BCA SEMESTER - II

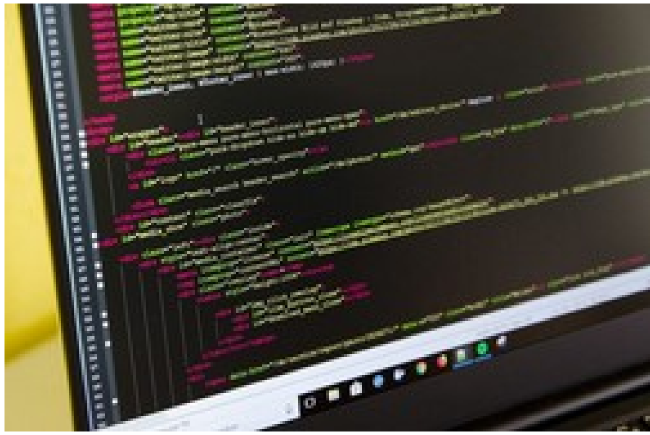
0302203
HISTORY OF COMPUTING

UNIT – 3
HISTORY OF PROGRAMMING LANGUAGES

- Dr. Disha Shah

Index

- **History of Programming Languages**
  - Ada Lovelace’s machine algorithm
  - Machine Language
  - Symbolic Programming Language
  - Lower Level Languages
  - Higher Level Languages
  - FORTRAN
  - ALGOL (Algorithmic Language)
  - LISP (List Processor)
  - COBOL (Common Business Oriented Language)



Index

- **Mid 1900s (Any 3)**
  - BASIC
  - PASCAL
  - Smalltalk,
  - C,
  - PROLOG
  - Ada
  - C++
  - Python
  - Ruby
  - Java,
  - PHP
  - Java Script
- **Mid 2000s (Any 3)**
  - Scala
  - Go
  - Dart
  - Swift
  - AlphaGo
  - Rust
  - Kotlin
  - Flutter
  - NLP

Introduction

- Computer programming is essential in our world today, running the systems for almost every device we use.
- Computer programming languages allow us to tell machines what to do.
- Machines and humans "think" very differently, so programming languages are necessary to bridge that gap.

Introduction

- The first computer programming language was created in 1883, when a woman named Ada Lovelace worked with Charles Babbage on his very early mechanical computer, the Analytical Engine.
- While Babbage was concerned with simply computing numbers, Lovelace saw that the numbers the computer worked with could represent something other than just amounts of things.
- She wrote an algorithm for the Analytical Engine that was the first of its kind.
- Because of her contribution, Lovelace is credited with creating the first computer programming language.
- As different needs have arisen and new devices have been created, many more languages have followed.

Ada Lovelace’s machine algorithm

- Ada Lovelace: The First Computer Programmer

Ada Lovelace's machine algorithm

- Ada Lovelace invents the first-ever machine algorithm for Charles Babbage's Difference Machine to compute Bernoulli number that lays the foundation for all programming languages.
- Her work was a formula showing how to configure the Engine to calculate a particular complex sequence of numbers, called Bernoulli numbers.
- The formula is now widely recognized as the first computer algorithm in history.

Machine Language

- Sometimes referred to as machine code or object code, machine language is a collection of binary digits or bits that the computer reads and interprets.
- Machine language is the only language a computer is capable of understanding.
- Below is an example of machine language (binary) for the text "Hello World."

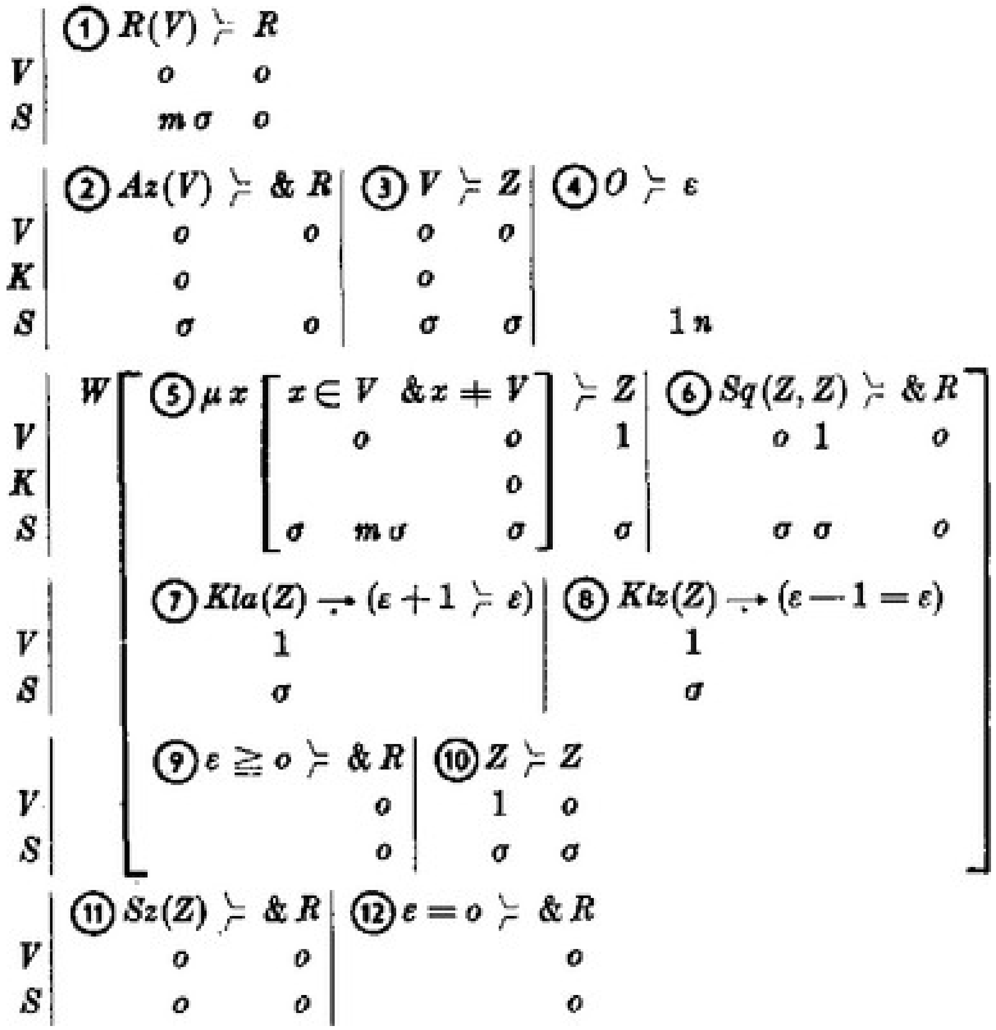
  01001000 01100101 01101100 01101100 01101111 00100000 01010111 01101111 01110010 01101100 01100100

