Reading:
  1. Dale, Chapter 13 (Section 1)

Objectives:
  1. Introduce recursion
  2. Discuss any project issues or problems

Concepts:
  1. Recursion
1. Introduce recursion
   - What is recursion?
     - Recursive methods (a method that calls itself either directly or indirectly through another method)
   - Recursive problem solving approaches’ elements
     - A recursive method is called to solve a problem
     - The method actually knows how to solve only the simplest case(s)
     - If the method is called with a base case, the method simply returns a result.
     - If the method is called with a more complex problem, the method divides the problem into two conceptual pieces: a piece that the method knows how to do and a piece that the method doesn’t know how to do
   - Give recursion examples
   - Explain why is recursion used, and show how in some cases it is the only possible solution

2. Dedicate last 30 minutes of class to any project questions that may have arisen as the students have been working. Suggest some possible implementation solutions if asked.
Objective:
1. Practice recursion

Student Activities:
1. Write a program that uses recursion to calculate and print the factorials of the integers from 0 to 10. The way factorial works is the following:

\[
\begin{align*}
5! &= 5 \times 4! \\
4! &= 4 \times 3! \\
3! &= 3 \times 2! \\
2! &= 2 \times 1! \\
1! &= 1 \\
\end{align*}
\]

The output of your program should look like:

<table>
<thead>
<tr>
<th>n</th>
<th>n!</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>24</td>
</tr>
<tr>
<td>5</td>
<td>120</td>
</tr>
<tr>
<td>6</td>
<td>720</td>
</tr>
<tr>
<td>7</td>
<td>5040</td>
</tr>
<tr>
<td>8</td>
<td>40320</td>
</tr>
<tr>
<td>9</td>
<td>362880</td>
</tr>
<tr>
<td>10</td>
<td>3628800</td>
</tr>
</tbody>
</table>
// Factorial Test: Recursive factorial method
import java.awt.Graphics;
import java.applet.Applet;

public class FactorialTest extends Applet
{
    public void paint (Graphics g)
    {
        int yPosition = 25;
        for ( int j = 0; j <= 10; j++)
        {
            g.drawString(j + "! = " + factorial(j), 25, yPosition);
            yPosition +=15;
        }
    }

    //Recursive definition of method factorial
    public long factorial (int number)
    {
        if (number <= 1) // base case
            return 1;
        else
            return number * factorial(number – 1);
    }
}