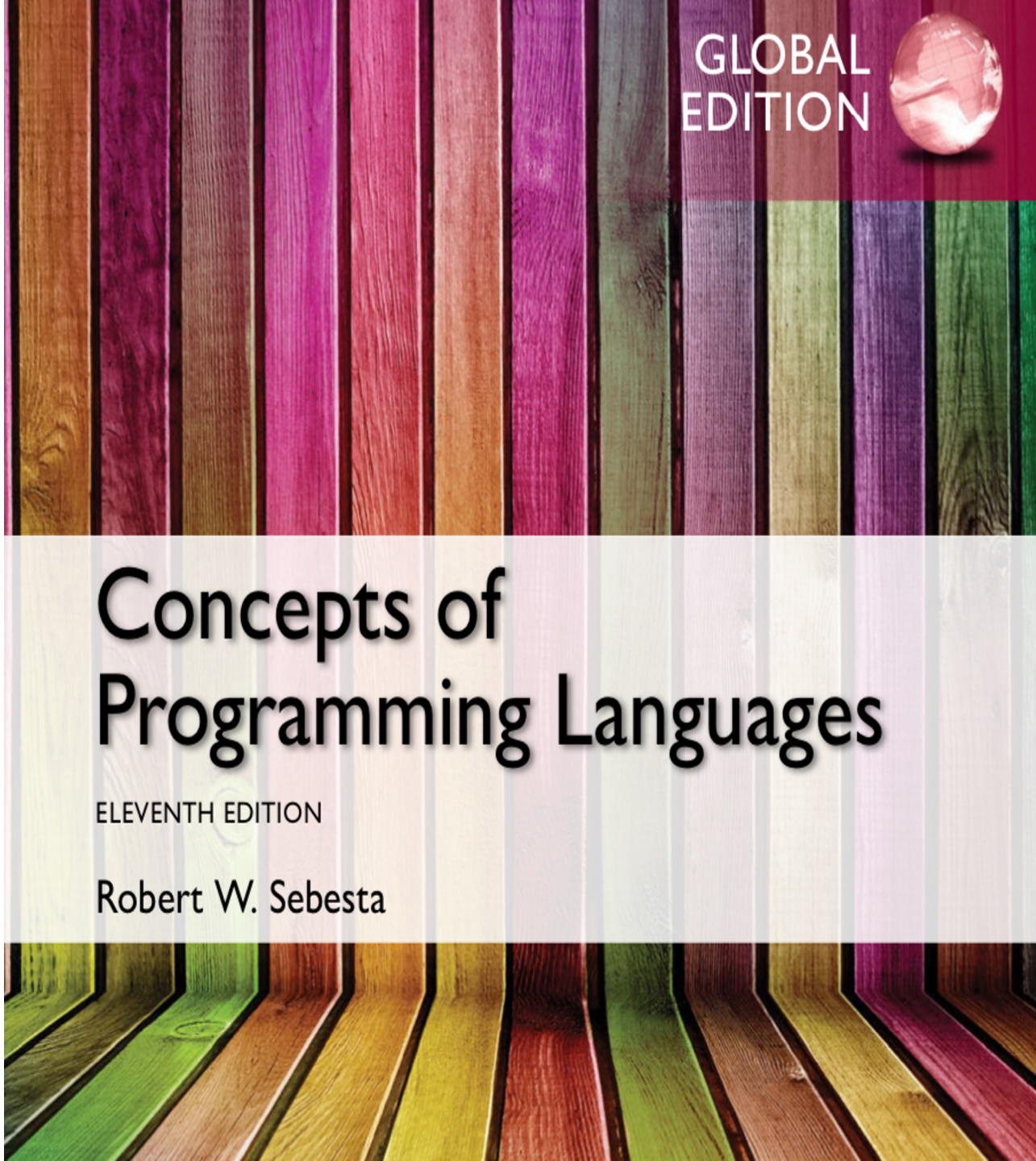
Chapter 2

Evolution of Major Programming Languages



Chapter 2 Topics

- Zuse's Plankalkül
- IBM 704 and Fortram
- ALGOL
- COBOL
- Functional Programming: Lisp
- C
- C++
- Prolog

Zuse Plankalkül

- Designed in 1945, but not published until 1972.
- Zeus was German Engineer, Plankalkül means program *calculus.*
- Never implemented. (No compilers)
- Advance data structure. (arrays, records, floats)
- include selection statement but did not include *else*
- Manuscripts included complex programs.(sorting array, test connectivity of graphs.)