Plankalkül

- Somewhere between 1944-45, Konrad Zuse developed the first ‘real’ programming language called Plankalkül (Plan Calculus) for the Z3.
- Zeus’s language (among other things) allowed for the creations of procedures, which stored chunks of code that could be invoked over and over to perform routine operations.

Plankalkül

Fig. 3.



$$\begin{array}{l|llll}
 & ① & R(V) & \succeq & R \\
V & & 0 & & 0 \\
S & & m\sigma & & 0
\end{array}$$

$$\begin{array}{l|llll|lll|lll}
 & ② & Az(V) & \succeq & \& R & ③ \; V & \succeq & Z & ④ \; 0 & \succeq & \varepsilon \\
V & & 0 & & 0 & 0 & & 0 & & & \\
K & & 0 & & & 0 & & & & & \\
S & & \sigma & & 0 & \sigma & & \sigma & & & 1n
\end{array}$$

$$\begin{array}{l|l}
 & W \left[\begin{array}{l}
\begin{array}{lll|l}
⑤ \; \mu x \left[\begin{array}{ll} x \in V & \& x \neq V \\ \quad 0 & \quad 0 \\ & \quad 0 \\ \sigma \quad m\sigma & \quad \sigma \end{array} \right] & \succeq & \begin{array}{l} Z \\ 1 \\ \\ \sigma \end{array} & \begin{array}{llll} ⑥ \; Sq(Z, Z) & \succeq & \& R \\ \quad 0 \;\; 1 & & 0 \\ & & \\ \quad \sigma \;\; \sigma & & 0 \end{array}
\end{array} \\
\begin{array}{l|l}
⑦ \; Kla(Z) \xrightarrow{\cdot} (\varepsilon + 1 \succeq \varepsilon) & ⑧ \; Klz(Z) \xrightarrow{\cdot} (\varepsilon - 1 = \varepsilon) \\
\quad 1 & \quad 1 \\
\quad \sigma & \quad \sigma
\end{array} \\
\begin{array}{llll|lll}
⑨ \; \varepsilon \geqq 0 & \succeq & \& R & & ⑩ \; Z & \succeq & Z \\
 & & 0 & & 1 & & 0 \\
 & & 0 & & \sigma & & \sigma
\end{array}
\end{array} \right] \\
V & \\
K & \\
S & \\
V & \\
S & \\
V & \\
S &
\end{array}$$

$$\begin{array}{l|llll|llll}
 & ⑪ & Sz(Z) & \succeq & \& R & ⑫ & \varepsilon = 0 & \succeq & \& R \\
V & & 0 & & 0 & & & & 0 \\
S & & 0 & & 0 & & & & 0
\end{array}$$

Symbolic Programming Language

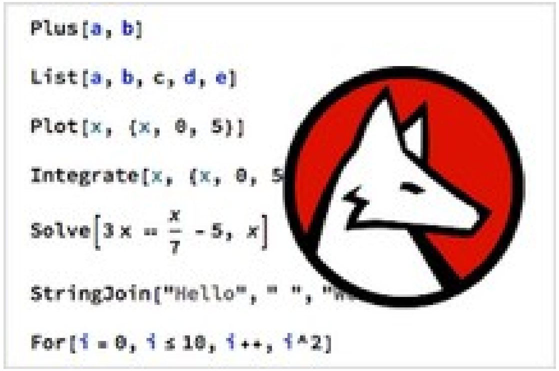
- In computer programming, symbolic programming is a programming paradigm in which the program can manipulate its own formulas and program components as if they were plain data.
- Languages that support symbolic programming are Wolfram Language, LISP and Prolog.
- All modern programming languages are symbolic languages.

