Chapter 4: Control structures

Introduction

Objectives

• What are control structures
• Relational Operators
• Logical operators
• Boolean expressions
• Conditional (Decision) statements
• Loop statements
What are Control Structures

- Without control structures, a computer would evaluate the instructions in a program step-by-step.

- Control structures allow:
  - Defining which instructions are evaluated
  - Changing the order in which instructions are evaluated
  - Controlling the “flow” of the program

- Control structures include:
  - Block statements (anything contained within curly brackets)
  - Decision statements
  - Loops

Relational Operators

Relational operators produce boolean values.

- `==` Equal to
- `!=` Not equal to
- `<` Less than
- `<=` Less than or equal to
- `>` Greater than
- `>=` Greater than or equal to
Use of relational Operators

left_operand relational_operator right_operand

- counter < 5
- counter <= maximum

- Relational operators can be combined with arithmetic operators:
  5 + 3 < 4 → false because 8 is not < 4
  myNumber % 2 == 1 → false if myNumber is odd
  → true otherwise

- Relational operators are always performed last!!

Logical Operators

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Operator Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;&amp;</td>
<td>AND</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>!</td>
<td>NOT</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>&amp;&amp;</th>
<th>T</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

| || | T | F |
|----|---|---|
| T  | T | T |
| F  | T | F |

<table>
<thead>
<tr>
<th>!</th>
<th>T</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>
Boolean Expressions

• **Boolean expression** is an expression that is evaluated to a boolean value.
• **Atomic Boolean expression** uses one and only one of the relational operators.
  - `myBalance <= yourBalance`
• **Complex Boolean expressions** may be defined by *linking* other Boolean expressions using logical operators.
  - `(myBalance <= yourBalance) && (yourAge > 20)`
  - `!(myBalance <= yourBalance) && (yourAge <= 20)`
• **Boolean expressions** may be assigned to boolean variables.
  - `boolean isHeOlder = (myAge < hisAge);`