Compilers

Arthur Hoskey, Ph.D.
Farmingdale State College
Computer Systems Department

Today's Lecture

- Semantic Analysis

- Compiler Front End
  - Lexical Analysis
  - Parsing
  - **Semantic analysis**
- Compiler Back End
  - Intermediate code generation
  - Optimization
  - Target code generation

Compiler Phases

Semantic Analysis

- **Semantic Analysis** – Semantic analysis examines the meaning (semantics) of the program on the basis of its syntactic structure.
- Finishes the analysis task by performing a variety of correctness checks (for example, enforcing type and scope rules).
- Taken from Crafting a Compiler 2nd edition by Fischer, Cytron, and Le Blanc, 2010.

Semantic Analysis

- The parsing phase cannot find certain types of errors in a program so a semantic analysis phase may be required.
- Most parsers use a context-free grammar which is not powerful enough to find certain types of errors.
- For example, cannot find type mismatch errors using a context-free grammar.

Semantic Analysis Error Types

- Semantic analysis checks for the following types of errors (this is not an exhaustive list):
- **Undeclared Variables** - Checks that variables are properly declared. For example:

```
int x;
x = 10;
y = 20;  // y has not been declared
```

- **Type Checking** - Checks that variable types are compatible in statements.

```
int x = 10;
String y = "Compiler";
x = y;   // Type mismatch error
```

- **Duplicate Variable Declaration**

```
int x;
int y;
int x;    // Variable x has already been declared
```

- **Undefined Method** – Check that a method definition exists for a given method call.

```
int x;
x = 10;
show();   // This will be an error if show() was not defined
```

- **Wrong Number of Parameters**

```
void greeting() {
   System.out.println("Hello");
}

int x;
x = 10;
greeting("Hi");    // The greeting method takes 0 parameters
```

Semantic Analysis Error Types

- **Improper Use of a Keyword** - Checks if the keyword can be used in a particular part of a program. For example, in Java a break can only appear inside a switch or a loop.

```
int x = 0;
while (x < 3) {
    x++;
}
break;   // Break cannot be used outside a switch or loop in Java
```

Semantic Analysis Error Types

Semantic vs Syntactic Analysis

- Here is syntactically correct Java code that has a semantic error:

```
int x;
x = 10;
y = 20;  // y has not been declared
```

Syntactic Analysis

- Each line has the correct Java syntax.
- They all properly use semicolons at the end of the line.
- The declaration uses a valid type (int). The type appears before the variable and there is a space between them.
- The assignments have identifiers to the left of =.