Symbolic Programming Language

- Through symbolic programming, complex processes can be developed that build other more intricate processes by combining smaller units of logic or functionality.
- Thus, such programs can effectively modify themselves and appear to "learn", which makes them better suited for applications such as artificial intelligence, expert systems, natural language processing, and computer games.

Wolfram Language

- The Wolfram Language is a general multi-paradigm programming language developed by Wolfram Research.
- It emphasizes symbolic computation, functional programming, and rule-based programming and can employ arbitrary structures and data.



```
Plus[a, b]

List[a, b, c, d, e]

Plot[x, {x, 0, 5}]

Integrate[x, {x, 0, 5

Solve[3 x == x/7 - 5, x]

StringJoin["Hello", " ", "

For[i = 0, i ≤ 10, i++, i^2]
```

Lower Level Languages

- A low-level language, often known as a computer's native language, is a sort of programming language.
- It is very close to writing actual machine instructions, and it deals with a computer's hardware components and constraints.
- It works to control a computer's operational semantics and provides little or no abstraction of programming ideas.
- In contrast to high-level language that used for developing software, low-level code is not human-readable, and it is often cryptic.
- Assembly language and machine language are two examples of low-level programming languages.

Lower Level Languages

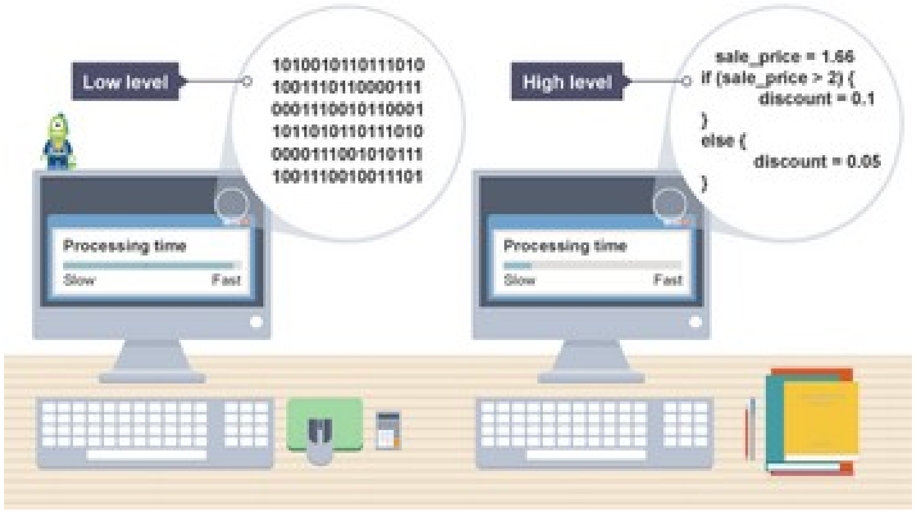
- The languages that come under this category are the Machine level language and Assembly language.

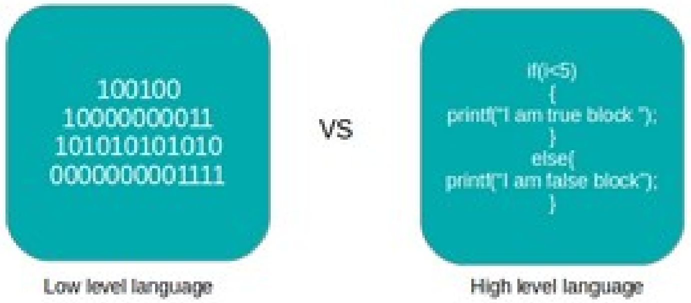
High Level Languages

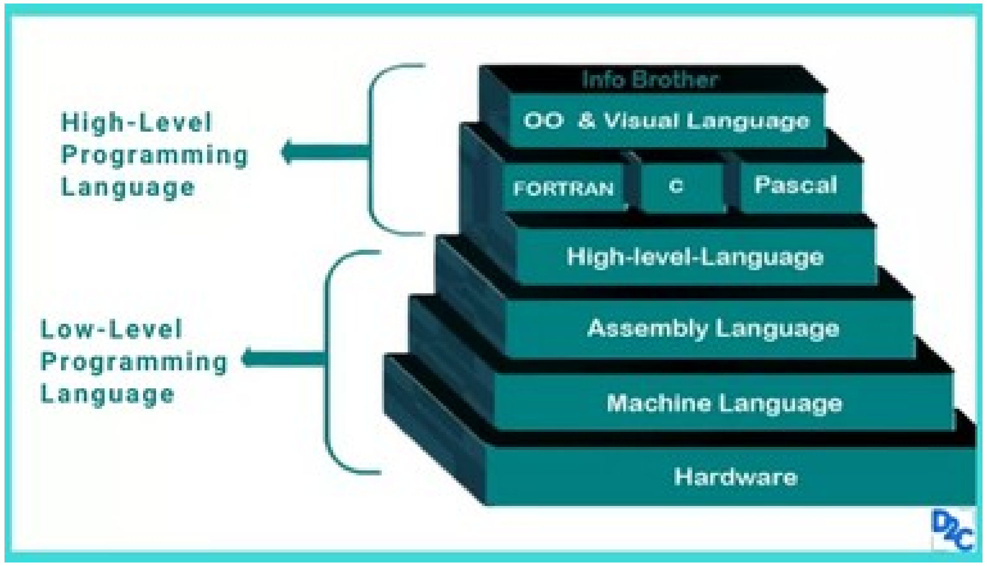
- A high-level language (HLL) is a programming language such as C, FORTRAN, or Pascal that enables a programmer to write programs that are more or less independent of a particular type of computer.
- Such languages are considered high-level because they are closer to human languages and further from machine languages.
- In contrast, assembly languages are considered low-level because they are very close to machine languages.

