

Lecture 2

A Historical View

Hardware Programming

Short Code (John Mauchly, 1949) for UNIVAC I; two-six-bit byte instructions; implemented as a pure interpreter

FORTRAN

Previous all floating point calculations implemented in software. IBM 704 developed which included floating point hardware. FORTRAN (FORMula TRANlator) developed as the first compiled language.

FORTRAN I (1956) Writing compiler an expensive process.
FORTRAN II (1958)
FORTRAN 77
FORTRAN 90

FORTRAN Evaluation: Long use; heavy investment in FORTRAN software; extremely efficient compilers; large mathematical libraries

Example (Sebesta)

```
C FORTRAN 90 EXAMPLE PROGRAM
C INPUT: AN INTEGER, LIST_LEN
C   WHERE LIST_LEN IS LESS THAN
C   100, FOLLOWED BY LIST_LEN-INTEGERS
C   VALUES
C OUTPUT: THE NUMBER OF INPUT VALUES
C   THAT ARE GREATER THAN THE AVERAGE
C   OF ALL INPUT VALUES
INTEGER INTLIST(99)
INTEGER LIST_LEN, COUNTER, SUM, AVERAGE, RESULT
RESULT = 0
SUM = 0
READ *, LIST_LEN
IF ((LIST_LEN .GT. 0) .AND. (LIST_LEN .LT. 100)) THEN
C READ INPUT DATA INTO AN ARRAY AND COMPUTE ITS SUM
  DO 10 COUNTER = 1, LIST_LEN
    READ *, INTLIST(COUNTER)
    SUM = SUM + INTLIST(COUNTER)
10  CONTINUE
C COMPUTE THE AVERAGE
  DO 20 COUNTER = 1, LIST_LEN
    IF ( INTLIST(COUNTER) .GT. AVERAGE) THEN
```

```

                RESULT = RESULT + 1
            END IF
20    CONTINUE
C PRINT THE RESULT
    PRINT *, 'NUMBER OF VALUES > AVERAGE IS: ', RESULT
ELSE
    PRINT *, 'ERROR-LIST LENGTH VALUE IS NOT LEGAL'
END IF
STOP
END

```

Functional Programming

LISP (John McCarthy, 1958) MIT: LIS^t Processor

Need for an artificial intelligence language that would process symbolic language in linked lists - and to handle recursion.

Scheme (more details later in the course) is a descendant of LISP developed at MIT in the mid-1970's.

ALGOL

Stemming out of a move to create a universal programming language and some fear of IBM dominants; ALGOL (ALGORithmic Language) was designed. It emphasized the concept of block structures allowing localization of code. It broadened the means of parameter passing methods. It was also easily defined using BNF (more later).

(From Sebesta)

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, averge, result;
    sum := 0;
    result :=0;
    readint (listlen);
    if (listlen > 0) ^ (listlen < 100) then
        begin
comment Read input into an array and computer the average;
            for counter := 1 step 1 until listlen do
                begin
                    readint (intlist[counter]);
                    sum := sum + intlist[counter];

```

```

        end;
comment Computer 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
end

```

COBOL

COBOL (COmmon Business Oriented Language) has been used more than any other programming language; yet, it has had little or no direct influence on the design of new languages. Designed in 1959 it was to assist in data manipulation. Program sections are divided into a DATA DIVISION to describe the data to be manipulated and a PROCEDURE DIVISION to manipulate the data.

BASIC

Designed (1964) at Dartmouth College BASIC (Beginners All-purpose Symbolic Instruction Code) was primarily designed as a language for learning programming. It was an interpreted language for years.

(From Sebesta)

```

REM QuickBASIC 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
REM greater 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 computer 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

```

PL/1

PL/1 (Programming Language 1, 1965, IBM) Designed as a combination of the best of FORTRAN and ALGOL. Programs can develop concurrently executing tasks, had extensive exception-handling, recursive procedures, pointers as a data type, and referencing of array cross-sections.

APL

APL (A Programming Language, 1975) Interpreted. Has very high level mathematical operators.

SNOBOL

SNOBOL ("Snowball", 1971) Was designed as language for text editing.

Pascal

Pascal (1971) was designed primarily by Niklaus Wirth. Introduced the case statement and records. It's greatest impact is on programming education.

(Partially from Sebesta)

{Pascal Example Program

Input: An integer, listlen, where listlen is less than 1000,
followed by listlen-integer values

Output: The number of input values that are greater than
the average of all input values }

program pasex (input, output);

type intlisttype = array [1..99] of integer;

var

```

        intlist : intlisttype;
        listlen, counter, sum, average, result : integer;
begin
    result := 0;
    sum := 0;
    readln( listlen );
    if ((listlen > 0) and (listlen < 100)) then
        begin
            for counter := 1 to listlen do
                begin
                    readln(intlist[counter]);
                    sum := sum + intlist[counter]
                end
            { Computer the average }
            average := sum div listlen;
            { Count the number of input values that are > average }
            for counter := 1 to listlen do
                if (intlist[counter] > average) then
                    result := result + 1;
            { Print result }
            writeln('The number of values > average is ', result);
        end
    else
        writeln('Error-input list length is not legal')
end.

```

C

C (1971) was developed at Bell Labs as a systems programming language. Its characteristics come both from ALGOL and in-house developments.

(Partially from Sebesta)

/* C 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 input values */

