Discussion Questions for "Recursion"

1. Why did the professor modify the Wikipedia definition of "divide and conquer" to one, instead of two or more sub-problems? (Hint: How would you code a Java recursive routine to calculate factorials?)
2. Why are Fibonacci numbers interesting? Can you find some applications that might benefit from being able to calculate a Fibonacci number?
3. What would happen if you invoke fib(-5)? Would the fib function recurse infinitely? Would it return the "correct" answer?
4. What would happen if you switched the two lines of the fib function? I.E.:

```
public static int fib(int n) {
  return fib(n-1) + fib(n-2);
  if (n<2) return n;
}
```

5. The javaTpoint tutorial on recursion states "It makes the code compact but complex to understand." Do you agree? Why is recursion hard to understand?
6. Consider the following mathematic theorem proof:
   Theorem: For any non-negative integer, x, then fib(3x) is even, but fib(3x+1) and fib(3x+2) are odd.
   Lemma: Given a+b=c

If a is:	And b is:	Then a+b=c is
Even	Even	Even
Even	Odd	Odd
Odd	Even	Odd
Odd	Odd	Even

   Proof by Induction:
   For x=0…

   fib(3x)=fib(0)=0 =even
   fib(3x+1)=fib(1)=1 =odd
   Fib(3x+2)=fib(2)=0+1=1 =odd

   Assume theorem is true for x-1, prove true for x:

fib(3x)	=fib(3x-1)	+fib(3x-2)	
	=fib(3(x-1)+2)	+fib(3(x-1)+1)	
	=odd	+odd	=even
fib(3x+1)	=fib(3x)	+fib(3x-1)	
	=even	+ odd	=odd
fib(3x+2)	=fib(3x+1)	+fib(3x)	
	=odd	+even	=odd

   Q.E.D.

What is the relationship of this inductive proof to recursive programming? Are the same basic concepts being used? How is an inductive proof different from recursive programming?