High Level Languages

- The first high-level programming languages were designed in the 1950s.
- Now there are dozens of different languages, including Ada, Algol, BASIC, COBOL, C, C++, FORTRAN, LISP, Pascal, and Prolog.


Low level
1010010110111010
1001110110000111
0001110010110001
1011010110111010
0000111001010111
1001110010011101
Processing time
Slow
Fast
High level
sale_price = 1.66
if (sale_price > 2) {
discount = 0.1
}
else {
discount = 0.05
}
Processing time
Slow
Fast


100100
10000000011
101010101010
0000000001111
Low level language
VS
if(i<5)
{
printf("I am true block ");
}
else{
printf("I am false block");
}
High level language


High-Level Programming Language
Low-Level Programming Language
Info Brother
OO & Visual Language
FORTRAN
C
Pascal
High-level-Language
Assembly Language
Machine Language
Hardware

FORTRAN

- Fortran was originally developed by a team at IBM in 1957 for scientific calculations.
- Fortran, as derived from Formula Translating System, is a general-purpose, imperative programming language. It is used for numeric and scientific computing.

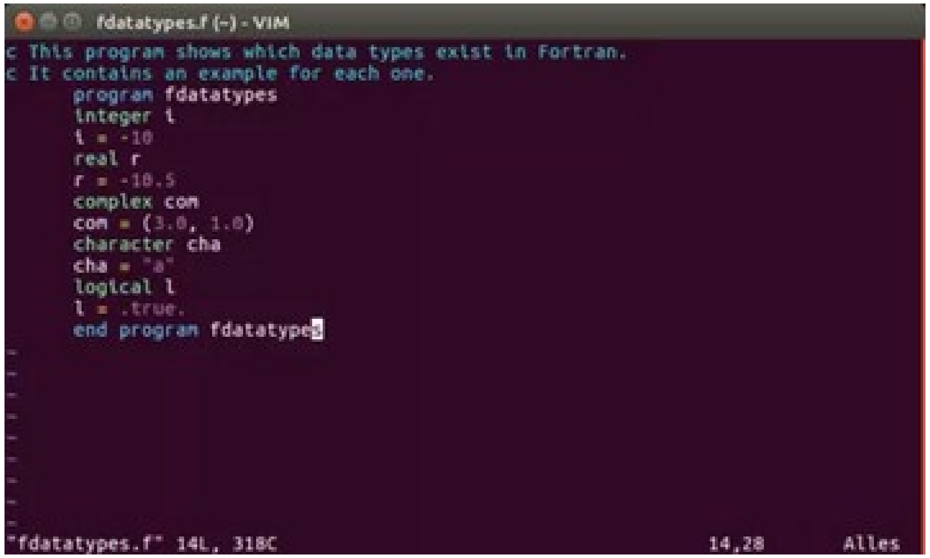
  It supports −

  - Numerical analysis and scientific computation
  - Structured programming
  - Array programming
  - Modular programming
  - Generic programming
  - High performance computing on supercomputers
  - Object oriented programming
  - Concurrent programming
  - Reasonable degree of portability between computer systems

FORTRAN

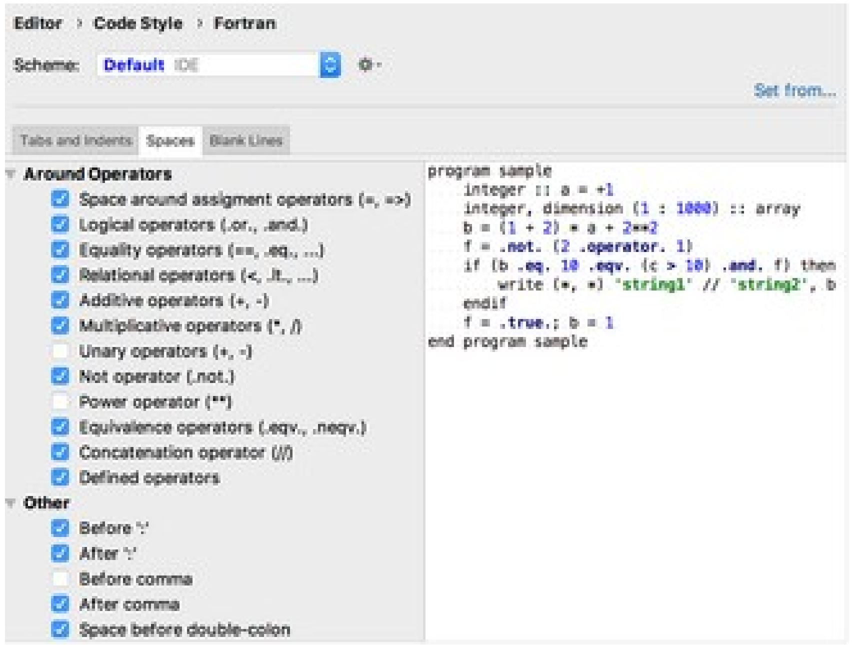
- Original versions, Fortran I, II and III are considered obsolete now.
- Oldest version still in use is Fortran IV, and Fortran 66.
- Most commonly used versions today are : Fortran 77, Fortran 90, and Fortran 95.
- Fortran 77 added strings as a distinct type.
- Fortran 90 added various sorts of threading, and direct array processing.