```

void main() {
    int intlist[99], listlen, counter, sum, average, result;
    sum = 0;
    result = 0;
    scanf("%d", &listlen);
    if ((listlen > 0) && (listlen < 100)) {
        /* Read input into an array and compute the sum */
        for ( counter = 0; counter < listlen; counter++ ) {

```

```

        scanf("%d", &intlist[counter]);
        sum += intlist[counter];
    }
    /* Compute the average */
    average = sum / listlen;
    /* Count the input values that are > average */
    for ( counter = 0; conter < listlen; counter++ )
        if ( intlist[counter] > average)
            result++;
    /* Print result */
    printf("Number of values > average is: %d\n", result);
else
    printf("Error-input list length is not legal\n");
}

```

MODULA-2

Like Pascal, MODULA-2 was designed by Niklaus Wirth (1977). Its power derived from the natural support of abstract data types low-level, system programming features. It gained a great deal of popularity especially as a teaching language.

Prolog

Prolog (Programing logic) s a "logic programming language" that was somewhat popular among artificial intelligence researchers and practitioners.

Ada

Ada (after Ada Lovelace, daughter of Lord Byron and the first computer programmer) is the result of a length design process that was complete in 1983. Ada was the standard Department of Defense language for many years; the standardization of the language and the control over compilers meeting the standard exactly forced it to be probably the most portable language ever developed. It traces its design from Pascal although it also has many of the characteristics of PL/1. (more in class lectures)

```

--Ada 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
with TEXT_IO; use TEXT_IO;
procedure ADA_EX is

```

```

package INT_IO is new INTERGER_IO(INTEGER);
use INT_IO;
type INT+LIST+TYPE is array (1..99) of INTEGER;

```

```

INT_LIST : INT_LIST_TYPE;
LIST_LEN, SUM, AVERAGE, RESULT : INTEGER;
begin
    RESULT := 0;
    SUM := 0;
    GET(LIST_LEN);
    if (LIST_LEN > 0) and (LIST_LEN < 100) then
        -- Read input data into an array and computer the sum
        for COUNTER := 1 .. LIST_LEN loop
            GET(INT_LIST(COUNTER));
            SUM := SUM + INT_LIST(COUNTER);
        end loop;
        -- Compute the average
        AVERAGE := SUM div LIST_LEN;
        -- Count the number of values that are > average
        -- for COUNTER := 1 .. LIST_LEN loop
            if INT_LIST(COUNGER) > AVERAGE then
                RESULT := RESULT + 1;
            end if;
        end loop;
        -- Print result
        PUT("The number of values > average is : ");
        PUT(RESULT);
        NEW_LINE;
    else
        PUT_LINE("Error-input list length is not legal");
    end if;
end ADA_EX;

```

C++

C++ is an extension of C++ with many modern improvements plus a reasonably solid object-oriented implementation. Changes included parameter type checking, public and private access to class elements, constructors and destructors, inline functions, default parameters, operator overloading, parameter call by reference, and so forth. It was developed at Bell Labs by Bjarne Stroustrup. (1985)

```

// Standard C++ 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 input values
int main() {
    int listlen;
    cin >> listlen;
    if ((listlen > 0) && (listlen < 100)) {

```

```

-- Read input into an array and compute the sum
int sum = 0;
for (int counter = 0; counter < listlen; counter++) {
    cin >> intlist[counter];
    sum += intlist[counter];
}
-- Compute the average
int average = sum / listlen;
-- Count the input values that are > average
int result = 0;
for ( int counter = 0; conter < listlen; counter++)
    if ( intlist[counter] > average)
        result++;
-- Print result
cout << "Number of values > average is " << result << endl;
else
    cout << "Error-input list length is not legal" << endl;
return 0;
}

```

Java

Java (Sun Microsystems, 1995) was designed to have similar functionality as C++ but smaller and safer with an entire object-oriented environment.

(from Sebesta)

```

// Java 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 input values

import java.io.*;
class IntSort {
public static void main(String args[]) throws IOException {
    DataInputStream in = new DataInputStream(System.in);
    int listlen,
        counter,
        sum = 0;
        average,
        result = 0;
    int[] intlist = int[99];
    listlen = Integer.parseInt(in.readLine());
    if ((listlen > 0) && (listlen < 100)) {

```

```

        /* Read input into an array and sumputer the sum */
        for ( counter = 0; counter < listlen; counter++ ) {
            intlist[counter] = Integer.valueOf(in.readLine()).intValue();
            sum += intlist[counter];
        }
        /* Computer the average */
        average = sum / listlen;
        /* Count the input values that are > average */
        for ( counter = 0; counter < listlen; counter++ )
            if (intlist[counter] > average)
                result++;
        /* Print result */
        System.out.println( "\nNumber of values > average is: " + result );
    }
    else
        System.out.println("Error-input list length is not legal \n");
}
}

```

Graphical Languages

LabView
EONReality