# DESIGNING TOOLS

## & PROGRAMMING LANGUAGE

# COMPUTER

# Introduction

- **Design Tools:**

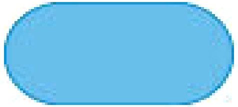
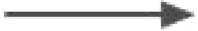
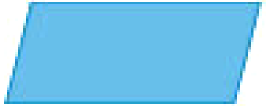
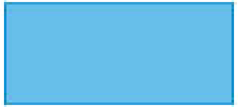
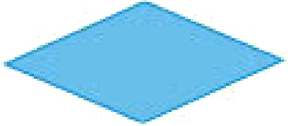
  Before any program coding, input, output, flow of data and logic should be defined.

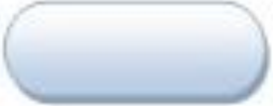
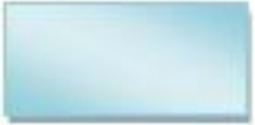
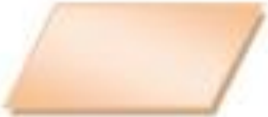
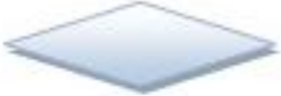
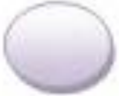
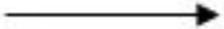
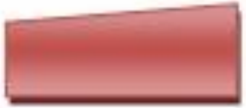
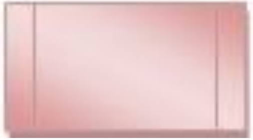
- For this purpose we need design tools.
- There are some design tools:

# Data Flow Diagram

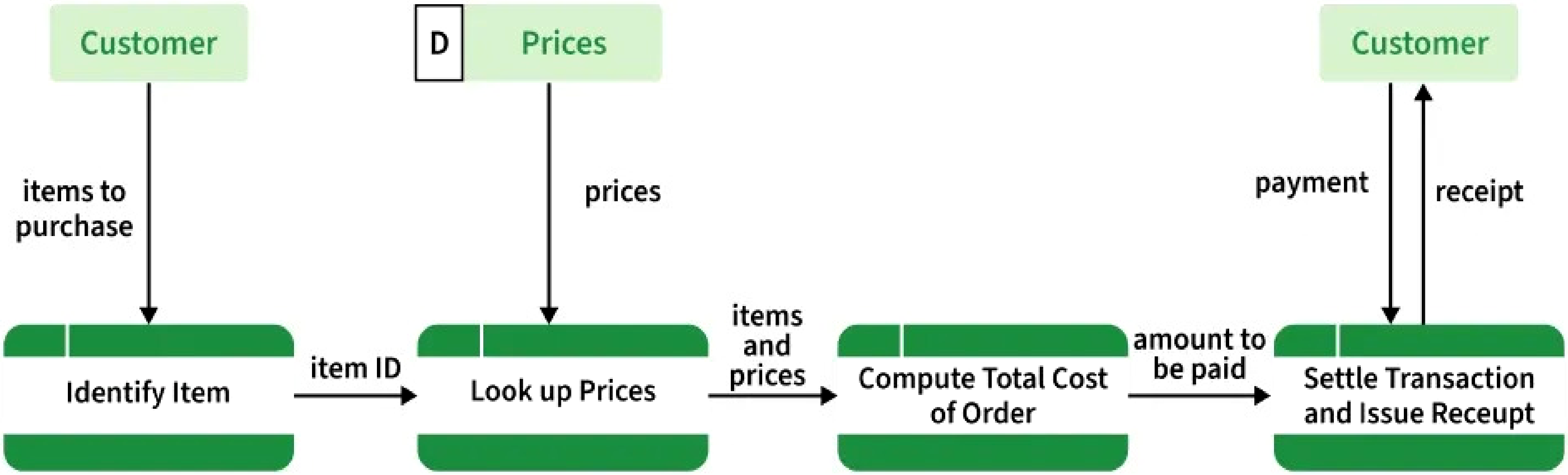
- **DFD (Data Flow Diagram):**

  Data flow diagrams are the most commonly used as a pictorial way of showing the flow of data through a system or subsystem.
- It is easier to understand and grasp.
- For diagrammatic representation it also uses symbols and notation.

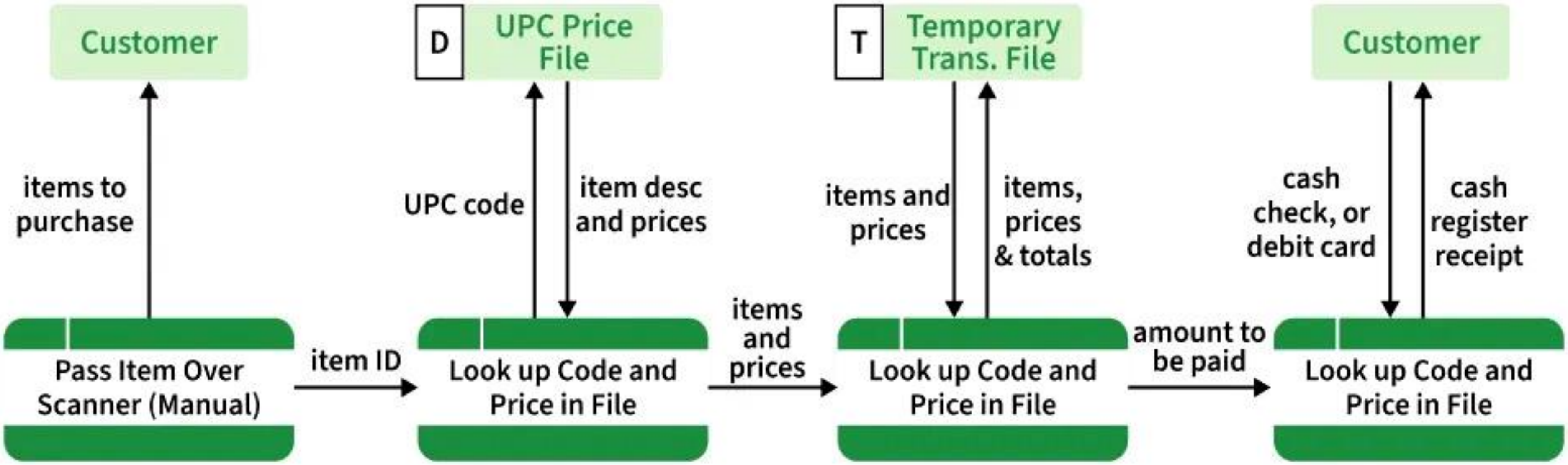
| Symbol | Name | Function |
|---|---|---|
|  | Start/end | An oval represents a start or end point |
|  | Arrows | A line is a connector that shows relationships between the representative shapes |
|  | Input/Output | A parallelogram represents input or output |
|  | Process | A rectangle represents a process |
|  | Decision | A diamond indicates a decision |

| Symbol | Meaning |
|---|---|
|  | Start/Stop |
|  | Process |
|  | Input/Output |
|  | Decision/Branching |
|  | Connector |
|  | Flow |
|  | Manual Input |
|  | Predefined Process |

