```
 1 /*************************************************************
 2 * Dalton Nofs                                                *
 3 * Login ID: nofs5491                                         *
 4 * CS-102, Summer 2017                                        *
 5 * Programming Assignment 5                                   *
 6 * TreeNode class:  TreeNode object for creating trees         *
 7 *************************************************************/
 8 class TreeNode<T>
 9 {
10     T datum;
11     TreeNode<T> left;
12     TreeNode<T> right;
13 
14     /*************************************************************
15     * Method: Node()                                           *
16     * Purpose: default constructor for TreeNode obj               *
17     *                                                            *
18     * Parameters:                N/A                           *
19     * Returns: void:             N/A                           *
20     *************************************************************/
21     public TreeNode()
22     {
23         datum = null;
24         left = null;
25         right = null;
26     }
27 
28     /*************************************************************
29     * Method: getData()                                          *
30     * Purpose: get previous node or next node                   *
31     *                                                            *
32     * Parameters:                N/A                           *
33     * Returns: T:      data stored in node                        *
34     *************************************************************/
35     public T getDatum()
36     {
37         return datum;
38     }
39 
40     /*************************************************************
41     * Method: setDatum()                                           *
42     * Purpose: get previous node or next node                   *
43     *                                                            *
44     * Parameters: Object:         Data to be stored in node      *
45     * Returns: void:             N/A                           *
46     *************************************************************/
47     public void setDatum(T datum)
48     {
49         this.datum = datum;
50     }
51 
52     /*************************************************************
53     * Method: getLeft()/getRight()                               *
54     * Purpose: get left node or right node                       *
55     *                                                            *
56     * Parameters:                N/A                           *
57     * Returns: Node:             Node pointing to next or prev   *
58     *************************************************************/
59     public TreeNode<T> getLeft()
60     {
61         return left;
62     }
63     public TreeNode<T> getRight()
64     {
65         return right;
66     }
67 
```

```
68      /*************************************************************
69      * Method: setLeft()/setRight()                                *
70      * Purpose: Set the left or right node                        *
71      *                                                              *
72      * Parameters: Node:          Next/previous node              *
73      * Returns: Void:             N/A                             *
74      *************************************************************/
75      public void setLeft(TreeNode<T> left)
76      {
77          this.left = left;
78      }
79      public void setRight(TreeNode<T> right)
80      {
81          this.right = right;
82      }
83  }
```