Group 6 – Lab5

Student Name	Github	Contribution
Terry Ulery	Tujin	2
Kristen Tourek	csiebler86	2
Joe Cruz		0 - dropped?

https://github.com/Tujin/CSE220-T6-LAB5

Lab 5 Pseudocode

```
class LiteralToken {
private:
      T literal
public:

      bool getLiteral(T lit)
            lit = literal
            return true

      bool setLiteral(const T &lit)
                  literal = lit
                  return true

      std::string toString()

            std::ostringstream ss
            ss << literal
            newStr = ss.str()
            return ss.str()

class Literal : public Token
public:

      getLiteral(T &lit) = 0
      setLiteral(const T &lit) = 0
      toString() = 0


IdentifierTree::IdentifierTree()
      // Initialize the Root Node to NULL (Signifying an Empty Binary
Tree)
      this->root = NULL


IdentifierTree::~IdentifierTree
      // Delete the Binary Tree Children (Should recursively delete the
objects lower in the tree)
      delete this->root->leftChild
```

```
      delete this->root->rightChild
      delete this->root

      // Initialize the Root Node to NULL (Signifying an Empty Binary
Tree)
      this->root = NULL;

IdentifierTree::insertIdentifier(Identifier *parentNode, Identifier
*identifier) {
      if (strcmp(identifier->getTokenString().c_str(),parentNode-
>getTokenString().c_str()) == 0)
            parentNode->lines->insertLineNode(identifier->lines->first-
>line)
 else if (identifier->getTokenString() < parentNode->getTokenString())
            // Identifier is lower than the Root
            if (parentNode->leftChild == NULL) //no left child
                  parentNode->leftChild = identifier
            else // left child exists
                  insertIdentifier(parentNode->leftChild, identifier)
       else if (identifier->getTokenString() > parentNode-
>getTokenString())
            Identifier is higher than the Root
            if parentNode->rightChild == NULL // no right child
                  parentNode->rightChild = identifier
            else // right child exists
            insertIdentifier(parentNode->rightChild, identifier)


      return


void IdentifierTree::insertIdentifier(Identifier *identifier)

      if(this->root == NULL)
            this->root = identifier
      else

            insertIdentifier(this->root, identifier)

IdentifierTree::printTree(Identifier *node)
      if (node != NULL)

            printTree(node->leftChild)

            string nodeStr = node->getTokenString()
            printf("%-16s\t", nodeStr.c_str())
            node->lines->printList()

            printTree(node->rightChild)

      return
```

TestCases

Function to test	Result
insertLineNode(int)	Line node successfully added to list in token
insertIdentifier(*Token)	Token inserted into tree
insertIdentifier(*Token, *Token)**	Token inserted into tree
printTree(*Token)	Tree successfully prints
printList()	List of line numbers prints successfully
Main(NEWTON.PAS)	Program executes properly