# Logical Data Flow Diagram (DFD)



# Physical Data Flow Diagram (DFD)



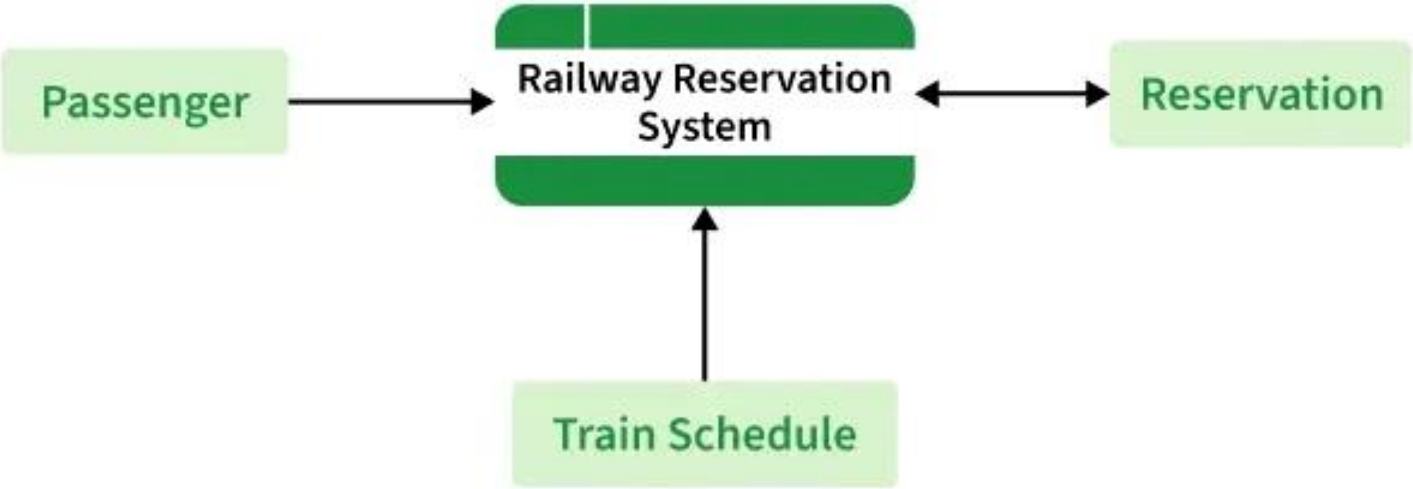
# Data Flow Diagram

- Data Flow:
  - It is represented by line arrow.
  - It shows the direction of flow of data.
- Process:
  - It changes the incoming data flow to outgoing data flow.
- Decision:
  - It shows the logical process which has resultant Yes or No.

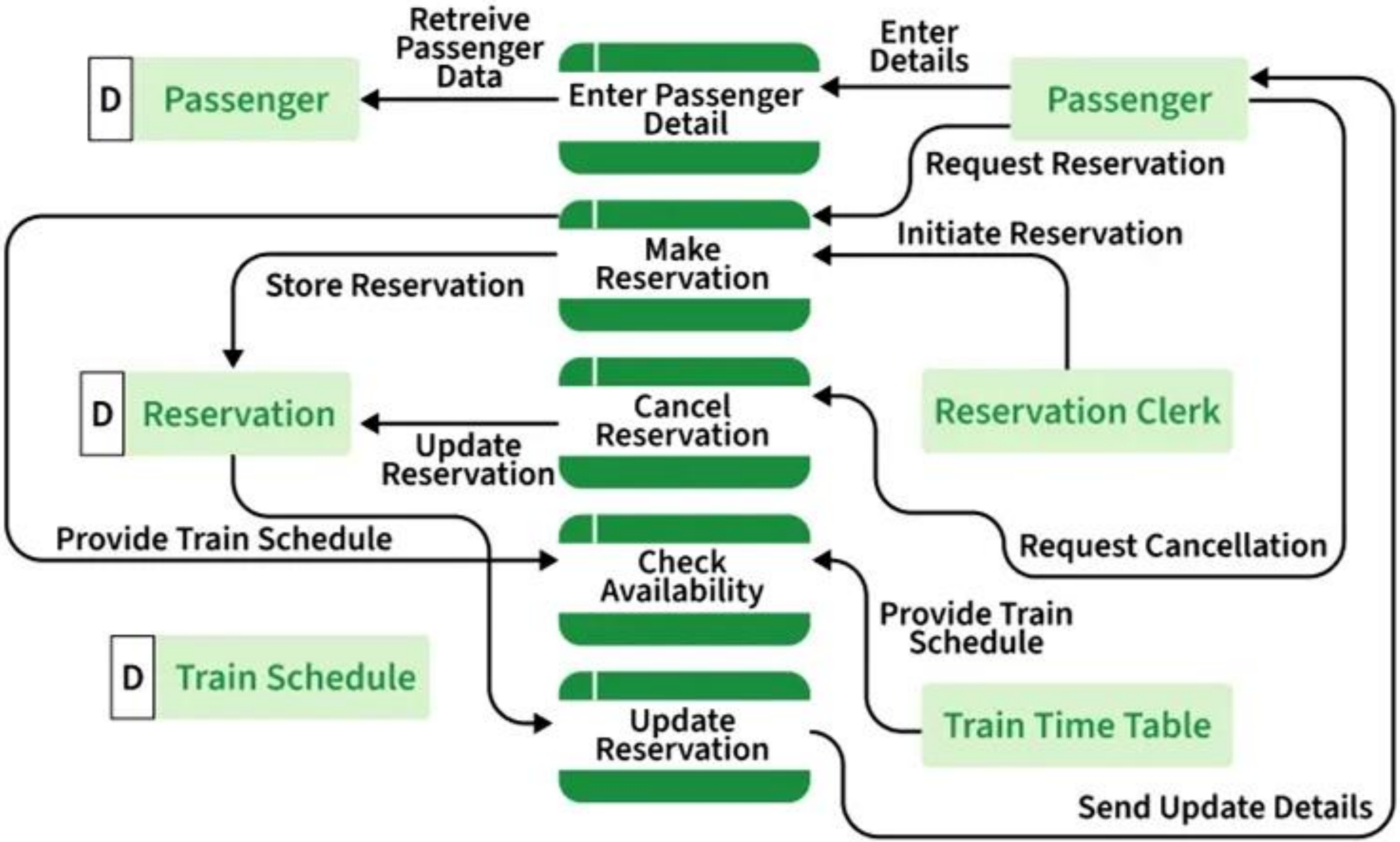
# Data Flow Diagram

- Connector:
  - It connects the flowcharts of more than one page.
- Input / Output:
  - It shows input/output in program.
- Data Store:
  - It shows storage of data.

