1 Exercises

Exercise 1. Consider the following *j*-- program:

```
package pass;

import java.lang.Integer;
import java.lang.System;

public class Sum {
    private static String MSG = "SUM = ";
    private int n;

    public Sum(int n) {
        this.n = n;
    }

    public int compute() {
        int sum = 0, i = n;
        while (i > 0) {
            sum += i--;
        }
        return sum;
    }

    public static void main(String[] args) {
        int n = Integer.parseInt(args[0]);
        Sum sum = new Sum(n);
        System.out.println(MSG + sum.compute());
    }
}
```

a. How does pre-analysis (`JCompilationUnit.preAnalyze()`) of the program work?

b. How does analysis (`JCompilationUnit.analyze()`) of the program work?

c. How are the declarations of the local variables `sum` and `i` handled in the `compute()` method?

d. How are offsets assigned to the parameters/variables in the program's constructor and the two methods?

e. How is the simple variable `n` resolved in the `main()` method?

f. How is the field selection `MSG` resolved in the `main()` method?

g. How are the message expressions `System.out.println(...)` and `sum.compute()` resolved in the `main()` method?

h. How is argument to `System.out.println()` analyzed in the `main()` method?

Exercise 2. When can you cast an expression of type `Type1` to another type `Type2`?

Exercise 3. Consider the following *j*-- program:

```
public class Mystery {
    public int f(int x) {
        int y = x * x;
        return z;
    }
}
```

Is the program syntactically/semantically correct? If not, why and how does *j*-- figure it out?

Exercise 4. How would you do semantic analysis for the do-while statement, ie, implement `analyze()` in `JDoWhileStatement.java`?

2 Solutions to Exercises

Solution 1.

a. See section 4.4 of our text.

b. See section 4.5 of our text.

c. See section 4.5.2 of our text.

d. See section 4.5.2 of our text.

e. See section 4.5.3 of our text.

f. See section 4.5.4 of our text.

g. See section 4.5.4 of our text.

h. See section 4.5.5 of our text.

Solution 2. See section 4.5.6 of our text.

Solution 3. The program is syntactically correct, but semantically wrong since the variable `z` is not declared before use. During analysis of the return statement, the simple variable `z` is looked up in the chain of contexts, starting at the local context. The lookup is unsuccessful, and an error is reported that the variable has not been declared.

Solution 4.

```java
public JDoWhileStatement analyze(Context context) {
     body = (JStatement) body.analyze(context);
     condition = condition.analyze(context);
     condition.type().mustMatchExpected(line(), Type.BOOLEAN);
     return this;
}
```