Plankalkül Syntax

- **Assignment statement to assign the expression**
- **A[4] +1 to A[5]**

| A + 1 => A

V | 4 5 Operahand subscript

S | 1.n 1.n Data types —n bit integer

FORTRAN

- John Backus's team at IBM developed FORTRAN (for FORmula TRANslator) in 1955-1957.
- While FORTRAN was designed for numerical computation,
- it included control structures conditions and input/output.
- Initially used on IBM 704 in 1954
- FORTRAN popularity led to FORTRAN II in 1958, FORTRAN IV in 1962, leading it standardization in 1966, with revised standards coming out in 1977 and 1990.

FORTRAN

```
! Fortran 95 Example program
!  Input:   An integer, List_Len, where List_Len is less
!           than 100, followed by List_Len-Integer values
!  Output: The number of input values that are greater
!           than the average of all input values
Implicit none
Integer Dimension(99) :: Int_List
Integer :: List_Len, Counter, Sum, Average, Result
Result= 0
Sum = 0
Read *, List_Len
If ((List_Len > 0) .AND. (List_Len < 100)) Then
! Read input data into an array and compute its sum
   Do Counter = 1, List_Len
      Read *, Int_List(Counter)
      Sum = Sum + Int_List(Counter)
   End Do
```

FORTRAN

```
! Compute the average
   Average = Sum / List_Len
! Count the values that are greater than the average
   Do Counter = 1, List_Len
      If (Int_List(Counter) > Average) Then
         Result = Result + 1
      End If
   End Do
! Print the result
   Print *, 'Number of values > Average is:', Result
Else
   Print *, 'Error - list length value is not legal'
End If
End Program Example
```

ALGOL(ALGOrithmic Language)

- FORTRAN's success led to fear that IBM would dominate the computer industry.

- GAMM(German Applied Mathematics and Mechanics) and ACM(Association for Computing Machinery) created universal language ALGOL56(which led to ALGOL60 and ALGOL62)
- Later languages derived from ALGOL, including Pascal and Ada.

ALGOL(ALGOrithmic Language)

- Design of ALGOL
- ALGOL notation should be close to standard mathematics.
- ALGOL should be useful in describing algorithms
- Programs in ALGOL should be compliable to machine language.
- ALGOL should not be tied to a single architecture.
- Influence of ALGOL.
- ALGOL use was limited in US
- Backus and Naur developed the notation still used to express language syntax (BNF).

ALGOL(ALGOrithmic Language)

```
comment ALGOL 60 Example Program
 Input: An integer, listlen, where listlen is less than
        100, followed by listlen-integer values
 Output: The number of input values that are greater than
        the average of all the input values ;
begin
  integer array intlist [1:99];
  integer listlen, counter, sum, average, result;
   sum := 0;
   result := 0;
   readint (listlen);
   if (listlen > 0) ^ (listlen < 100) then
      begin
 comment Read input into an array and compute the average;
      for counter := 1 step 1 until listlen do
        begin
        readint (intlist[counter]);
        sum := sum + intlist[counter]
        end;
 comment Compute the average;
      average := sum / listlen;
 comment Count the input values that are > average;
      for counter := 1 step 1 until listlen do
        if intlist[counter] > average
          then result := result + 1;
 comment Print result;
      printstring("The number of values > average is:");
      printint (result)
      end
   else
      printstring ("Error-input list length is not legal";
 end
```

COBOL

- Commercial data processing was one of the earliest commercial applications of computers
- The US Defense Dept. funded to develop COBOL (COmmon Business-Oriented Language) which was standardized in 1960, revised in ‘61 and ‘62 and re-standardized in 1968, 1974 and 1984.
- Popular due to self-documenting style. Record structure made it easy to organize data.
- COBOL has influence on most database manipulation languages.

COBOL

```
IDENTIFICATION DIVISION.
PROGRAM-ID. PRODUCE-REORDER-LISTING.

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER. DEC-VAX.
OBJECT-COMPUTER. DEC-VAX.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT BAL-FWD-FILE ASSIGN TO READER.
    SELECT REORDER-LISTING ASSIGN TO LOCAL-PRINTER.

DATA DIVISION.
FILE SECTION.
FD  BAL-FWD-FILE
    LABEL RECORDS ARE STANDARD
    RECORD CONTAINS 80 CHARACTERS.

01  BAL-FWD-CARD.
    02 BAL-ITEM-NO          PICTURE IS 9(5).
    02 BAL-ITEM-DESC        PICTURE IS X(20).
    02 FILLER               PICTURE IS X(5).
    02 BAL-UNIT-PRICE       PICTURE IS 999V99.
    02 BAL-REORDER-POINT    PICTURE IS 9(5).
    02 BAL-ON-HAND          PICTURE IS 9(5).
    02 BAL-ON-ORDER         PICTURE IS 9(5).
    02 FILLER               PICTURE IS X(30).
FD  REORDER-LISTING
    LABEL RECORDS ARE STANDARD
    RECORD CONTAINS 132 CHARACTERS.

01  REORDER-LINE.
```