# Level 0 Diagram of Railway Reservation System



# Level 1 DFD of Railway Reservation System



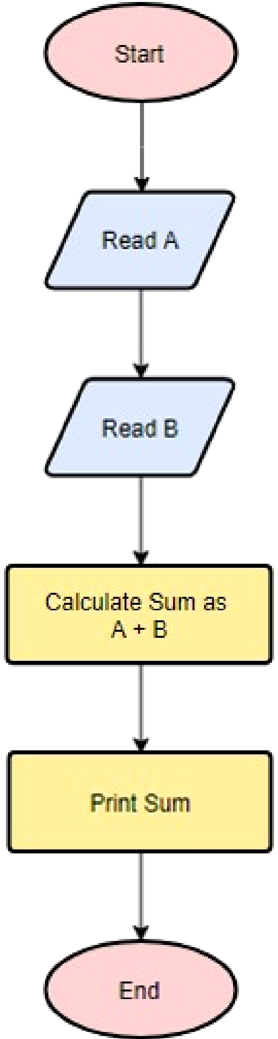
# Data Flow Diagram

- Algorithm:
  - To make a computer do something, we need to write a
  - computer program or group of instructions.
  - To write a computer program, we have to tell the computer, step by step, exactly what we want it to do.
  - The computer then executes the program, following each step to find the end goal.
  - When we are telling the computer what to do, we also get to choose how it is going to do it.

# Data Flow Diagram

- That's the point where computer algorithms come in.
- Algorithm is a set of instructions and basic techniques used to get a job done.
- An algorithm is an effective method for solving a problem using
- sequence of instructions.
- Each algorithm is a list of well defined instructions for completing a task from an initial state to the final state.
- A mistake in an algorithm that causes incorrect results is called a logical error.

Find the sum of 529 and 256



Start

Read A

Read B

Calculate Sum as A + B

Print Sum

End

Start

A = 529

B = 256

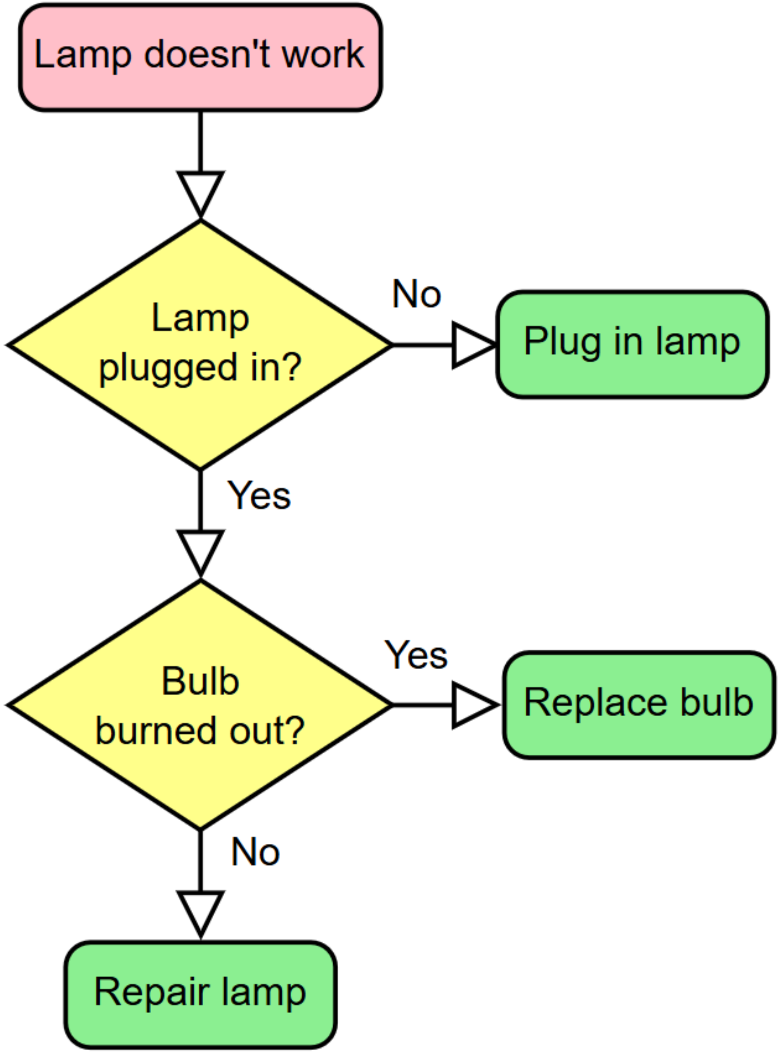
Sum = 529 + 256

Sum = 785

End

# Data Flow Diagram

- Flowchart:
  - The pictorial representation of a program or the algorithm is known as a flowchart.
  - It is nothing but a diagrammatic representation of the various steps involved in designing a system.
  - The purpose of using flowcharts is to graphically present the logical flow of data in the system and defining major phases of processing.



Lamp doesn't work

Lamp plugged in?

No

Plug in lamp

Yes

Bulb burned out?

