

# Compiler Project, Stage 3: Typechecking II

Computer Science 371  
Amherst College  
Fall 2015

This assignment, due on **Tuesday, October 27**, focuses on the second part of typechecking.

## 1 Your Goal

Your goal is implement the second phase of typechecking, which processes the bodies of the methods in the input program.

## 2 Things You'll Probably Want to Do

The only new files this week are in directory `tests3`. You should go into your `cs371` directory and do

```
cp -r hw2 hw3          // Make a new copy of your directory.
cp -r ~lamcgeoch/cs371/hw3/* hw3
[chmod -R a+w hw3]    // Fix permissions only if you're working in a shared directory.
```

In `hw3`, I suggest that you make a second copy of `analysis/AmhTraversal.java` and call it `Typechecker/Phase2.java`. Add lines to `Main.java` and `Typechecker/Typechecker.java` that mimic the calls used in phase 1.

I also suggest that you add the following class in directory `Typechecker`:

```
package minijava.Typechecker;

public class ExprType {
    static class Expr {}

    Expr expr;
    Type type;

    ExprType (Expr e, Type t) {
        expr = e;
        type = t;
    }
}
```

The purpose of these classes is to get you ready for the building of your intermediate tree, which can be done as part of phase 2. An `ExprType` object rolls together both a type and a fragment of intermediate code that computes a value. For now we ignore intermediate code, so class `Expr` is empty.

There are numerous node classes for which the return type of `process(NodeType)` should be `ExprType`. These include almost all versions of `process()` that appear in **Phase2.java** at and

below the entry for *process(PExpr)*. (PArgList, AListArgList, PEmptydim, and AEmptyDim seem to be the exceptions.) To return a type as an ExprType, simply use

```
return new ExprType (null, type);
```

where **type** is the chosen type.

As you edit **Phase2.java**, don't remove any methods or any of the calls that are part of the traversal. You might need them in building the intermediate tree. (Maybe not, but why prematurely optimize your code?)

Throw an exception whenever you encounter a typechecking error. If there are no errors, print the contents of the symbol tables that you have built.

### 3 Test Instances

There are many new test instances in **tests3**. Feel free to send me more to share with your classmates.

### 4 Submitting Your Work

Submit your project electronically at the usual site.