

Plan for Today

Symbol Table Uses

- "pass info from declarations to uses"

Scope

- environments
- static versus dynamic scope
- scope examples
- scoping for MiniJava

Symbol Table Implementation

- info maintained and basic operation
- how type information is represented with SymTable and Type data structures

CS453 Lecture

Symbol Table

1

Scope

Terms

- environment
- scope

Example scopes

- global scope
- file scope
- named space
- package
- unnamed scopes

Scoping in MiniJava

CS453 Lecture

Symbol Table

2

Static versus Dynamic Scope

Static Scope

- also called lexical scope because can determine scoping by analyzing the program
- each use of a variable is bound to a location statically

Dynamic Scope

- each use of a variable is bound to the most recently visible defined value for that same variable name

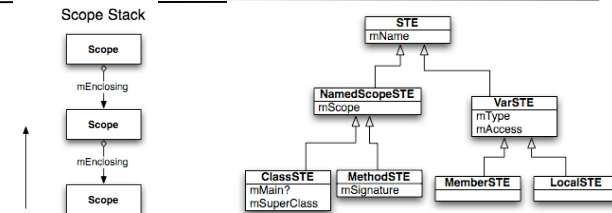
```
int x = 0;
int f () { return x; }
int g () { int x = 1; return f(); }
```

CS453 Lecture

Symbol Table

3

SymTable and STE classes



SymTable interface (DO NOT change the outputDot methods)

- void insertAndPushScope(NamedScopeSTE ste)
- void pushScope(String name)
- void popScope()
- STE lookup(String sym)
- void insert(STE ste)
- int outputDot(java.io.PrintStream out, int nodeCount)

CS453 Lecture

Symbol Table

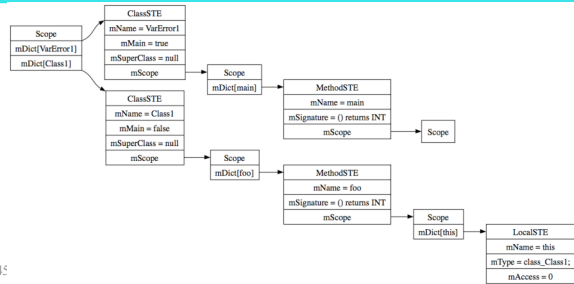
4

Example SymTable dot output

```

class VarError1{
  public static void main(String[] a){ System.out.println(1); } }

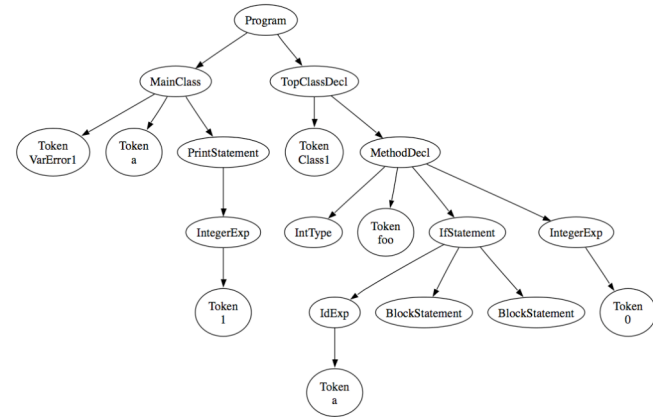
class Class1 {
  public int foo() {
    if (a) {} else {};
    return 0;
  }
}
    
```



CS45

5

Using the SymTable interface



CS453 Lecture

Symbol Table

6