Yes

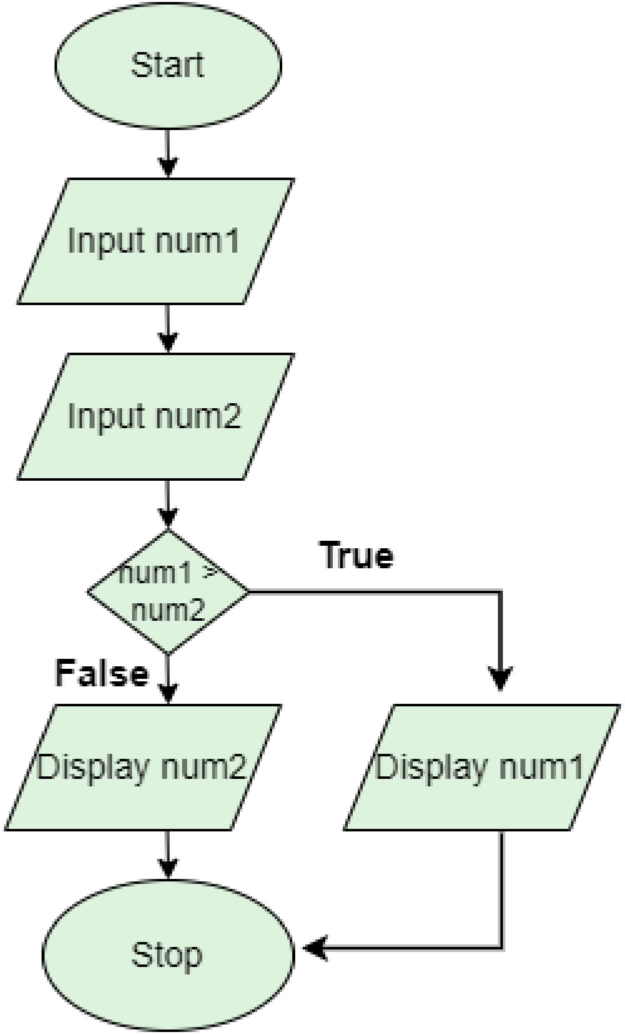
Replace bulb

No

Repair lamp

# Data Flow Diagram

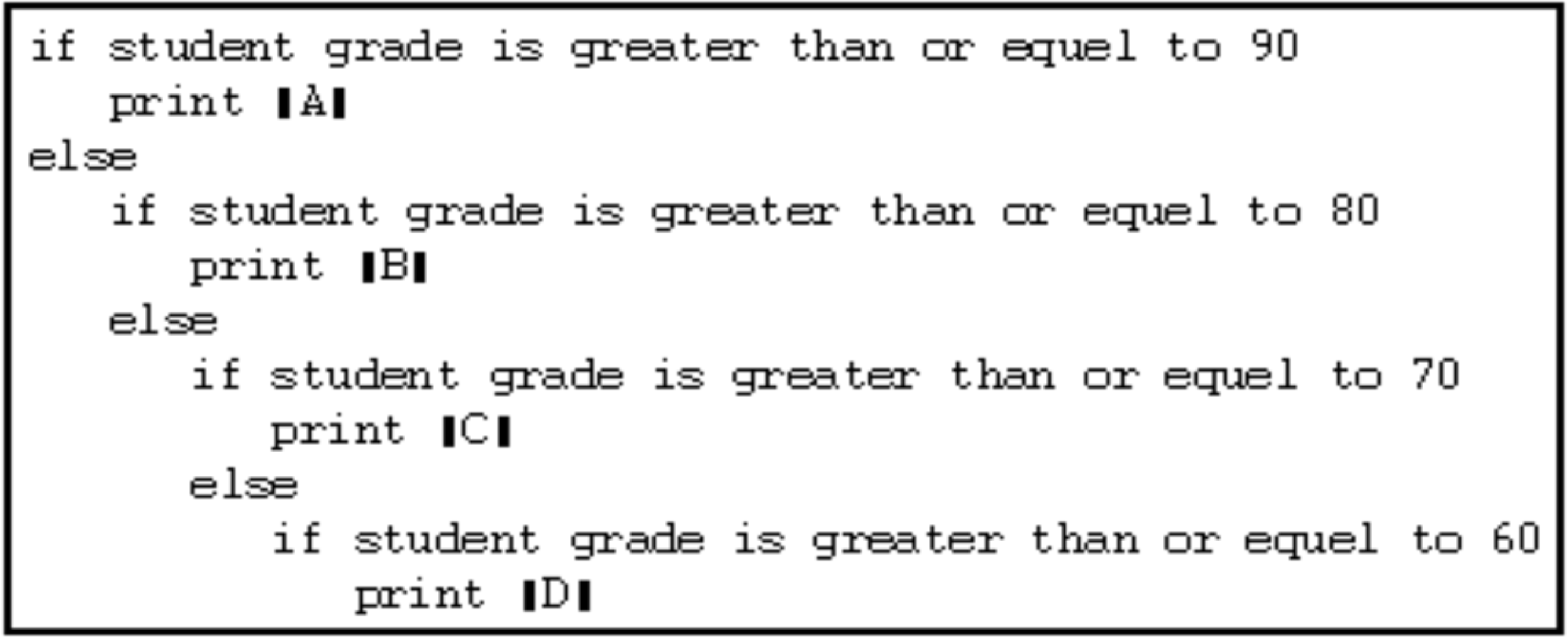
- For diagrammatic representation it also uses symbols and notations.
  - Start & Stop Decision box
  - Output & input
  - Connector Flow of data
  - Process


Start
Input num1
Input num2
num1 > num2
True
False
Display num2
Display num1
Stop

# Data Flow Diagram

- Pseudocode:
  - It is also called Program Design Language (PDL) and is an alternative to flowcharts.
  - Pseudocode allows the programmer to represent logic in English in like manner.
  - It is easy to modify, so many programmers prefer it.

# Example Pseudocode / C



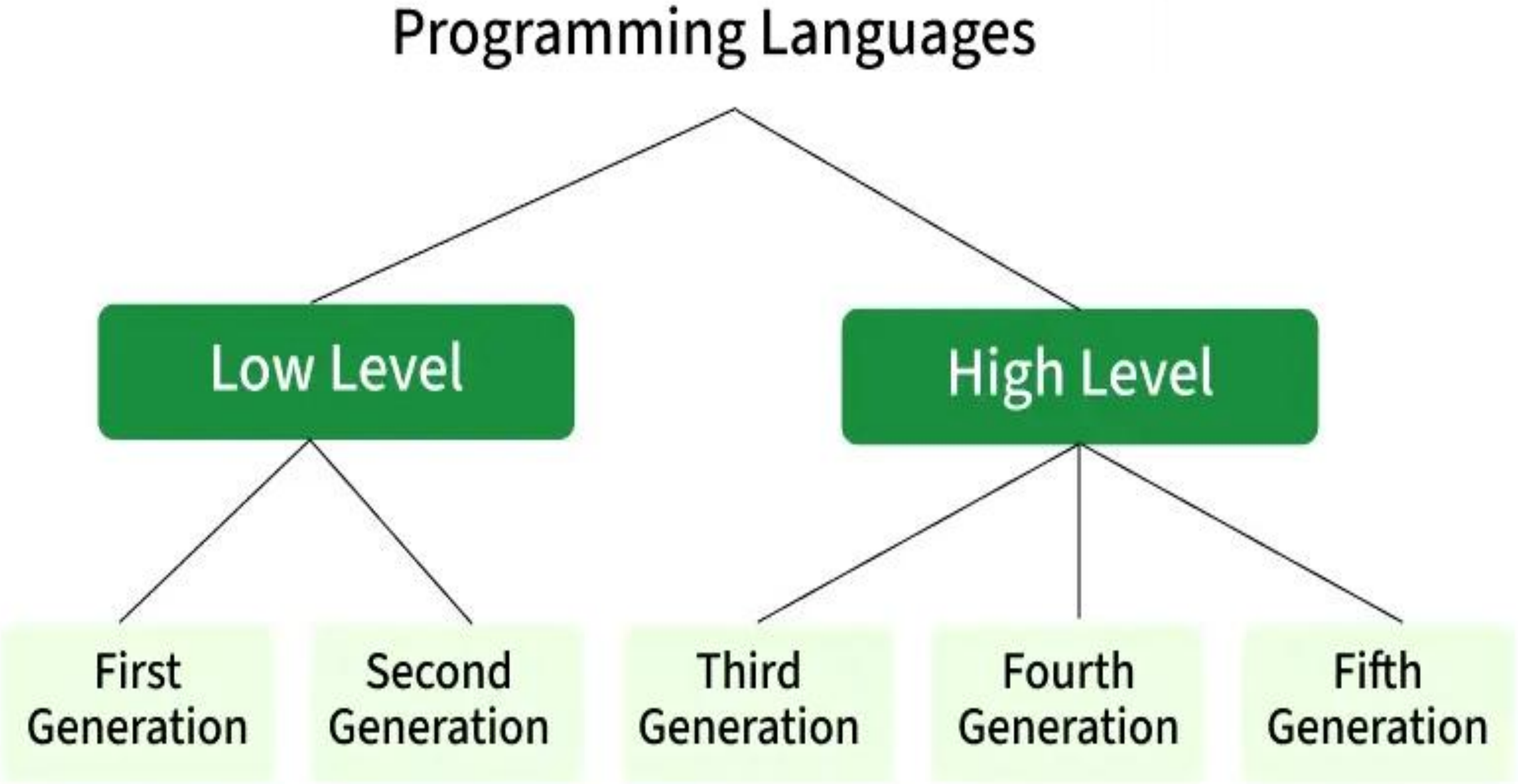
```
if student grade is greater than or equel to 90
    print ▌A▌
else
    if student grade is greater than or equel to 80
        print ▌B▌
    else
        if student grade is greater than or equel to 70
            print ▌C▌
        else
            if student grade is greater than or equel to 60
                print ▌D▌
```

