# Compilers Coursework 2 Report

## Team Members: Jasmine Lu, Theodore Turner, Yee Chong Tan

### Optimisation Algorithm

**Arithmetic Operators:**

If an arithmetic instruction is detected, first the calculation is carried out by the function:

**public Number optimizedOperations(InstructionList instructionList, InstructionHandle handle, ConstantPoolGen cpgen)**

The function returns the result of the calculation, which depends on the operation (**ADD, SUB, MUL, DIV, NEG, REM**) and also the type of the value (**INT, LONG FLOAT DOUBLE**).

In order to get the correct values to calculate the result this function is called:

**public Number getValueFromStack(InstructionList instructionList, InstructionHandle handle, ConstantPoolGen cpgen)**

This function iterates through the instruction list until it finds the most recent value that has been pushed onto the stack from the bytecode instructions:

**BIPUSH, SIPUSH, ICONST, LCONST, FCONST, DCONST, LDC, LDC2\_W**

The value is also deleted from the instruction list, since it will be replaced by the new calculated result.

The value is deleted with the function:

**public void deleteInstruction(InstructionList instructionList, InstructionHandle handle)**

**Comparison Operators:**

The comparison operators (**LCMP, FCMPG, FCMPL, DCMPG,DCMPL**) are dealt with in a very similar way to the arithmetic operators. Firstly, they are detected by a Boolean function:

**public Boolean comparisonOperator(InstructionHandle handle)**

These are then also analysed with the **optimizedOperations** function, where the 2 values are gathered from the **getValueFromStack** function, and compared, returning an **int** value of 0, 1 or -1 and the instruction is deleted like before.