**DCU Engineering & Computing  
Assignment Submission**

|  |  |
| --- | --- |
| **Student Name(s):** | **Alan Kehoe** |
| **Student Number(s):** | **10735389** |
| **Programme:** | **CASE4 - BSc in Computer Applications (Sft.Eng.)** |
| **Project Title:** | **A Top-Down Parser with Semantic Analysis for the ImpCA Language** |
| **Module code:** | **CA448** |
| **Lecturer:** | **Dr. David Sinclair** |
| **Project Due Date:** | **16-DEC-2013** |

|  |
| --- |
| **Declaration**  **I the undersigned declare that the project material, which I now submit, is my own work. Any assistance received by way of borrowing from the work of others has been cited and acknowledged within the work. I make this declaration in the knowledge that a breach of the rules pertaining to project submission may carry serious consequences.**  **I am aware that the project will not be accepted unless this form has been handed in along with the project.** |

The first thing I done was convert your grammar into a .jjt file the grammar will then be compiled with jjtree however it gave me allot of warnings about look ahead and left recursion detected in the grammar.

The left recursion was solved using new non-terminals. After solving left recursion the look ahead warnings were next these were solved by mixing around the grammar. By default you get a look ahead of 1 but it wasn’t too hard to change the grammar to work with only look ahead of 1 for example changing

void statement () :{}{

<ID><ASSIGN>(expression() | <STRING>)\*

|<EXLM>expression()

|<QUESTION><ID>

|<ID><LBR>arg\_list()<RBR>

|<LCBR>(statement()<SEMIC>)\*<RCBR>

|<IF>condition()<THEN>statement()

|<IF>condition()<THEN>statement()<ELSE>statement()

|<WHILE>condition()<DO>statement()

|{}

}

to

void statement () :{}{

<ID>(assignment() | <LBR>arg\_list()<RBR>)

|<EXLM>expression()

|<QUESTION><ID>

|<IF>condition()<THEN>statement() (<ELSE>statement() | {})

|<WHILE>condition()<DO>statement()

|{}

}

void assignment (): {}{

<ASSIGN>(expression() | <STRING>)

}