# Programming Languages

- Programming Languages:
  - Programming Languages are the medium used by one to communicate instructions to a computer.
  - A programming language is an artificial language to express computation that can be performed by a computer.
  - It is a set of keywords, symbols, and a system of rules for constructing statements by which humans can communicate instructions to be executed by a computer.

# Programming Languages

- Each programming language has its own syntax that is the set of specific rules and words that express the logical steps of an algorithm.
- Programming languages are mainly of two types:
  - (a) Low level language
  - (b) High level language



Programming Languages

Low Level

High Level

First Generation

Second Generation

Third Generation

Fourth Generation

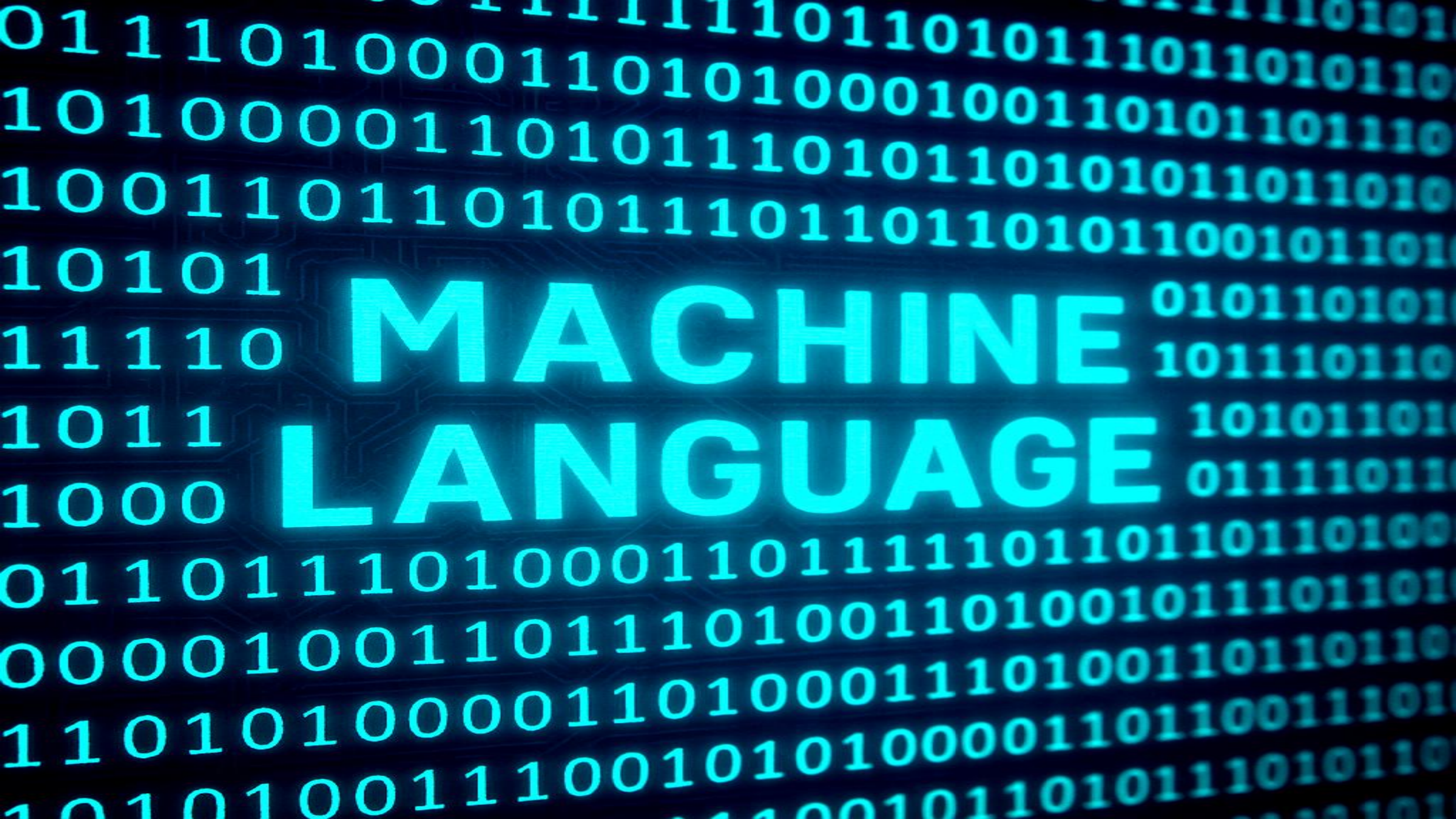
Fifth Generation

# Programming Languages

- Machine Language:
  - It is a low level programming language, also called machine code or object code.
  - It is the only language understood directly by computer's central processing unit because it is a collection of binary digits. It has no need of translator program.
  - At the early era of programming it was used for program coding.
  - While easily understood by computers, machine languages are almost impossible for humans to use because they consist of numbers, that is, series of '0' and '1'.

# Programming Languages

- Coding in machine language is very difficult and has more possibilities of error.
- Machine language instruction has two parts:
- one is the operation code or opcode that specifies the operation to be performed and the other is operand such as data on which the operation should act.


