \times \frac{b + c}{e}
\]

If we want to change the sequence of calculating the equation we have to use brackets ( )

\[
\text{mybox} = a \times \frac{(b + c)}{e}
\]

\[
\text{mybox} = a \times (\frac{b + c}{e})
\]
Evaluating Expressions

✦ How do we use the following operators:

\[
\begin{align*}
&\lt &\leq &\gt &\geq &== &!= \\
&\text{mybox < 5} &\text{mybox }\geq 5 &\text{mybox }== 5 &\text{mybox }!= 5
\end{align*}
\]

✦ What about the following logical operators too...

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
</tr>
</tbody>
</table>
Evaluating Expressions

What about the following logical operators too...

&& and

|| or

! not

mybox < 5 && num == 10

True

mybox >= 5 && num != 0

True

num = 10
mybox = 5
midterm = 28
finalexam = 96

midterm == 0 || finalexam < 50

False

midterm > 27 || finalexam > 95

True

False
What do you think the following code do?

boolean status;
status = mybox < 5;  // False

boolean status;
status = num != 0;   // True

status = ( midterm == 0 || finalexam > 50 )

num = 10
mybox = 5
midterm = 28
finalexam = 96