Swinburne University Of Technology

Faculty of Science, Engineering and Technology

ASSIGNMENT COVER SHEET

Subject Code:	CSO30023
Subject Title:	Languages in Software Development
Assignment number and title:	5, JavaCC – First Steps
Due date:	**September 22, 2014, 10:30, on paper**
Lecturer:	Dr. Markus Lumpe

Your name:_____

Marker's comments:

Problem	Marks	Obtained
1	-	-
2	72	
Total	72	

Extension certification:

This assignment has been given an extension and is now due on _____

Signature of Convener:_____

Problem Set 5: JavaCC – First Steps

Problem 1

Study the description if the JavaCC grammar file available on the course page. Review Lab4 and MiniJava and identifier how a BNF is mapped to a JavaCC specification.

Problem 2

Consider the following BNF specification (productions in italic, terminals in bold font):

```
CompilationUnit    ::=  Statements <EOF>

Statements         ::=  Statement (";" Statement)*

Statement          ::=  <IDENTIFIER> ":=" Expression
                      |  "print" "(" ExpressionList ")"

Expression         ::=  Term (("+" | "-") Term)*

Term               ::=  Primary (("*" | "/") Primary)*

Primary            ::=  <IDENTIFIER>
                      |  <INTEGER>
                      |  "(" Expression ")"
                      |  "{" Statement "," Expression "}"

ExpressionList     ::=  Expression ("," Expression )*
```

Lexical Aspects:

- **Comments and Whitespace Characters**

  A comment may appear between any two tokens and starts with "**//**" and goes to the end of the line. Whitespace characters include "** **", "**\t**", "**\n**", and "**\r\n**".

- **<IDENTIFIER>**

  An *identifier* is a sequence of letters, digits, and underscores, starting with a letter. Uppercase letters are distinguished from lowercase.

- **<INTEGER>**

  A sequence of decimal digits is an *integer constant* that denotes the corresponding integer value.

Program Test:

```
a := 5 + 3; b := {print (a, a-1), 10 * a}; print (b)
```

Define a front-end in JavaCC for this grammar using `StraightLine` as parser name.

Use to following main method:

```
public static void main( String[] Args )
{
   try
   {
     StraightLine lParser = new StraightLine( new FileInputStream( Args[0] ) );
     lParser.CompilationUnit();
     System.out.println( "SUCCESS" );
   }
   catch (ParseException e)
   {
     System.out.println( "Syntax Error : \n"+ e.toString() );
   }
   catch (FileNotFoundException e)
   {
     System.out.println( e.toString() );
   }
}
```

Submission deadline: Monday, September 22, 2014, 10:30.

Submission procedure: on paper.