MACHINE
LANGUAGE

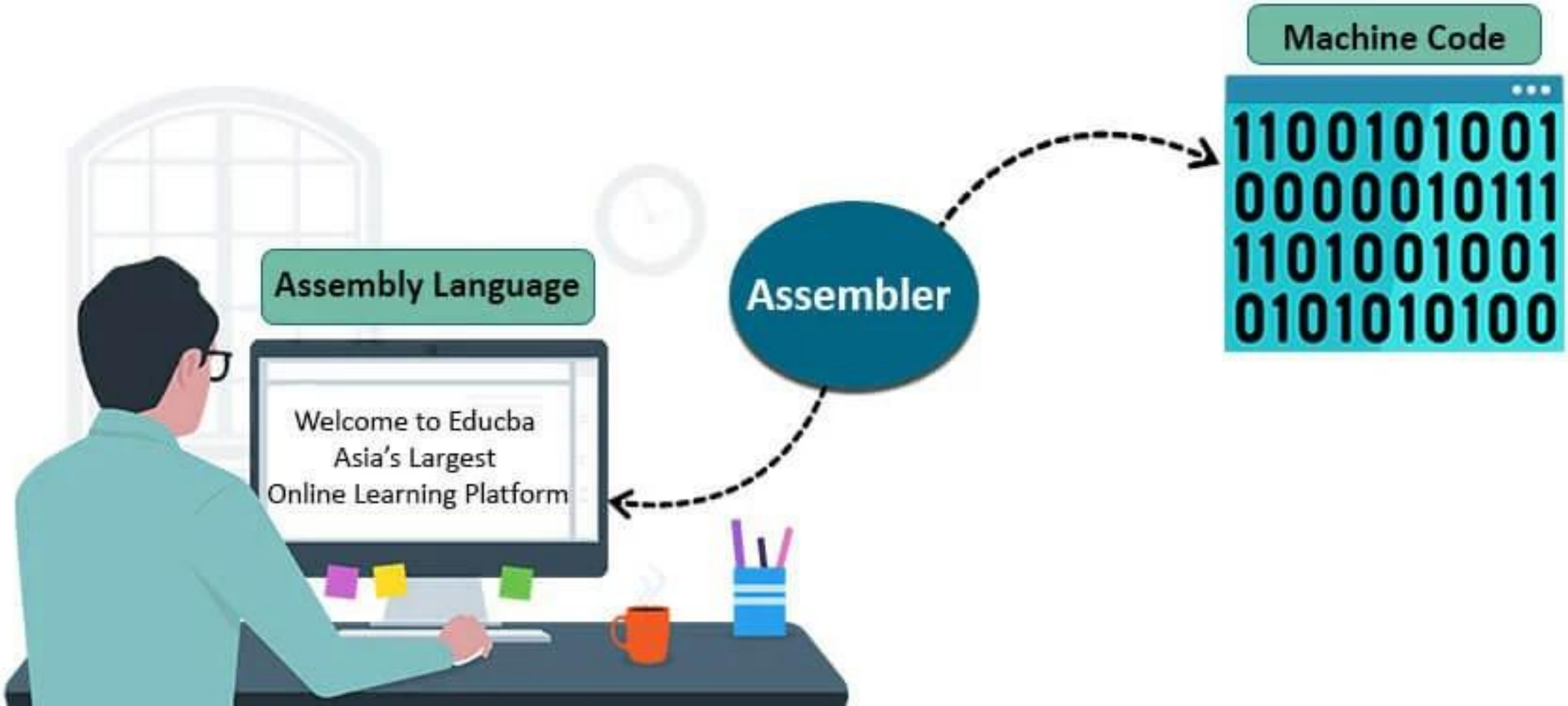
# Programming Languages

- Assembly Language:
  - Assembly language was developed to make coding easier than machine language.
  - At the place of binary code of machine language mnemonic code and symbolic addresses were developed, that were easy to remember.
  - This symbolic language made program writing easy.

# Programming Languages

- But it must be translated into machine codes before being used operationally.
- The program used to convert or translate programs written in assembly code to machine code is called assembler.
- Coding in assembly language is simpler than machine language and error detection is easy.

# Assembly Language



Assembly Language

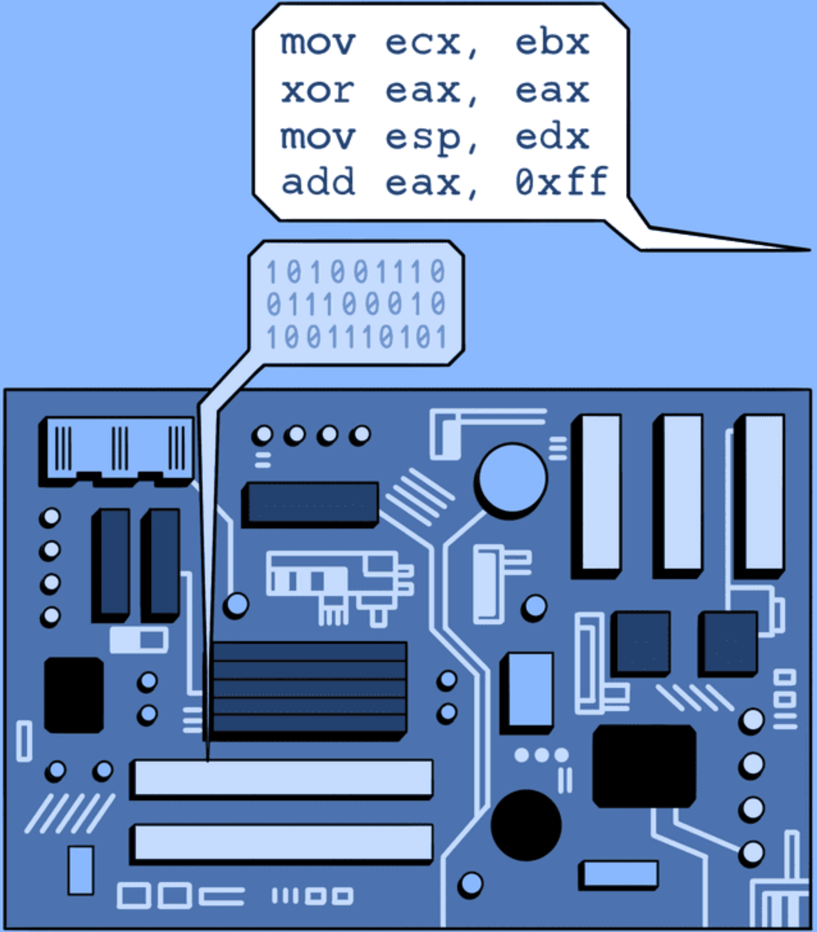
Welcome to Educba
Asia’s Largest
Online Learning Platform

