Compiler Construction:
CST parts Ib, II, II (General)
Diploma in Computer Science

Revised Syllabus for 1998/99

For examination purposes the syllabus listed in the course notes will be restricted to cover the following topics only:

```
Survey of language constructs and their implementation.
  Expressions, applicative structure, lambda expressions. Environments
  and a simple lambda evaluator. Evaluation of function calls using
  static and dynamic chains, Dijkstra displays. Situations where a simple
  stack is inadequate. Objects and inheritance. Implementation of labels,
  jumps, arrays and exceptions. L-value and r-value; choices for argument
  passing and free-variable association. Dynamic and static binding.
  Dynamic and static types, polymorphism.

Survey of execution mechanisms.
  The spectrum of interpreters and compilers; compile-time and run-time.
  Structure of a simple compiler.

Lexical analysis and syntax analysis.
  Regular expressions and finite state machine implementations. Grammars,
  Chomsky classification of phrase structured grammars. Parsing
  algorithms: recursive descent and SLR(k). Syntax error
  recovery.

Simple type-checking.
  Type of an expression determined by type of subexpressions; inserting
  coercions.  Polymorphism.

Translation phase.
  Intermediate code design. Translation of commands, expressions and
  declarations. Translating variable references into access paths.

Code generation.
  Typical machine codes. Code generation from
  intermediate code. Simple optimisation.

Compiler compilers.
  Summary of Lex and Yacc.

Runtime.
  Object modules and linkers. Resolving external references.
```

In particular the following topics are not examinable for 1998/99:

```
Java virtual machine.
Precedence parsing.
Code generation from the parse tree (section 3.5 and 3.6 of the notes).
Debuggers, break points and single step execution. Profiling, portability.
```

Alan Mycroft
May 1999