City University London

IN2009 Language Processors (2007/2008)

Assessment 2 (Type Checking)

Organisation: This coursework assessment may be completed individually, or in pairs, and there is no penalty for working in a pair. You will hand work in as individuals but the handin procedure will allow you to say who you collaborated with (and you must do so!). *Pairs* only – not threesomes!

Hand in: The deadline and the electronic handin procedure, and exactly what you should hand-in, are documented online. Obviously you should change file and directory permissions while you are working so that your work is not visible to others – remember plagiarism carries severe penalties. Finally, guard your work, don't risk plagiarism charges by leaving a USB key with your work around, or "sharing answers" etc.

Weighting: This component is worth 100% of assessment 2 (or if you prefer, 18% of the entire module).

The task

Modify the MiniJava implementation to undertake *type checking*.

Details

You will extend the MiniJava implementation, including the typechecker. Copy the MiniJava implementation directory to where you want to work, using:

```
cp -R /soi/sw/courses/daveb/IN2009/minijava/chap5
```

This is available as a zip file on Cityspace too. Files README in the various directories give a brief description of the structure of the implementation, and the symbol table mechanism and typechecker were studied in Session 5.

1. (a) Add the following three binary operators:
   - `/` (divide)
   - `>` (greater than - NOTE CHANGE TO BRIEF!)
   - `^` (XOR - e**X**clusive **OR**)

   (b) Add the operator `a?b:c` (shortcut for `if(a){b}else{c}`) to your implementation.

   These all have the same syntax and meaning as they do in Java (you may need to research this if you are not familiar with how it works!), and appropriate typechecks should be implemented for them (`op1 ? op2 : op3` returns `op2` if `op1` is true, and returns `op3` if `op1` is false). [30% in total]

2. Add appropriate typechecking for the `do` statement you implemented in assessment 1 (for those who did not hand in the first coursework, I will publish some guidance on what needs to be added - this will be published after two weeks from the deadline of assessment 1). [50%]

3. For highest marks: add appropriate typechecking for the `throw` and `try` statements implemented in assessment 1. ***Please do not attempt this section until you have completed the previous two - they are worth a much higher proportion of the marks!*** [20%]