2 - Characteristics of First-Generation Programming Languages (page 1)

[source: MacLennan, p. 92]

- the classic example: FORTRAN

- "In general, ... the structures of first generation languages are based on the structures of the computers in the early 1960's"

  - "...natural, since the only experience people had in programming was in programming these machines"...!
3 - First-Generation (page 2)

• "This machine orientation is especially apparent in first generation control structures"

  – non-nested

  – "depend heavily on the GOTO for building any but the simplest control structures"

  – "One exception ... the definite iteration statement [FORTRAN's DO-loop] ... which IS hierarchical in first-generation languages."

• "Recursive procedures are not permitted in most first-generation languages (BASIC is an exception)"

• "there is generally only one parameter passing mode (typically, pass by reference)"
"machine orientation ... can also be seen in the types of data structures provided ... patterned after the layout of memory on the computers available around 1960."

- "data structure primitives ... are fixed and floating point numbers of various precisions, characters, and logical values -- just the kinds of values manipulated by the instructions on these computers"

- "The data structure constructors are arrays and, in business-oriented languages, records, which are the ways storage was commonly organized."

- "As with control structures, first-generation languages provide little facility for hierarchical data organization (an exception is COBOL's record structure). That is, data structures cannot be nested."
5 - First-Generation (page 4)

• "characterized by a relatively weak type system;
  – that is, it is easy to subvert the type system or do representation-dependent programming."
  – "(Machine independence and portability were not major concerns in the first generation.)

• "Hierarchical structure is also absent from first-generation name structures, with disjoint scopes being the rule."
  – "variable names are bound directly and statically to memory locations since there is no dynamic memory management."
"syntactic structures ... are characterized by a card-oriented, linear arrangement of statements patterned after assembly language"

- "...most of these languages had numeric statement labels that are suggestive of machine addresses"

BUT they "go significantly beyond assembly languages ... in their provision of algebraic notation"

"Their usual lexical conventions are to ignore blanks and to recognize keywords in context."
7 - First-Generation (page 6)

- "In summary, the salient characteristics of the first generation are:
  - **machine** orientation and
  - **linear** structures."

- "second generation makes important moves in the directions of
  - **application** orientation and
  - **hierarchical** structure."