FORTRAN



fdatatypes.f (~) - VIM

```
c This program shows which data types exist in Fortran.
c It contains an example for each one.
      program fdatatypes
      integer i
      i = -10
      real r
      r = -10.5
      complex com
      com = (3.0, 1.0)
      character cha
      cha = "a"
      logical l
      l = .true.
      end program fdatatypes
~
~
~
~
~
~
~
~
~
"fdatatypes.f" 14L, 318C                             14,28         Alles
```



Editor › Code Style › Fortran

Scheme: Default IDE

Set from...

Tabs and Indents | Spaces | Blank Lines

Around Operators
- [✓] Space around assigment operators (=, =>)
- [✓] Logical operators (.or., .and.)
- [✓] Equality operators (==, .eq., ...)
- [✓] Relational operators (<, .lt., ...)
- [✓] Additive operators (+, -)
- [✓] Multiplicative operators (*, /)
- [ ] Unary operators (+, -)
- [✓] Not operator (.not.)
- [ ] Power operator (**)
- [✓] Equivalence operators (.eqv., .neqv.)
- [✓] Concatenation operator (//)
- [✓] Defined operators

Other
- [✓] Before ':'
- [✓] After ':'
- [ ] Before comma
- [✓] After comma
- [✓] Space before double-colon

```
program sample
    integer :: a = +1
    integer, dimension (1 : 1000) :: array
    b = (1 + 2) * a + 2**2
    f = .not. (2 .operator. 1)
    if (b .eq. 10 .eqv. (c > 10) .and. f) then
        write (*, *) 'string1' // 'string2', b
    endif
    f = .true.; b = 1
end program sample
```

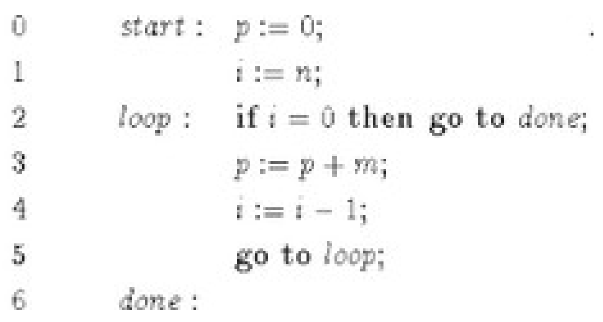
ALGOL

- ALGOL (ALGOrithmic Language) is one of several high level languages designed specifically for programming scientific computations.
- It started out in the late 1950's, first formalized in a report titled ALGOL 58, and then progressed through reports ALGOL 60, and ALGOL 68.
- It was designed by an international committee to be a universal language.
- Their original conference, which took place in Zurich, was one of the first formal attempts to address the issue of software portability.

