```
public class Node {
   private int value;
   protected Node next;

   public Node () {
      this.value = 0;
      this.next = null;
   }

   public Node (int value) {
      this.value = value;
      this.next = null;
   }

   public Node (int value, Node next) {
      this.value = value;
      this.next = next;
   }

   /* Purpose:  get the value of this Node
    * Parameters: none
    * Returns: int - this node's value */
   public int getValue() {
      return value;
   }

   /* Purpose: set the value of this Node to value
    * Parameters: int value - the new value for the node
    * Returns: void – nothing */
   public void setValue(int value) {
      this.value = value;
   }

   /* Purpose:  get the next of this Node
    * Parameters: none
    * Returns: Node - the node after this node in the list  */
   public Node getNext() {
      return next;
   }

   /* Purpose: set the next of this Node to next
    * Parameters: Node - the node to set to this node's next
    * Returns: void – nothing */
   public void setNext(Node next) {
      this.next = next;
   }
}
```

```
public class NodeTester {
  public static void main (String[] args) {
    /* 1a. create a new blank (value 0) node n1 */



    /* 1b. create a new node n2 with a value of 9 */



    /* 1c. print the values in n1 and n2 */







    /* 2a. connect n1 to n2 such that n2 is before n1 */



    /* 2b. print the values of n1 and n2 without using n1 */







    /* 3a. create a new node n3 with a value of 20 */



    /* 3b. assign our n1 variable so that it also references
           the newly created node with data value 20 */



    /* 3c. Can you still print the values in all 3 nodes? */









    /* 4a. connect n3 to the other two nodes such that n3 is
         in between the other two */



    /* 4b. print all 3 values using only the n2 variable.
         That is, you cannot use the variables n1 or n3 */



    /* 5. Create a method that accepts a node as a parameter
         and prints out the value of the given node as well
         as the values of all nodes that follow it. */

  }

}
```