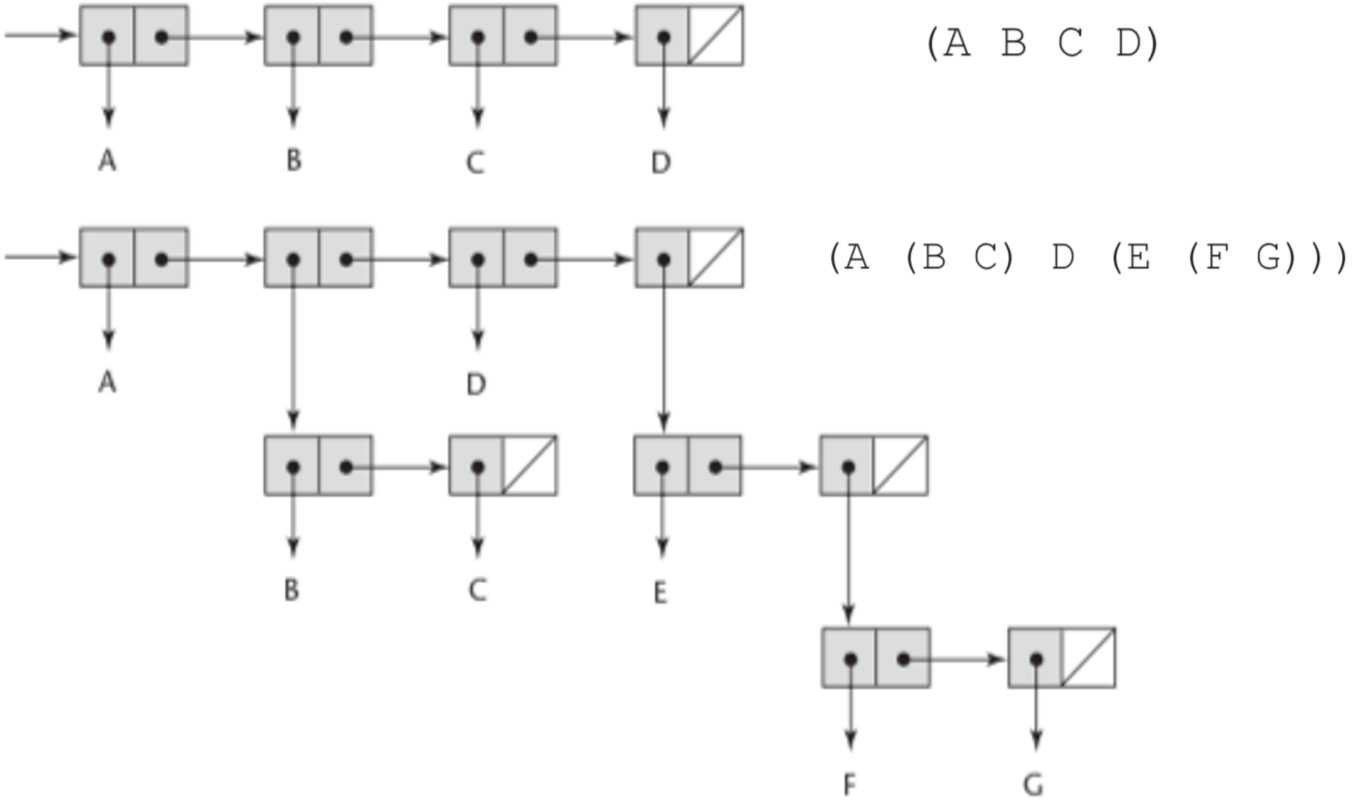
COBOL

```
     02 RL-ITEM-NO              PICTURE IS Z(5).
     02 FILLER                  PICTURE IS X(5).
     02 RL-ITEM-DESC            PICTURE IS X(20).
     02 FILLER                  PICTURE IS X(5).
     02 RL-UNIT-PRICE           PICTURE IS ZZZ.99.
     02 FILLER                  PICTURE IS X(5).
     02 RL-AVAILABLE-STOCK      PICTURE IS Z(5).
     02 FILLER                  PICTURE IS X(5).
     02 RL-REORDER-POINT        PICTURE IS Z(5).
     02 FILLER                  PICTURE IS X(71).

WORKING-STORAGE SECTION.
01   SWITCHES.
     02 CARD-EOF-SWITCH         PICTURE IS X.
01   WORK-FIELDS.
     02 AVAILABLE-STOCK         PICTURE IS 9(5).

PROCEDURE DIVISION.
000-PRODUCE-REORDER-LISTING.
     OPEN INPUT BAL-FWD-FILE.
     OPEN OUTPUT REORDER-LISTING.
     MOVE "N" TO CARD-EOF-SWITCH.
     PERFORM 100-PRODUCE-REORDER-LINE
          UNTIL CARD-EOF-SWITCH IS EQUAL TO "Y".
     CLOSE BAL-FWD-File.
     CLOSE REORDER-LISTING.
     STOP RUN.

100-PRODUCE-REORDER-LINE.
     PERFORM 110-READ-INVENTORY-RECORD.
     IF CARD-EOF-SWITCH IS NOT EQUAL TO "Y"]
          PERFORM 120-CALCULATE-AVAILABLE-STOCK
          IF AVAILABLE-STOCK IS LESS THAN BAL-REORDER-POINT
               PERFORM 130-PRINT-REORDER-LINE.

110-READ-INVENTORY-RECORD.
     READ BAL-FWD-FILE RECORD
          AT END
               MOVE "Y" TO CARD-EOF-SWITCH.

120-CALCULATE-AVAILABLE-STOCK.
ADD BAL-ON-HAND BAL-ON-ORDER
     GIVING AVAILABLE-STOCK.

130-PRINT-REORDER-LINE.
     MOVE SPACE                TO REORDER-LINE.
     MOVE BAL-ITEM-NO          TO RL-ITEM-NO.
     MOVE BAL-ITEM-DESC        TO RL-ITEM-DESC.
     MOVE BAL-UNIT-PRICE       TO RL-UNIT-PRICE.


      MOVE AVAILABLE-STOCK     TO RL-AVAILABLE-STOCK.
      MOVE BAL-REORDER-POINT   TO RL-REORDER-POINT.
      WRITE REORDER-LINE.
```