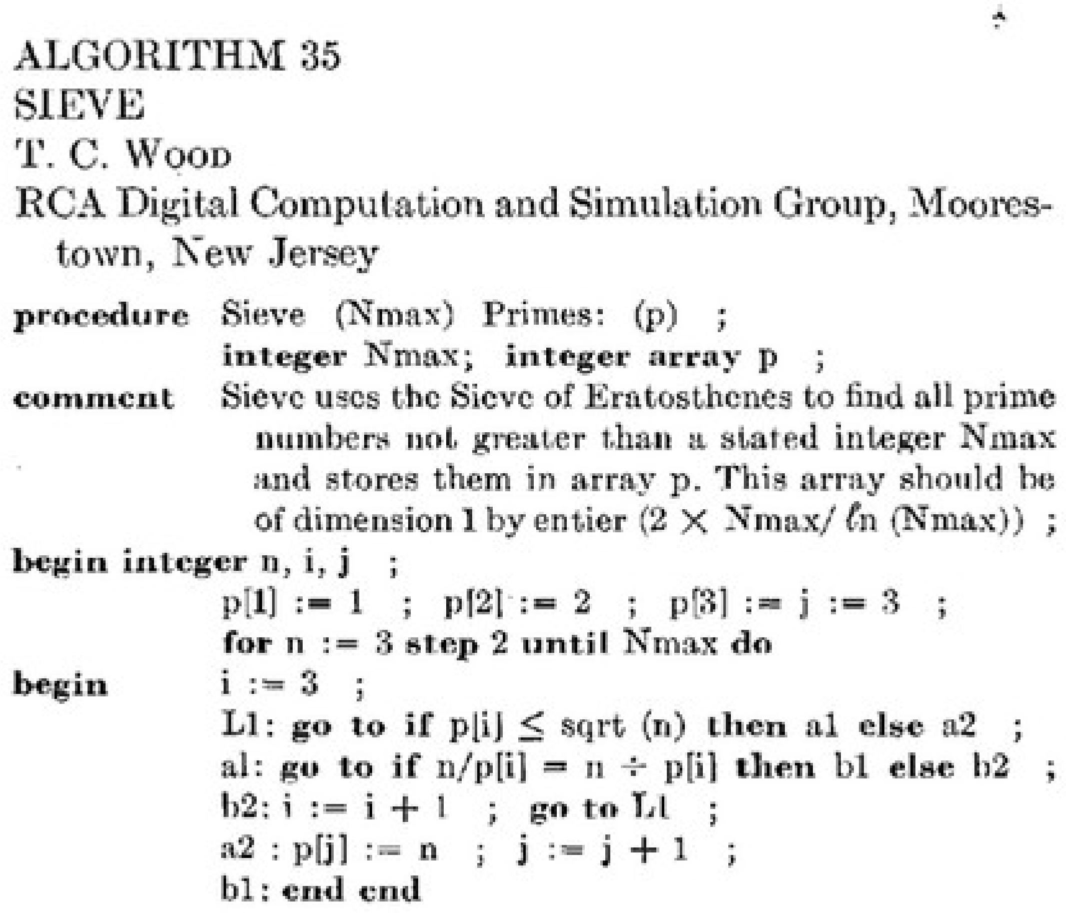
ALGOL

- ALGOL's machine independence permitted the designers to be more creative, but it made implementation much more difficult.
- Although ALGOL never reached the level of commercial popularity of FORTRAN and COBOL, it is considered the most important language of its era in terms of its influence on later language development.
- ALGOL’s lexical and syntactic structures became so popular that virtually all languages designed since have been referred to as "ALGOL - like"; that is they have been hierarchical in structure with nesting of both environments and control structures.

ALGOL



```
start :  p := 0;
         i := n;
loop :   if i = 0 then go to done;
         p := p + m;
         i := i − 1;
         go to loop;
done :
```



ALGORITHM 35
SIEVE
T. C. Wood
RCA Digital Computation and Simulation Group, Moorestown, New Jersey

```
procedure  Sieve  (Nmax)  Primes:  (p)   ;
           integer Nmax;  integer array p   ;
comment    Sieve uses the Sieve of Eratosthenes to find all prime
             numbers not greater than a stated integer Nmax
             and stores them in array p. This array should be
             of dimension 1 by entier (2 × Nmax/ ℓn (Nmax))  ;
begin integer n, i, j   ;
           p[1] := 1   ;  p[2] := 2   ;  p[3] := j := 3   ;
           for n := 3 step 2 until Nmax do
begin      i := 3   ;
           L1: go to if p[i] ≤ sqrt (n) then a1 else a2   ;
           a1: go to if n/p[i] = n ÷ p[i] then b1 else b2   ;
           b2: i := i + 1   ;  go to L1   ;
           a2 : p[j] := n   ;  j := j + 1   ;
           b1: end end
```