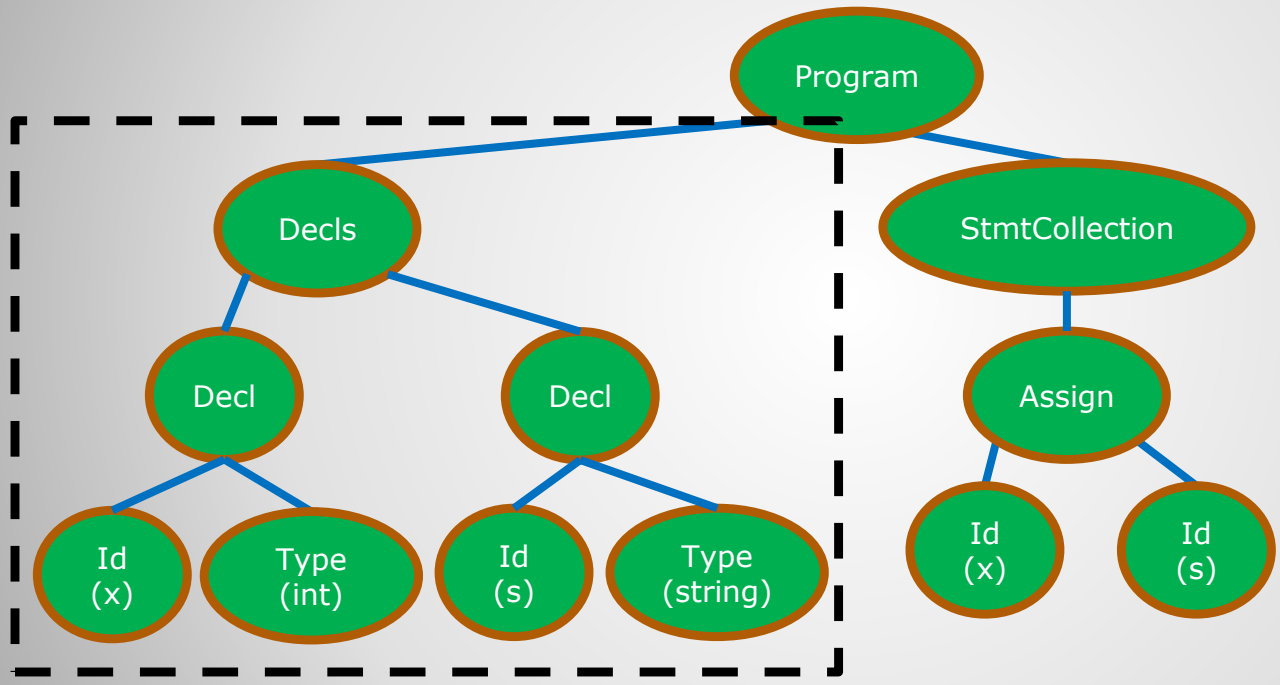
Semantic Analysis

- The issue is that y has not been declared. y is being used properly as an identifier from a syntax perspective, but it was never declared (this is a semantic issue).
- In general, it is easier to leave semantic analysis separate from syntactic analysis.
- It is a cleaner design to perform syntactic and semantic analysis separately.

Symbol Table

- **Symbol Table** - Data structure that stores information about variables and methods (names, data types, etc…).
- Symbol table entries are created prior to the semantic analysis phase.
- For example, when a variable declaration is encountered (prior to semantic analysis) an entry for that variable is added to the symbol table.
- The semantic analyzer can use the symbol table to check for errors (undeclared variables, type mismatch error etc…).

- Populate symbol table before semantic analysis.
- Add entries for each declared variable to the symbol table.

