Plan for Today

MiniJava types and type rules
- representation and interpretation will be discussed while doing IRT generation
- how type information is represented with SymTable and Type data structures
- what type errors can occur in some of the AST nodes

Type implementation in the MiniJava compiler

```java
public class Type {
    public static final Type ARRAY = new Type();
    public static final Type BOOL = new Type();
    public static final Type INT = new Type();
    // class type map (key: class name, value: type)
    private static final HashMap<String, Type> classTypes = new HashMap<String, Type>();
}
```

Only one instance of the type object per atomic type and class type
- to determine if types are equal just compare references
- does the Type class know about inheritance?

Implementing type checking for MiniJava (Slide 1)

Visitor over AST will check for type errors at each AST node

<table>
<thead>
<tr>
<th>Syntax</th>
<th>AST production AST node</th>
</tr>
</thead>
<tbody>
<tr>
<td>id = Exp ; statement = {assign} id exp</td>
<td>[LINENUM,POSNUM] Undeclared variable VARNAME; [LINENUM,POSNUM] Invalid expression type assigned to variable VARNAME</td>
</tr>
<tr>
<td>Exp op Exp</td>
<td>exp = {op} [l_exp]:exp [r_exp]:exp</td>
</tr>
<tr>
<td></td>
<td>[LINENUM,POSNUM] Invalid left operand type for operator OP; [LINENUM,POSNUM] Invalid right operand type for operator OP</td>
</tr>
</tbody>
</table>
Implementing type checking for MiniJava (Slide 2)

<table>
<thead>
<tr>
<th>Syntax</th>
<th>AST production AST node</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Image" /></td>
<td><img src="image2" alt="Image" /></td>
</tr>
</tbody>
</table>

Errors

- invalid operand type for operator `!`
- invalid operand type for new array operator
- operator length called on non-array type

Implementing type checking for MiniJava (Slide 3)

<table>
<thead>
<tr>
<th>Syntax</th>
<th>AST production AST node</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image3" alt="Image" /></td>
<td><img src="image4" alt="Image" /></td>
</tr>
</tbody>
</table>

Errors

- class `CLASSNAME` does not exist
- method `METHODNAME` does not exist in class `CLASSNAME`
- invalid argument type for method `METHODNAME`

Implementation Plan

Test-driven approach
- Write test cases for
  - one AST node at a time
  - one type check at a time
  - one possible type at a time (start with atomic types)
- Set up a regression testing script
  - capture your compiler output on test case to a temp file
  - compare output to a handwritten output for test case
- Implement
  - one AST node at a time
  - one type check at a time
  - one possible type at a time (start with atomic types)

Advantages
- turn in your program at any point to get partial credit
- separate two most difficult pieces: understanding MiniJava typing and implementing the typecheck with the provided data structures