LISP

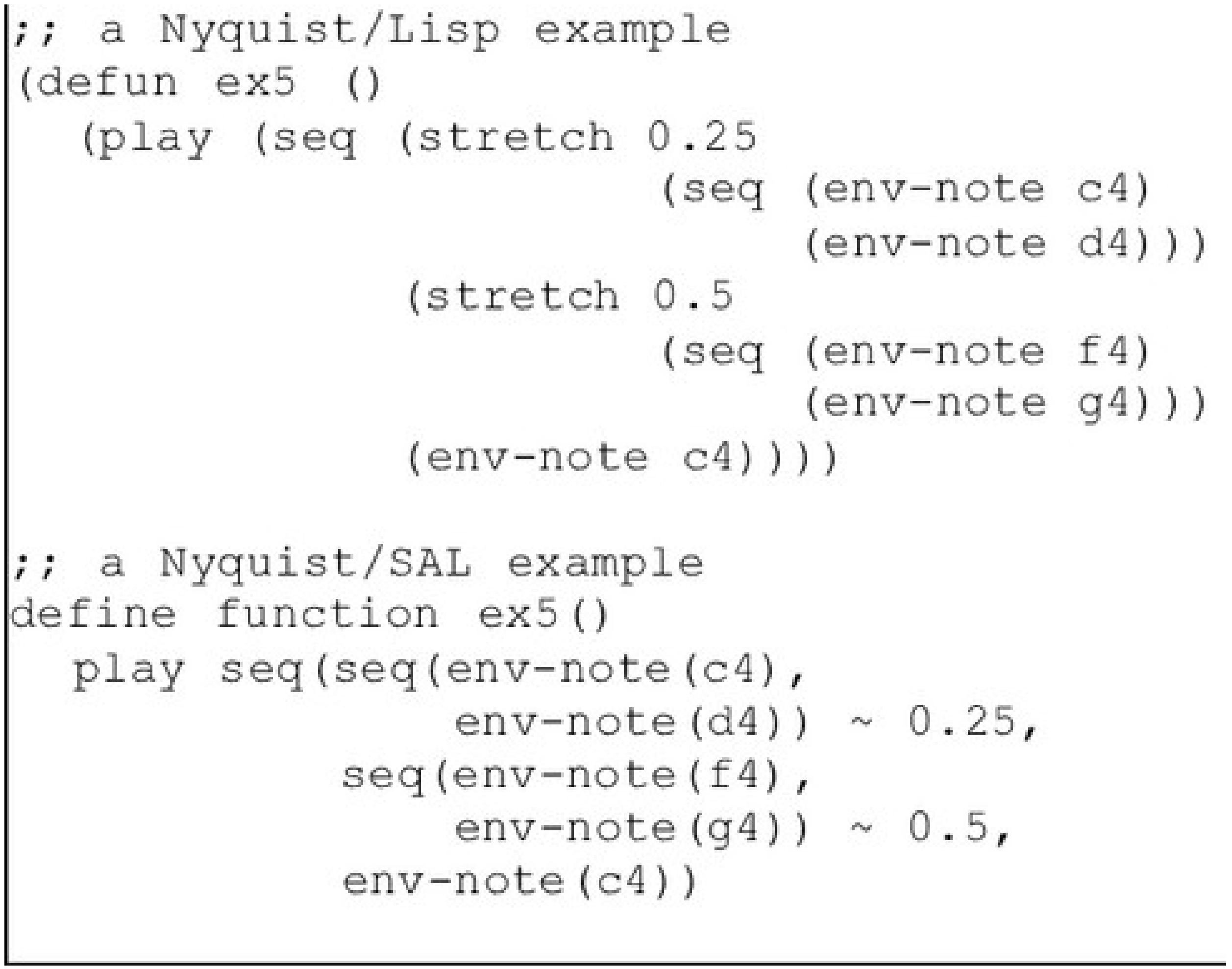
- John McCarthy invented LISP in 1958, shortly after the development of FORTRAN.
- It was first implemented by Steve Russell on an IBM 704 computer.
- It is particularly suitable for Artificial Intelligence programs, as it processes symbolic information effectively.
- Common Lisp originated, during the 1980s and 1990s, in an attempt to unify the work of several implementation groups that were successors to Maclisp, like ZetaLisp and NIL (New Implementation of Lisp) etc.
- It serves as a common language, which can be easily extended for specific implementation.
- Programs written in Common LISP do not depend on machine-specific characteristics, such as word length etc.

LISP

Features of Common LISP

- It is machine-independent
- It uses iterative design methodology, and easy extensibility.
- It allows updating the programs dynamically.
- It provides high level debugging.
- It provides advanced object-oriented programming.
- It provides a convenient macro system.
- It provides wide-ranging data types like, objects, structures, lists, vectors, adjustable arrays, hash-tables, and symbols.
- It is expression-based.
- It provides an object-oriented condition system.
- It provides a complete I/O library.
- It provides extensive control structures

LISP



```
;; a Nyquist/Lisp example
(defun ex5 ()
  (play (seq (stretch 0.25
                      (seq (env-note c4)
                           (env-note d4)))
             (stretch 0.5
                      (seq (env-note f4)
                           (env-note g4)))
             (env-note c4))))

;; a Nyquist/SAL example
define function ex5()
  play seq(seq(env-note(c4),
               env-note(d4)) ~ 0.25,
           seq(env-note(f4),
               env-note(g4)) ~ 0.5,
           env-note(c4))
```

COBOL

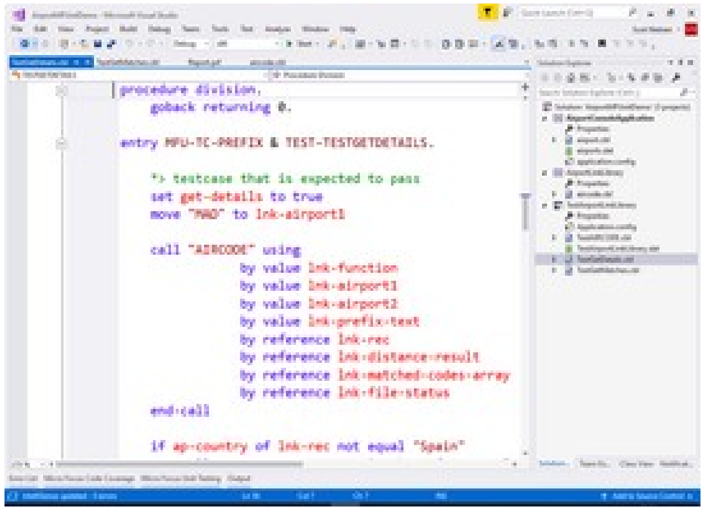
- COBOL(COmmon Business-Oriented Language) is a high-level language.
- It is an imperative, procedural and, since 2002, object-oriented language.
- COBOL is primarily used in business, finance, and administrative systems for companies and governments.
- COBOL is still widely used in applications deployed on mainframe computers, such as large-scale batch and transaction processing jobs.
- COBOL was designed in 1959 by CODASYL and was partly based on the programming language FLOW-MATIC designed by Grace Hopper.