Assembler

Machine Code

1100101001
0000010111
1101001001
0101010100

# Assembly Language



```
mov ecx, ebx
xor eax, eax
mov esp, edx
add eax, 0xff
```

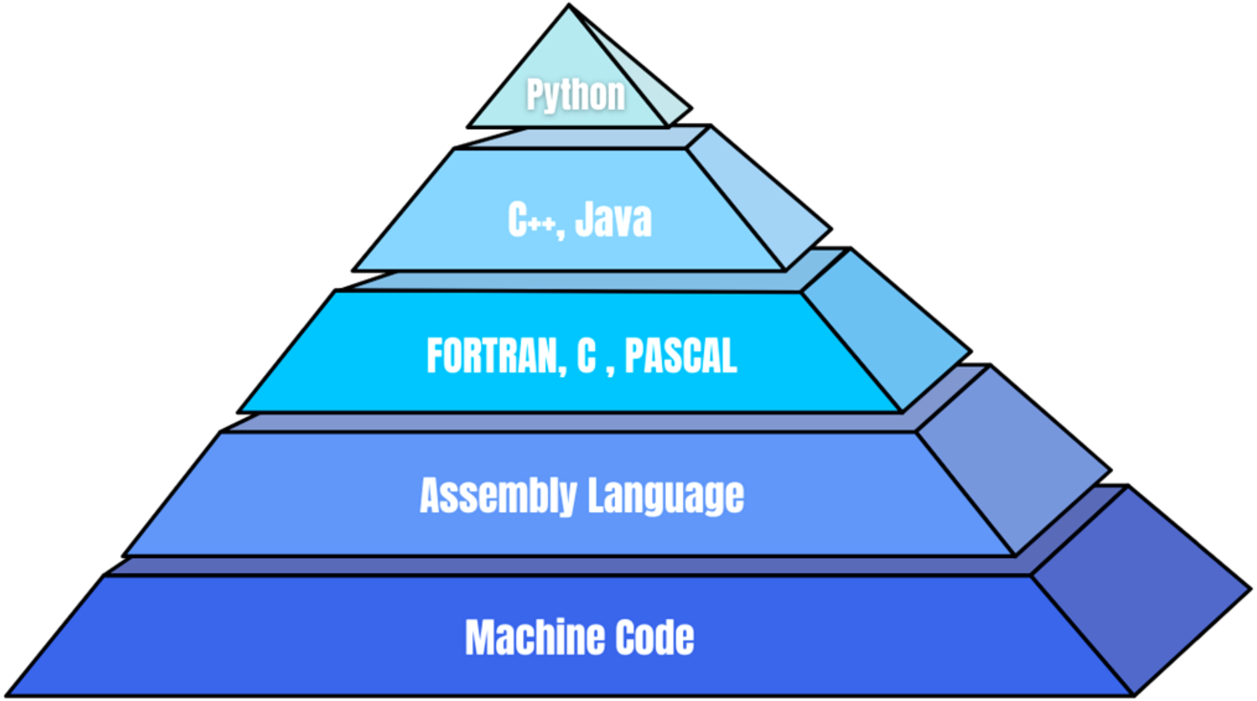
```
1010011110
0111000010
1001110101
```

Low-level programming language intended to communicate directly with a computer's hardware.

# Programming Languages

- High level language:
  - High level language is a programming language which is machine independent and uses translator.
  - It is closer to human languages.
  - It is also called a source code.
  - Some commonly used high level languages are C, BASIC, FORTRAN,ALGOL, PASCAL etc.