LISP

- John McCarthy of MIT developed LISP (LISt Processor) in 1950 for use in Artificial Intelligence
- Only two data types: atoms and List.
- Language pioneered in garbage collection and recursive procedure.
- It is not an imperative language, it does not run on standard computer architecture.
- Initial AI programs use LISP

LISP



LISP

```
; Lisp Example function
; The following code defines a Lisp predicate function
; that takes two lists as arguments and returns True
; if the two lists are equal, and NIL (false) otherwise
  (DEFUN equal_lists (lis1 lis2)
   (COND
     ((ATOM lis1) (EQ lis1 lis2))
     ((ATOM lis2) NIL)
     ((equal_lists (CAR lis1) (CAR lis2))
                 (equal_lists (CDR lis1) (CDR lis2)))
     (T NIL)
   )
)
```

SNOBOL

- SNOBOL(***S***tri***n***g ***O***riented Sym***bo***lic ***L***anguage) was developed by Griswold at AT&T Bell Labs.
- Frist String Processing Language. SNOBOL4 was powerful in pattern-matching capabilities.
- Early application was writing text editors.

BASIC

- JOHN and THOMAS originally designed this language to teach college students how to program in most user friendly way.
- Original design of BASIC was influenced by FOTRON.

```
REM  Basic Example Program
REM  Input:  An integer, listlen, where listlen is less
REM          than 100, followed by listlen-integer values
REM  Output: The number of input values that are greater
REM          than the average of all input values
  DIM intlist(99)
  result = 0
  sum = 0
  INPUT listlen
  IF listlen > 0 AND listlen < 100 THEN
REM  Read input into an array and compute the sum
    FOR counter = 1 TO listlen
      INPUT intlist(counter)
      sum = sum + intlist(counter)
    NEXT counter
REM  Compute the average
    average = sum / listlen
REM  Count the number of input values that are > average
    FOR counter = 1 TO listlen
      IF intlist(counter) > average
        THEN result = result + 1
    NEXT counter
REM  Print the result
    PRINT "The number of values that are > average is:";
          result
  ELSE
    PRINT "Error-input list length is not legal"
  END IF
END
```

C

- Dennis Ritchie of AT&T Bells Labs created based on early language BCPL and B to use it in writing a revised version of UNIX.
- Initially only datatype was word. Then offered 4 datatypes char, int , float, double.
- C is based heavily around the idea of an expression and allow easy conversion between data types.
- It is regarded a “mid level programming language”.

C++

- C++ was developed by Bjarne Stroustrup of AT&T Bells lab in the beginning in 1980 as “C with classes”.
- Standard was developed around 1998 but no compiler was yet formed that follow the standards completely.

Prolog

- Prolog is declarative language developed by Alain Colmerauer and Phillippe Roussel with Robert Kowalski in 1972 based in predicate calculus and mathematical logic.
- It became popular in 1980's because it was useful in expert system development. Initially developed for NLP.

```
mother(joanne, jake).
father(vern, joanne).

grandparent(X, Z) :- parent(X, Y), parent(Y, Z).
```