FORTRAN History

- The first high level language to become widely used.
- First developed by IBM for the IBM 704 computer in 1957.
- Emphesis on efficiency (for that machine).
- Static storage allocation and no recursion.
- Standards in 1966, 1977 and 1990.

FORTRAN

```
PROGRAM TRIVIAL
INTEGER I
I=2
IF(I .GE. 2) CALL PRINTIT
STOP
END
SUBROUTINE PRINTIT
PRINT *,'Hello World'
RETURN
END
```

FORTRAN 77

```
      PROGRAM MAIN
      PARAMETER (MAXSIZ=99)
      REAL A(MAXSIZ)
 10   READ (5,100,END=999) K
100   FORMAT(I5)
        IF (K.LE.0.OR. K.GT.MAXSIZ) STOP
        READ *,(A(I),I=1,K)
        PRINT *,(A(I),I=1,K)
        PRINT *,'SUM=',SUM(A,K)
        GOTO 10
999   PRINT *,'All Done'
      STOP
      END
```

FORTRAN 77

```
C   SUMMATION PROGRAM
        FUNCTION SUM(V,N)
        REAL :: V(N) ! New style declaration
        SUM = 0.0
        DO 20 I = 1,N
          SUM = SUM + V(I)
20        CONTINUE
        RETURN
        END
```

Features

- Static storage.
- No recursion (in FORTRAN 66), so no runtime stack needed.
- Separate compilation – extensive numerical libraries, e.g. NAG.
- Shared COMMON data areas.
- The EQUIVALENCE statement.
- Subroutine arguments passed by reference.
- Efficient compiled code.
- Extensively used for scientific work.
- Archaic syntax.

COBOL History

- COBOL (COmmon Business Oriented Language) has been widely used since the early 1960s.
- First version in 1960, revisions in 1974 and 1984.
- Uses nouns and verbs to describe actions.
- Complete separation of data descriptions from commands.
- Compilers used to be complex and slow.
- Programs somewhat easier to read than to write.
- Often blamed for the Y2K problem.

COBOL Example

```
IDENTIFICATION DIVISION
PROGRAM-ID. SUM-OF-PRICES.
AUTHOR. T-PRATT.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER. SUN.
OBJECT-COMPUTER. SUN.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT INP-DATA ASSIGN TO INPUT.
    SELECT RESULT-FILE ASSIGN TO OUTPUT.
DATA DIVISION.
FD INP-DATA LABEL RECORD IS OMITTED.
01 ITEM-PRICE.
    02 ITEM PICTURE X(30)
    02 PRICE PICTURE 9999V99.
WORKING-STORAGE SECTION.
77 TOT PICTURE 9999V99, VALUE 0, USAGE COMPUTATIONAL.
01 SUM-LINE.
    02 FILLER VALUE ' SUM =' PICTURE X(12).
    02 SUM-OUT PICTURE $$,$$$,$$9.99.
    02 COUNT-OUT PICTURE ZZZ9.
    ... More data
```

COBOL (Cont)

```
PROCEDURE DIVISION.
START.
    OPEN INPUT INP-DATA AND OUTPUT RESULT-FILE.
READ-DATA.
    READ INP-DATA AT END GO TO PRINT-LINE.
    ADD PRICE TO TOT.
    ADD 1 TO COUNT.
    MOVE PRICE TO PRICE-OUT.
    MOVE ITEM TO ITEM-OUT.
    WRITE RESULT-LINE FROM ITEM-LINE.
    GO TO READ-DATA.
PRINT-LINE.
    MOVE TOT TO SUM-TOT.
    ... More statements
    CLOSE INP-DATA AND RESULT-FILE.
    STOP RUN.
```

COBOL

- Data often held in a form that requires little conversion during input/output.
- Arithmetic operations simple and less important than data movement.
- Static data allocation and no recursion.
- Originally the only subroutine mechanism was the PERFORM statement which allowed specified paragraphs to be executed, possibly repeated.
- Such subroutines were parameterless.