Chapter 16 Lab
Recursion

Lab Objectives

- Be able to trace recursive function calls
- Be able to write non-recursive and recursive methods to find geometric and harmonic progressions.

Introduction

In this lab we will follow how the computer executes recursive methods, and will write our own recursive method, as well as the iterative equivalent. There are two common progressions in mathematics, the geometric progression and the harmonic progression. The geometric progression is defined as the product of the first ***n*** integers. The harmonic progression is defined as the product of the inverses of the first ***n*** integers. Mathematically, the definitions are as follows:

Geometric (n) = $\prod_{i=1}^{n} i = n * \prod_{i=1}^{n-1} i$

Harmonic (n) = $\prod_{i=1}^{n} \frac{1}{i} = \frac{1}{n} * \prod_{i=1}^{n-1} \frac{1}{i}$

Let's look at examples.
If we use ***n*** = 4, the geometric progression would be 1 * 2 * 3 * 4 = 24, and the harmonic progression would be $1 * \frac{1}{2} * \frac{1}{3} * \frac{1}{4} = \frac{1}{24}$ = 0.04166.

Task #1 Tracing Recursive Methods

1. Copy the file *Recursion.java* (see Code Listing 16.1) from the Student CD or as directed by your instructor.

2. Run the program to confirm that the generated answer is correct. Modify the factorial method in the following ways:
   a. Add these lines above the first `if` statement:
   ```
   int temp;
   System.out.println("Method call -- " +
                      "calculating " +
                      "Factorial of: " + n);
   ```

   b. Remove this line in the recursive section at the end of the method:
      ```
      return (factorial(n - 1) * n);
      ```
   c. Add these lines in the recursive section:
      ```
      temp = factorial(n - 1);
      System.out.println("Factorial of: " +
                          (n - 1) + " is " +
                          temp);
      return (temp * n);
      ```
3. Rerun the program and note how the recursive calls are built up on the run-time stack and then the values are calculated in reverse order as the run-time stack "unwinds".

Task #2 Writing Recursive and Iterative Versions of a Method

1. Copy the file *Progression.java* (see code listing 16.2) from the Student CD or as directed by your instructor.
2. You need to write **class** (`static`) methods for an iterative and a recursive version of each of the progressions. You will create the following methods:
   a. **`geometricRecursive`**
   b. **`geometricIterative`**
   c. **`harmonicRecursive`**
   d. **`harmonicIterative`**.

   Be sure to match these methods to the method calls in the `main` method.

Code Listing 16.1 (**`Recursive.java`**)

```
/**
   This program demonstrates factorials using recursion.
*/

public class Recursion
{
   public static void main(String[] args)
   {
      int n = 7;

      // Test out the factorial
      System.out.println(n + " factorial equals ");
      System.out.println(Recursion.factorial(n));
      System.out.println();
   }
```

```
   /**
      This is the factorial method.
      @param n A number.
      @return The factorial of n.
   */

   public static int factorial(int n)
   {
      int temp;

      if (n == 0)
      {
         return 1;
      }
      else
      {
         return (factorial(n - 1) * n);
      }
   }
}
```

Code Listing 16.2 (`Progression.java`)

```
import java.util.Scanner;

/**
   This program calculates the geometric and
   harmonic progression for a number entered
   by the user.
*/

public class Progression
{
   public static void main(String[] args)
   {
      Scanner keyboard = new Scanner (System.in);

      System.out.println("This program will calculate " +
                         "the geometric and harmonic " +
                         "progression for the number " +
                         "you enter.");

      System.out.print("Enter an integer that is " +
                       "greater than or equal to 1: ");

      int input = keyboard.nextInt();
      // Match the method calls with the methods you write
      int geomAnswer = geometricRecursive(input);
      double harmAnswer = harmonicRecursive(input);

      System.out.println("Using recursion:");
      System.out.println("The geometric progression of " +
                         input + " is " + geomAnswer);

      System.out.println("The harmonic progression of " +
                         input + " is " + harmAnswer);

      // Match the method calls with the methods you write
      geomAnswer = geometricIterative(input);
      harmAnswer = harmonicIterative(input);

      System.out.println("Using iteration:");
      System.out.println("The geometric progression of " +
                         input + " is " + geomAnswer);

      System.out.println("The harmonic progression of " +
                         input + " is " + harmAnswer);
   }

   // ADD LINES FOR TASK #2 HERE
   // Write the geometricRecursive method
   // Write the geometricIterative method
   // Write the harmonicRecursive method
   // Write the harmonicIterative method
}
```