

Include files

```
//-----  
//C++ standard include files  
//-----  
#include <cstdlib>  
#include <cstring>  
#include <iostream>  
#include <fstream>  
#include <iomanip>  
#include <string>  
using namespace std;  
//-----  
//Symbol Table include files  
//-----  
#include "List.h"  
#include "Typ.h"  
#include "Sym.h"  
#include "Label.h"  
#include "Namespace.h"  
#include "Locality.h"  
#include "SymbolTable.h"
```

Source File Location

~tt/cs4173/SymbolTable/

Locality Management

```
{  SymbolTable ST;      //Create Locality 0 and insert standard symbols  
    ST.NewLocality();   //Create Locality 1  
    ST.PopLocality();   //Discard Locality 1  
    ST.Print(cout);     //Print Locality 0  
}
```

Variables

```
ST.NewLocality();

Sym* b=new VariableSymbol("b",tbool);    //var b:Boolean;
ST.Insert(b);
Sym* c=new VariableSymbol("c",tchar,1);    //var c:char;
ST.Insert(c);
Sym* i=new VariableSymbol("i",tinteger,1,6); //var i:integer;
ST.Insert(i);
Sym* r=new VariableSymbol(string("r"),treal); //var r:real;
ST.Insert(r);

ST.Print(cout);
ST.PopLocality();
```

Arrays

```
ST.NewLocality();
Typ* tndx=new Range(tinteger,"1","10");
Typ* tary=new Array(tndx,treal);
Sym* A=new VariableSymbol("A",tary);    //var A:array[1..10] of real;
ST.Insert(A);

ST.Print(cout);
ST.PopLocality();
```

Procedures

```
List<Typ*>* pp=new List<Typ*>;    //Parameter list types
pp->Insert(ST.TVoid());    //The first parameter (return
                           //type) of a procedure must i
                           //have type void
pp->Insert(ST.TInteger());    //Integer Type
pp->Insert(ST.TInteger());    //Integer Type
Typ* tp=new Subprogram(pp);    //Subprogram type
Sym* p=new ProcedureSymbol("swap",tp); //procedure swap(m,w:integer);
ST.Insert(p);
```

Functions

```
List<Typ*>* fp=new List<Typ*>;    //Parameter list types
fp->Insert(ST.TInteger());    //The first parameter is the
                           //return type
fp->Insert(ST.TReal());    //Real Type
Typ* tf=new Subprogram(fp);    //Subprogram type
Sym* f=new FunctionSymbol("round",tf); //function round(r:real):integer;
ST.Insert(f);
```

Program

```
Sym* ps=new ProgramSymbol("Test");    //program test;  
ST.Insert(ps);
```