COBOL

Importance of COBOL

- COBOL was the first widely used high-level programming language. It is an English-like language which is user friendly. All the instructions can be coded in simple English words.
- COBOL is also used as a self-documenting language.
- COBOL can handle huge data processing.
- COBOL is compatible with its previous versions.
- COBOL has effective error messages and so, resolution of bugs is easier

COBOL



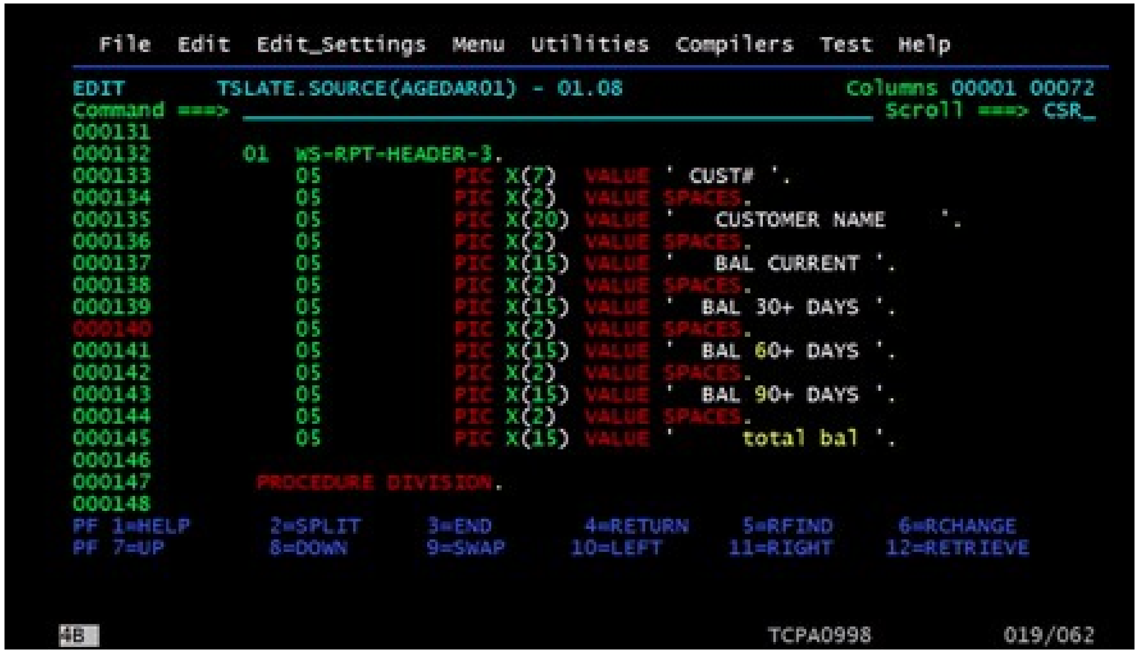
```
procedure division.
    goback returning 0.

entry MFU-TC-PREFIX & TEST-TESTGETDETAILS.

    *> testcase that is expected to pass
    set get-details to true
    move "MAD" to lnk-airport1

    call "AIRCODE" using
                by value lnk-function
                by value lnk-airport1
                by value lnk-airport2
                by value lnk-prefix-text
                by reference lnk-rec
                by reference lnk-distance-result
                by reference lnk-matched-codes-array
                by reference lnk-file-status
    end-call

    if ap-country of lnk-rec not equal "Spain"
```



```
  File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help

EDIT       TSLATE.SOURCE(AGEDAR01) - 01.08             Columns 00001 00072
Command ===>                                               Scroll ===> CSR_

             01  WS-RPT-HEADER-3.
                 05           PIC X(7)  VALUE ' CUST# '.
                 05           PIC X(2)  VALUE SPACES.
                 05           PIC X(20) VALUE '   CUSTOMER NAME    '.
                 05           PIC X(2)  VALUE SPACES.
                 05           PIC X(15) VALUE '   BAL CURRENT '.
                 05           PIC X(2)  VALUE SPACES.
                 05           PIC X(15) VALUE '  BAL 30+ DAYS '.
                 05           PIC X(2)  VALUE SPACES.
                 05           PIC X(15) VALUE '  BAL 60+ DAYS '.
                 05           PIC X(2)  VALUE SPACES.
                 05           PIC X(15) VALUE '  BAL 90+ DAYS '.
                 05           PIC X(2)  VALUE SPACES.
                 05           PIC X(15) VALUE '     total bal '.


              PROCEDURE DIVISION.

PF 1=HELP      2=SPLIT     3=END       4=RETURN    5=RFIND     6=RCHANGE
PF 7=UP        8=DOWN      9=SWAP     10=LEFT     11=RIGHT    12=RETRIEVE


    4B                                                 TCPA0998      019/062
```