CS 1120 – 543

Lab TA: Omofolakunmiel Olagbemi

**Tyler Thompson**

**Lab Report – LA1: Binary Trees**

**Design Phase**

**Basic Structure**

* class Main:
  + call inode class and create a new data structure
  + add childs given in the example
  + print the menu
  + loop back to menu
* class TreeDataStructure:
  + attributes - node id represented by a string
  + constructor - initializes a node based on id
  + method addChild - add a child to a parent
  + method find - find the node in the tree
  + method getParent - get the parent of a child
  + method size - return how many children a parent has
  + method toString - converts data into a string
  + method getId - get the ID of a node
  + method printTree - print the tree
* interface Inode:
  + pointers for the TreeDataStructure class

**Pseudocode**

* public TreeDataStructure(String id)
* public boolean addChild(String ID, String parentID);
  + go to TreeDataStructure constructor
  + pass the child and parent id
  + printTree()
* public Node find(String value);
  + go to the TreeDataStructure constructor
  + pass the inode id
  + return the node if it exist
* public Node getParent();
  + go to the TreeDataStructure constructor
  + get the parent of the Inode passed
  + return the parent
* public int size();
  + go to the TreeDataStructure constructor
  + add up all the current amount of nodes
  + return the number of nodes
* public String toString();
  + go to the TreeDataStructure constructor
  + convert the data to a string
  + return the string
* public String getId();
  + go to the TreeDataStructure constructor
  + use the node that called the method to find it’s id
  + return the id
* public void printTree();
  + go to the TreeDataStructure constructor
  + use toString() to get the node information
  + print out the result of toString for all nodes.