**44-542 Object Oriented Programming**

**Recursion KEY**

**Objective** This worksheet will give you practice in working with recursive methods.

1. The greatest common divisor of two integers a and b, where a > 0 and b >= 0, is the largest integer that is a divisor of both a and b. For example, the greatest common divisor of 18 and 12 is 6 and the greatest common divisor of 9 and 8 is 1. Euclid discovered an algorithm for computing greatest common divisor about 2300 years ago. It may be stated using recursion:

**gcd(a, b) = gcd(b, a modulo b) if b > 0;**

**gcd(a, 0) = a.**

Using this definition, we see that

**gcd(18, 12) = gcd(12, 6) = gcd(6, 0) = 6, and**

**gcd(9, 8) = gcd(8, 1) = gcd(1, 0) = 1**

Find each of the following, showing each step as in the examples above.

* 1. **gcd(48, 39) = gcd(39, 9) = gcd(9, 3) = gcd(3, 0) = 3**
  2. **gcd(12, 15) = gcd(15, 12) = gcd(12, 3) = gcd(3, 0) = 3**
  3. **gcd(1234, 456) = gcd(456, 322) = gcd(322, 134) = gcd(134, 54) = gcd(54, 26) = gcd(26, 2) = gcd(2, 0) = 2**

1. Consider a method f defined as follows:

if n == 0, then f returns 0;

otherwise, f returns n \* n + f(n – 1);

For example,

**f(2) = 2 \* 2 + f(1) = 4 + f(1) = 4 + (1 \* 1 + f(0))**

**= 4 + (1 + f(0)) = 4 + (1 + 0) = 4 + 1 = 5**

Find the value returned by each of the following. Show each step, as in the example above.

* 1. **f(3) =**  3 \* 3 + f(2) = 9 + f(2) = 9 + 5 = 14
  2. **f(5) =** 5 \* 5 + f(4) = 25 + f(4) = 25 + 4 \* 4 + f(3) = 25 + 16 + f(3) = 41 + f(3) = 41 + 14 = 55
  3. **f(10)** = 10 \* 10 + f(9) = 100 + f(9) = 100 + 9 \* 9 + f(8) = 100 + 81 + f(8) = 181 + f(8) = 181 + 8 \* 8 + f(7) = 181 + 64 + f(7) = 245 + 7 \* 7 + f(6) = 245 + 49 + f(6) = 294 + f(6) = 294 + 6 \* 6 + f(5) = 294 + 36 + f(5) = 330 + f(5) = 330 + 55 = 385
  4. What does method f do in general?

**f(n)** returns the sum of the squares from **1** to **n**.

1. Recall that the factorial fact(n) of a positive integer n is the product of the integers 1 through n. Thus fact(3) has the value 6 = 1 \* 2 \* 3. Also, fact(0) is defined to be 1. Factorial may be defined recursively:

**fact(n) = n \* fact(n - 1) if n > 0;**

**fact(0) = 1**

For example,

**fact(2) = 2 \* fact(1) = 2 \* (1 \* fact(0)) = 2 \* (1 \* 1) =**

**2 \* 1 = 2**

Find the value returned by each of the following. Show each step, as in the example above.

* 1. **fact(3) = 3 \* fact(2) = 3 \* 2 = 6**
  2. **fact(5) = 5 \* fact(4) = 5 \* (4 \* fact(3)) =**

**5 \* (4 \* 6) = 5 \* 24 = 120**

* 1. **fact(7) = 7 \* fact(6) = 7 \* (6 \* fact(5)) =**

**7 \* (6 \* 120) = 7 \* 720 = 5040**

1. Consider the method mystery:

**public static void mystery (long n)**

**{**

**System.out.print(n % 10);**

**if ((n / 10) != 0)**

**{**

**mystery(n / 10);**

**}**

**}**

Find the output produced by each of the following calls to the method **mystery**:

* 1. **mystery(29) = 92**
  2. **mystery(12345) = 54321**
  3. What does the method **mystery** do in general?

It reverses the order of the digits