CA4003 Compiler Construction Assignments

Course Notes	**Assignments**

Assignment 1	Assignment 2	Resit

Assignment 1

Assignment 1: A Lexical and Syntax Analyser

The aim of this assignment is to implement a lexical and syntax analyser using JavaCC for a simple language called CCAL.

The details of the CCAL language are given here.

Submission

This is an individual assignment. You should submit, by email to David.Sinclair@computing.dcu.ie, ***all the source files*** (JavaCC file and other files which were used to generate your parser) in a Winzip file along with ***a declaration*** that this is solely your own work (except elements that are explicitly attributed to another source), ***a brief description*** of each of the major components of your lexical and syntax analyser that describes how you implemented them, by **10am on Monday 13th November 2017**.

Submissions without the declaration of that the assignment is the student's own work will not be assessed. The assignment carries 15 marks and **late submissions will incur a 1.5 mark penalty for each 24 hours after the submission**.

Please click here to review the School's policy on plagarism. All submissions will be checked for plagarism and severe penalties will apply.

Assignment 2

Assignment 2: Semantic Analysis and Intermediate Representation

Aim

The aim of this assignment is to add semantic analysis checks and intermediate representation generation to the lexical and syntax analyser you have implement in Assignment 1. The generated intermediate code should be a 3-address code and stored in a file with the ".ir" extension.

You will need to extend your submission for Assignment 1 to:

- Generate an Abstract Syntax Tree.
- Add a Symbol Table **that can handle scope**.
- Perform a set of semtantic checks. This following is a list of typical sematic checks:
    - Is every identifier declared within scope before its is used?
    - Is no identifier declared more than once in the same scope?
    - Is the left-hand side of an assignment a variable of the correct type?
    - Are the arguments of an arithmetic operator the integer variables or integer constants?
    - Are the arguments of a boolean operator boolean variables or boolean constants?
    - Is there a function for every invoked identifier?
    - Does every function call have the correct number of arguments?
    - Is every variable both written to and read from?
    - Is every function called?
- Generate an Intermediate Representation using 3-address code.

  Feel free to add any additional semantic checks you can think of!

Submission

You should submit, by email to david.sinclair(at)dcu.ie, all the source files (JavaCC file and other files which were used to generate your parser) in a Winzip file along with a description of your abstract syntax tree structure and symbol table, how you implemented them and how you implemented the semantic checking and intermediate code generation, by **10am on Monday 11th December 2017**.

Submissions without the declaration of that the assignment is the student's own work will not be assessed. The assignment carries 15 marks and **late submissions will incur a 1.5 mark penalty for each 24 hours after the submission**.

Please click here to review the School's policy on plagarism. All submissions will be checked for plagarism and severe penalties will apply.

The results from both Assignments 1 and 2 can be found here.

CA4003 Compiler Construction Resit Assignment

The aim of this assignment is to have another attempt to build a complier for the CCAL language that has semantic checks and intermediate code generation. Your symbol table must be able to haandle scope. This is an individual assignment. Submisions will be checked very closely for signs of plagarism, for which **strict** penalties will apply. Please click here to review the School's policy on plagarism.

Submission

You should submit, by email to david.sinclair(at)dcu.ie, all the source files (JavaCC file and any other files that you used to generate your parser) in a **Winzip** file along with a description of:

- the scanne and parser,
- how you implemented your abstract syntax tree structure and symbol table,
- how you implemented the semantic checking, and
- how you generated the the 3-address intermediate code.

in a Word or PDF document by **1000am on Tuesday August 13th 2018**.

Your submission **must** include a declaration that the work submitted is solely your own work unless explicitly cited. Submissions without such a declaration will not be marked.

The assignment carries 30 marks and **late submissions will incur a 3 mark penalty for each 24 hours after the submission deadline**.