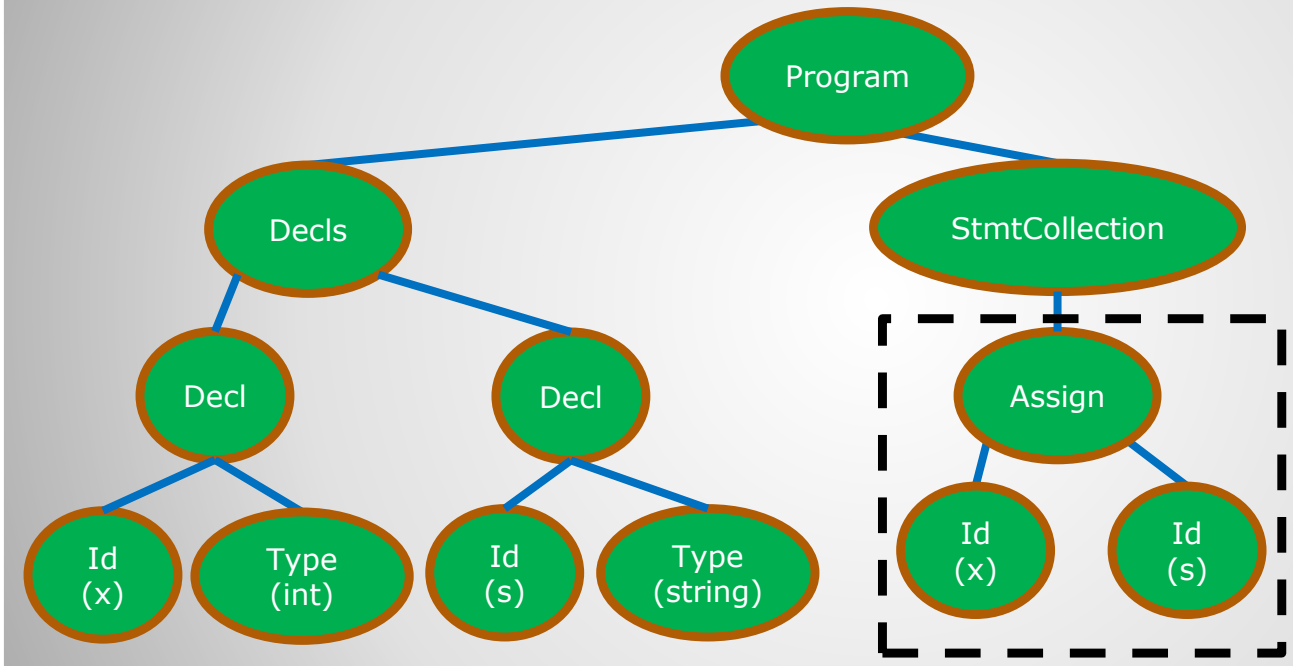


Symbol Table

Name	Type
x	int
s	string

Populate Symbol Table

- Type Mismatch. Traverse the tree and check that data types of variables match as necessary (in assignments, calculations, etc.).
- For example, when processing the assignment, lookup the variable types of x and s in the symbol table.



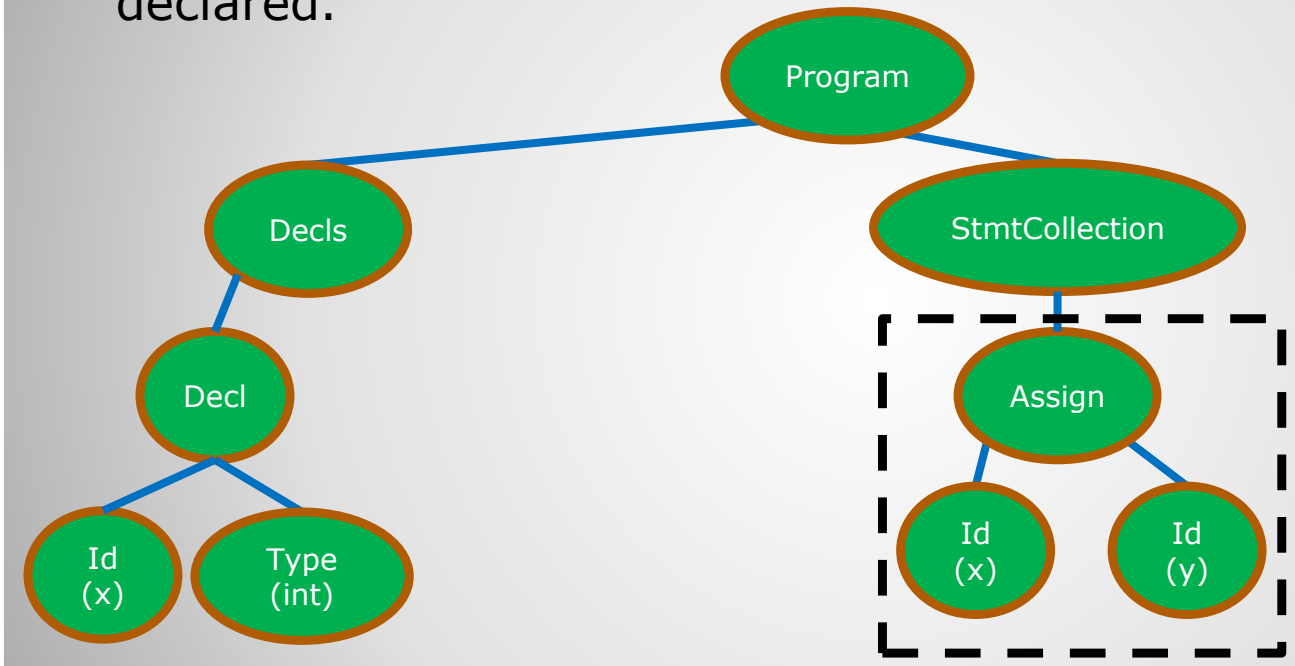
Symbol Table

Name	Type
x	int
s	string

Type mismatch error in the assignment. x is an int and s is a string.

Check for Type Mismatch

- Undeclared Variable. Traverse tree and confirm that variables were declared.
- For example, when processing the assignment, lookup the variable names in the symbol table to make sure they were declared.



Symbol Table

Name	Type
x	int

← **Undeclared Variable. y is not in the symbol table.**

Check for Undeclared Variable

- **End of Slides**

End of Slides