Foundations of Programming

A Java programmer is attempting to write a class `BigNo` which is intended to handle integers of arbitrary size. An integer is represented as a list of single digits arranged so that the least significant digit is at the head of the list. In outline, class `BigNo` is declared thus:

class BigNo
{
  private int dig;
  private BigNo rest;

  public BigNo(int n)
  {
    this.dig = n%10;
    if (n/10 == 0)
      this.rest = null;
    else
      this.rest = new BigNo(n/10);
  }

  private BigNo add(int c)
  .

  public BigNo add(BigNo that)
  .

  private BigNo add(BigNo that, int c)
  {
    if (this.rest == null)
      return that.add(this.dig+c);
    if (that.rest == null)
      return this.add(that.dig+c);
    int d = this.dig + that.dig + c;
    return new BigNo(d%10, this.rest.add(that.rest,d/10));
  }

}  

The final return statement refers to a constructor which is not shown. Why are two constructors needed? Provide the missing constructor. Does it have to be public?  

Why are there three `add()` methods? Explain why one is public and two are private. Provide bodies for the two `add()` methods for which only heading lines are shown. 

Provide a suitable `toString()` method. 

Suppose `jack` and `jill` are `BigNo` representations of the integers 46 and 57 respectively. Describe carefully the effect of the call `jack.add(jill)`