Example: GPA problem

Statement of GPA problem:
Write a program that reads the credits and quality points earned and outputs the gpa.

Algorithm:
variables: qp, credits, gpa
1. Input qp
2. Input credits
3. gpa = qp / credits
4. Print gpa

Solution 6

Algorithm:
variables: qp, credits, gpa
1. Input qp
2. Input credits
3. if credits = 0
   a) Print “No gpa yet”
   else
   a) gpa = qp / credits
   b) Print gpa
4. Print goodbye message

if (credits == 0)
   System.out.println ("\n\tGPA: None");
else
{
    gpa = (double) qp / credits;
    System.out.println ("\n\tGPA: " + gpa);
}
5.1 Boolean Expressions

- Java's relational operators return boolean results:

  ```
  ==   equal to  
  !=   not equal to  
  <    less than    
  >    greater than  
  <=   less than or equal to  
  =>   greater than or equal to  
  ```

- Java's boolean operators on boolean expressions:

  ```
  !    not  
  &&   and  
  ||   or  
  ```

Truth tables for boolean operators

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>!a</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
</tr>
</tbody>
</table>
Boolean Expressions

- Specific expressions can be evaluated using truth tables – let’s try this one:

<table>
<thead>
<tr>
<th>total &lt; MAX</th>
<th>found</th>
<th>!found</th>
<th>total &lt; MAX &amp;&amp; !found</th>
</tr>
</thead>
<tbody>
<tr>
<td>false</td>
<td>false</td>
<td>true</td>
<td></td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>true</td>
<td>false</td>
<td>true</td>
<td></td>
</tr>
<tr>
<td>true</td>
<td>true</td>
<td>false</td>
<td></td>
</tr>
</tbody>
</table>

Quick Check

What do the following statements do?

```java
if (total != stock + warehouse)
    inventoryError = true;
```

```java
if (found || !done)
    System.out.println("Ok");
```
Write an algorithm that reads two numbers and prints a message stating whether either one is divisible by the other.

```java
//********************************************************************
// MinOfThree.java Author: Lewis/Loftus
// Demonstrates the use of nested if statements.
//********************************************************************
import java.util.Scanner;
public class MinOfThree
{
    public static void main(String[] args)
    {
        int num1, num2, num3, min = 0;
        Scanner scan = new Scanner(System.in);
        System.out.println("Enter three integers: ");
        num1 = scan.nextInt();
        num2 = scan.nextInt();
        num3 = scan.nextInt();
        if (num1 < num2)
            if (num1 < num3)
                min = num1;
            else
                min = num3;
        else
            if (num2 < num3)
                min = num2;
            else
                min = num3;
        System.out.println("Minimum value: " + min);
    }
}
```

Sample Run

Enter three integers: 
84 69 90
Minimum value: 69
Another algorithm for computing the MIN of three numbers

Another algorithm for computing the MIN of three numbers?

Homework: Read Sections 5.1, 5.2
Do Exercises EX 5.1 – 5.5