Programming Languages (Principles and Design): Exam, Section A info

Dr Paris Yiapanis

room: John Dalton E151
email: p.yiapanis@mmu.ac.uk

Section A will assess your Lab work

- One multi-part question. Total of 50/100 marks
- Section A tests your understanding on the three phases of your decaf compiler
- There are no previous questions/answers for Section A (if you've been working on this during labs, understood what you were doing and coming to labs for feedback you should have no problems)
- It's important also to watch the screencasts related lab work

Lexical Analyzer (aka scanner, lexer)

- You will have questions based on regular expressions
- Sample question:

- *Write a regular expression that defines an identifier. An identifier, in this context, must start with a lowercase letter, uppercase letter, or underscore, followed by a sequence of zero or more lowercase letters, uppercase letters, underscores, or digits.*

**note: if you've used fragments in your answer, make sure you include them, too.*

Parser

- Given a part of the decaf grammar, you should be able to write the corresponding ANTLR rules
- Also, pay attention to expression precedence (parser screencast #4, week 4). Understand how ANLTR handles precedence vs how a hand-crafted parser would handle it

Semantical Analyzer (Symbol Table)

- Familiarize yourself with the symbol table implementation provided in week 6 under the labs sub-section (the downloadable file).
- Be able to explain how the symbol table from above works (you may need to revise screencast #3 on semantics, week 6) and
- explain the functionality (or provide code) for the different methods that make up the symbol table.
- Symbol table implementation is explained in screencast #3 on semantics, week 6 after the (minute 20:47 onwards). Nevertheless, to understand how the symbol table works, you may need to watch the screencast from the beginning.

Semantical Analyzer (Semantic Rules)

- Be able to explain (or provide code on) how would you implement a given semantic rule (you should be able to implement any rule asked from the last page of the decaf language reference) – approx. 150 words.
- In your answer for implementing a given semantic rule, include discussion on:
  - how the symbol table will be used
  - which specific "enter" / "exit" methods will be used in the ScopeListener and in what way; and how their "context" will be used
  - Revise also screencast 4 from week 6