Programming Example 5: Conditional Statements

Introduction

This example illustrates the use of an activity diagram to show conditional statements in a program, and demonstrates a number of important ideas student's should keep in mind when writing programs that use conditional expressions.

As you read through this example, look for code that illustrates these important principles:

1. **Conditions:** When you see words such as *when* or *if* in a problem description, it is likely that the problem will require one or more conditional statements. Programs can be written to execute differently, based on certain conditions being true or false.
2. **Relational Operators:** Conditions are described using **relational** operators, for example, `(value < maxValue)`. The relational operators are

   ```
   == (equals)
   != (not equals)
   < (less than)
   <= (less than or equals)
   > (greater than), and
   >= (greater than or equal).
   ```

3. **if statement:** The *if* statement allows a single statement or block of statements to be executed when a condition is true. Otherwise the statement or block of statements is skipped.
4. **if/else statement:** The *if/else* statement executes one block of code when a condition is true, and a different block of code when the condition is false.
5. **Nesting conditional statements:** Conditional statements can be nested inside of one another.
6. **Logical Operators:** Conditions can be combined with the logical operators || and &&.

The problem statement for this program is located here.

The activity diagram for this program is located here.

The example program is located here. An executable of this program can be found here.