c1:The study of programming languages is valuable for a number of reasons:

– Increase our capacity to use different constructs

– Enable us to choose languages more intelligently

– Makes learning new languages easier.

Most important criteria for evaluating programming languages include:

– Readability, writability, reliability, cost

Major influences on language design have been machine architecture and software development methodologies

The major methods of implementing programming languages are: compilation(Translate high-level program   
into machine code), pure interpretation (No translation), and hybrid implementation (A high-level language program is translated to an intermediate language that allows easy interpretation)

c2:Development, development environment, and evaluation of a number of important programming languages

• Perspective into current issues in language design.

c3:BNF and context-free grammars are equivalent meta-languages

– Well-suited for describing the syntax of programming languages

• An attribute grammar is a descriptive formalism that can describe both the syntax and the semantics of a language

• Three primary methods of semantics description

– Operation, axiomatic, denotational

Syntax: the form or structure of the expressions, statements, and program units

Semantics: the meaning of the expressions, statements, and program units

c4: Syntax analysis is a common part of language implementation

• A lexical analyzer is a pattern matcher that isolates small-scale parts of a program

– Detects syntax errors

– Produces a parse tree

• A recursive-descent parser is an LL parser

– EBNF

• Parsing problem for bottom-up parsers: find the substring of current sentential form

• The LR family of shift-reduce parsers is the most common bottom-up parsing approach

c5

c6

c7

c8

c9

c10

c11

c12