```
////////////////////////////////////////////////////////////////////////////
////Class:          CS 445
////Semester:       Fall 2011
////Assignment:     Homework 4
////Author:         Colby Blair
////File name:      tree_symtab_insert.c
////////////////////////////////////////////////////////////////////////////

#include "tree.h"
#include "symtabc.h"
#include "tree_symtab_insert.h"
#include "parser.tab.h"
#include "main.h"


int tree_symtab_insert_variableDefinition(struct tree *t)
{
        //t = variableDefinition, so get the variableDeclarator subtree
        struct tree *t_sub = NULL;
        tree_get_subtree("variableDeclarator", t, &t_sub);
        if(t_sub == NULL)
        {
                return(1); //failure
        }

        //get any aux flags for the variable definition. So far, just
        // 'const'
        char *aux_flag = tree_get_opt_aux_flag(t);

        //get symbol pointer to variable name
        //TODO: need to use more generic getter
        struct tree_token *symbol = t_sub->kids[0]->kids[0]->leaf;

        //get optional type
        char *type = tree_get_opt_type(t_sub);
        //TODO: could also find type from assignment
        // expression

        //if variable already declared, fail
        bool status = SymTab_insert(symbol->text, type, aux_flag, table_);
        if(status == false)
        {
                fprintf(stderr,
                 "ERROR: variable '%s' redeclared in '%s' on line '%d'\n",
                        symbol->text, symbol->fname, symbol->lineno);
                exit(ERROR_SEMANTIC);
        }

        return(0); //success
}


int tree_symtab_insert_methodDefinition(struct tree *t)
{
        //t = methodDefinition, so get the ident subtree
        struct tree *t_sub = NULL;
        tree_get_subtree("ident", t, &t_sub);
        if(t_sub == NULL)
        {
                return(1); //failure
        }

        //get symbol pointer to function name
        //TODO: need to use more generic getter
        struct tree_token *symbol = t_sub->kids[0]->leaf;

        //if function already declared, fail
        //TODO: type lookup, aux lookup
        bool status = SymTab_insert(symbol->text, "function", NULL, table_);
        if(status == false)
        {
                fprintf(stderr,
                 "ERROR: function '%s' redeclared in '%s' on line '%d'\n",
```

```
                        symbol->text, symbol->fname, symbol->lineno);
                exit(ERROR_SEMANTIC);
        }

        return(0); //success
}


int tree_symtab_insert_parameterDeclaration(struct tree *t)
{
        //t = parameterDeclaration, so get the ident subtree
        struct tree *t_sub = NULL;
        tree_get_subtree("ident", t, &t_sub);
        if(t_sub == NULL)
        {
                return(1); //failure
        }

        //get any aux flags for the parameterDeclaration.
        char *aux_flag = tree_get_opt_aux_flag(t);

        //get symbol pointer to variable name
        //TODO: need to use more generic getter
        struct tree_token *symbol = t_sub->kids[0]->leaf;

        //get optional type
        char *type = tree_get_opt_type(t);
        //TODO: could also find type from assignment
        // expression

        //if variable already declared, fail
        bool status = SymTab_insert(symbol->text, type, aux_flag, table_);
        if(status == false)
        {
                fprintf(stderr,
                 "ERROR: variable '%s' redeclared in '%s' on line '%d'\n",
                        symbol->text, symbol->fname, symbol->lineno);
                exit(ERROR_SEMANTIC);
        }

        return(0); //success
}
```