High Level Language



Python

C++, Java

FORTRAN, C , PASCAL

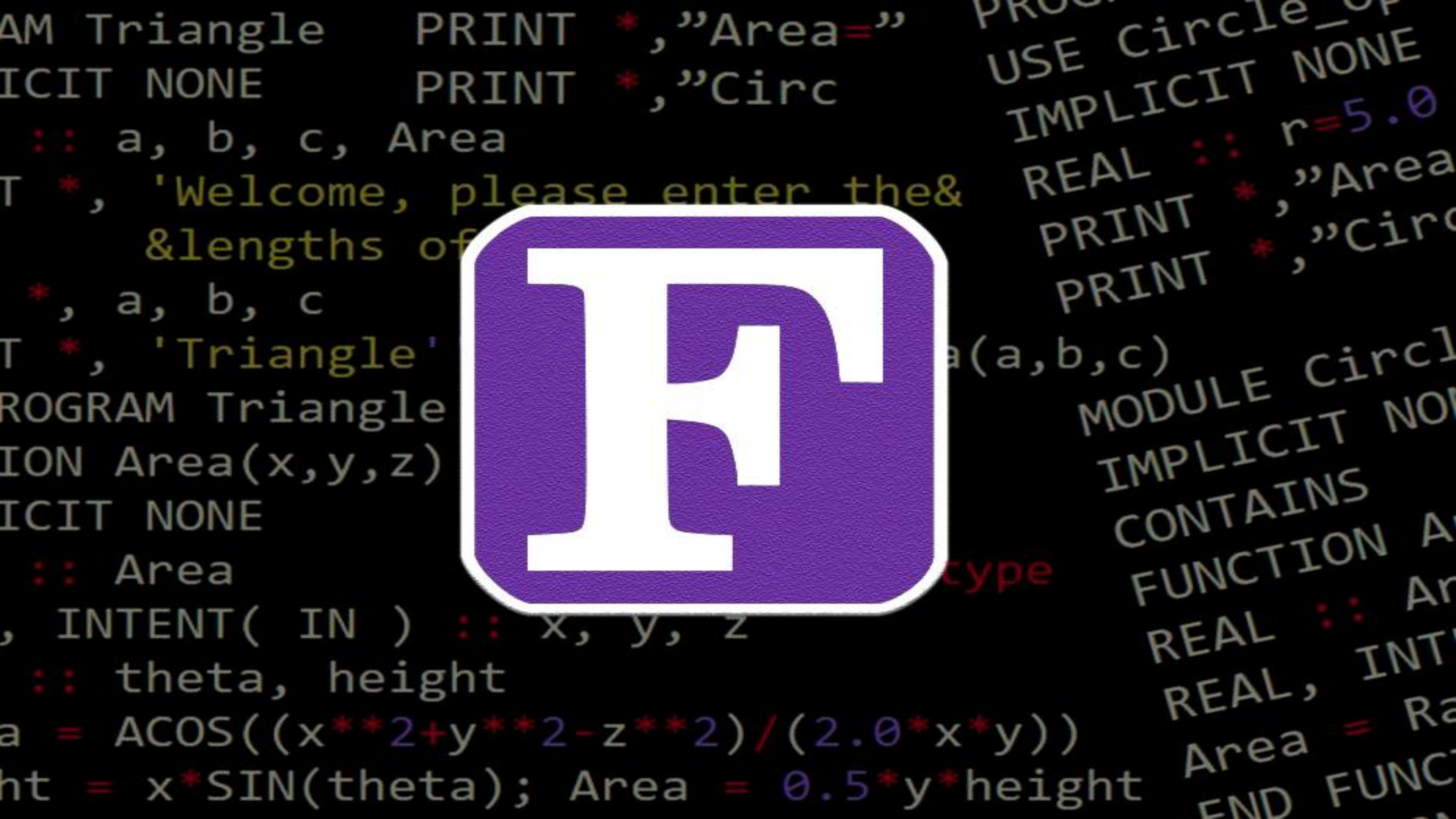
Assembly Language

Machine Code

Low Level Language

# Programming Languages

- There are five types of high level languages to solve a wide variety of problems.
- 1. Scientific language:
  - It is a programming
    language that was designed for the use of mathematical formulas and matrices.
  - Although all programming languages allow for this kind of processing but scientific language makes easier to express these actions.
  - Examples are FORTRAN, ALGOL etc.



```
AM Triangle   PRINT *,”Area=”
ICIT NONE     PRINT *,”Circ
 :: a, b, c, Area
T *, 'Welcome, please enter the&
     &lengths of
 *, a, b, c
T *, 'Triangle'            a(a,b,c)
ROGRAM Triangle
ION Area(x,y,z)
ICIT NONE
 :: Area                   type
, INTENT( IN ) :: x, y, z
 :: theta, height
a = ACOS((x**2+y**2-z**2)/(2.0*x*y))
ht = x*SIN(theta); Area = 0.5*y*height
```

```
PROG
USE Circle_Op
IMPLICIT NONE
REAL :: r=5.0
PRINT *,”Area
PRINT *,”Circ
PRINT
MODULE Circl
IMPLICIT NON
CONTAINS
FUNCTION A
REAL :: Ar
REAL, INT
Area = Ra
END FUNC
```

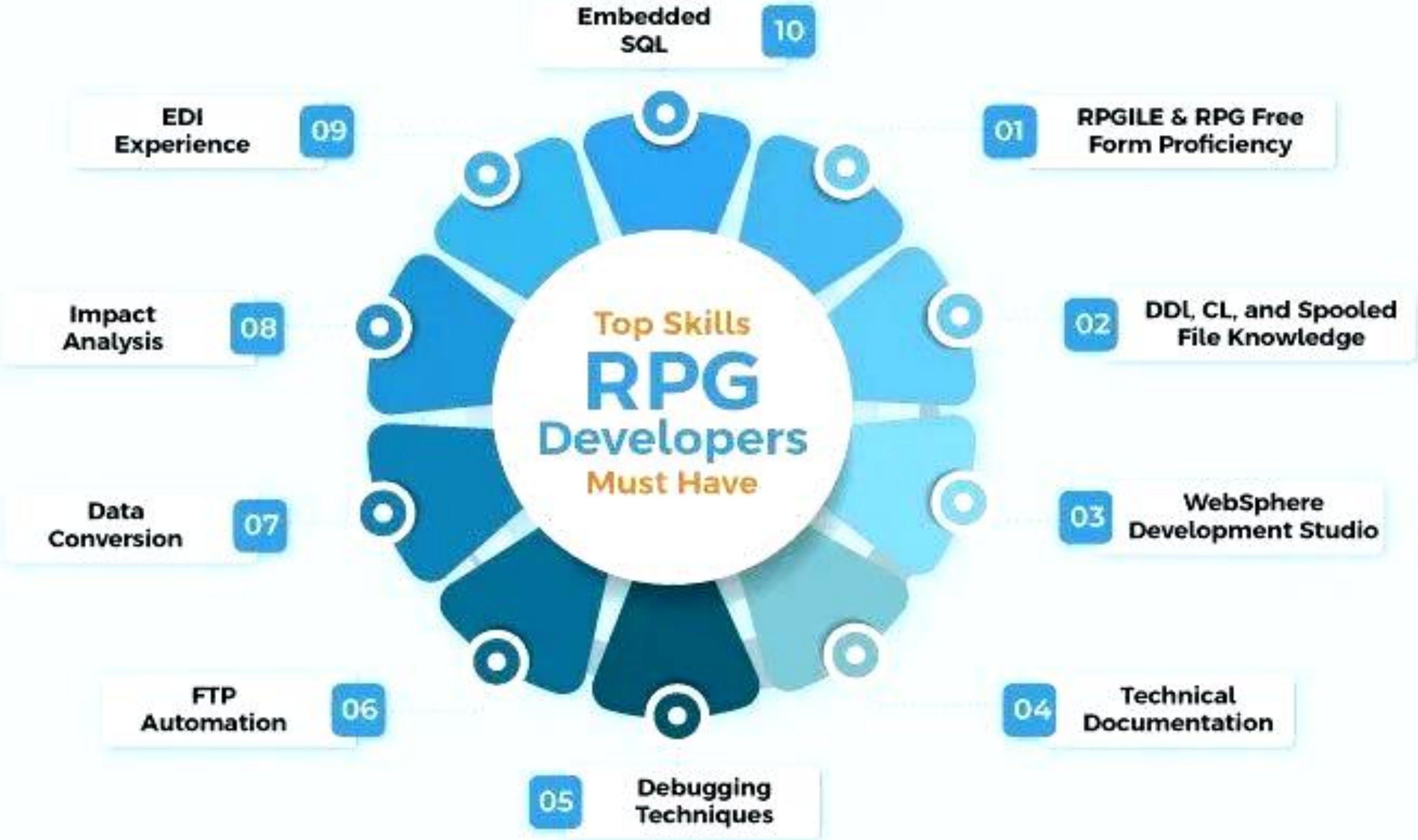
ALGOL

# Programming Languages

- 2. Commercial languages:
  - It is a programming language that was designed for solving everyday commercial problems.
  - Examples are COBOL, RPG, etc.

# COBOL

```
*> two numbers aafter the decimal. by default
01 FIRST-VAR PIC S9(3)V9(2).
*> do the same thing as above but actually ini
*> to a number -123.45
01 SECOND-VAR PIC S9(3)V9(2) VALUE
*> defines an alphabetic string and initialize it t
01 THIRD-VAR PIC A(6) VALUE 'ABCDEF'.
*> define an alphanumeric string and initialize
01 FOURTH-VAR PIC X(5) VALUE 'A121$'.
*> create a grouped variable
01 GROUP-VAR.
```



Top Skills

RPG

Developers

Must Have

01 RPGILE & RPG Free Form Proficiency

02 DDl, CL, and Spooled File Knowledge

03 WebSphere Development Studio

04 Technical Documentation

05 Debugging Techniques

06 FTP Automation

07 Data Conversion

08 Impact Analysis

09 EDI Experience

10 Embedded SQL

# Programming Languages

- 3. Special purpose languages:
  - It is a programming language that was
  - designed for a specific function such as payroll, simulation etc.
  - Examples are ADA, Modula and Modula, SQL, QUEL etc.

SQL

# Programming Languages

- 4. Multipurpose languages:
  - These are languages intended to cope with a
  - number of different types of application area such as business and scientific.
  - Examples are APL, BASIC, PL1,C and PASCAL

statement
POINTER
function
variables
array
PROGRAM
TYPE
return
define
int
char
CHARACTER
USED
operator
structure
string
declaration
double
write

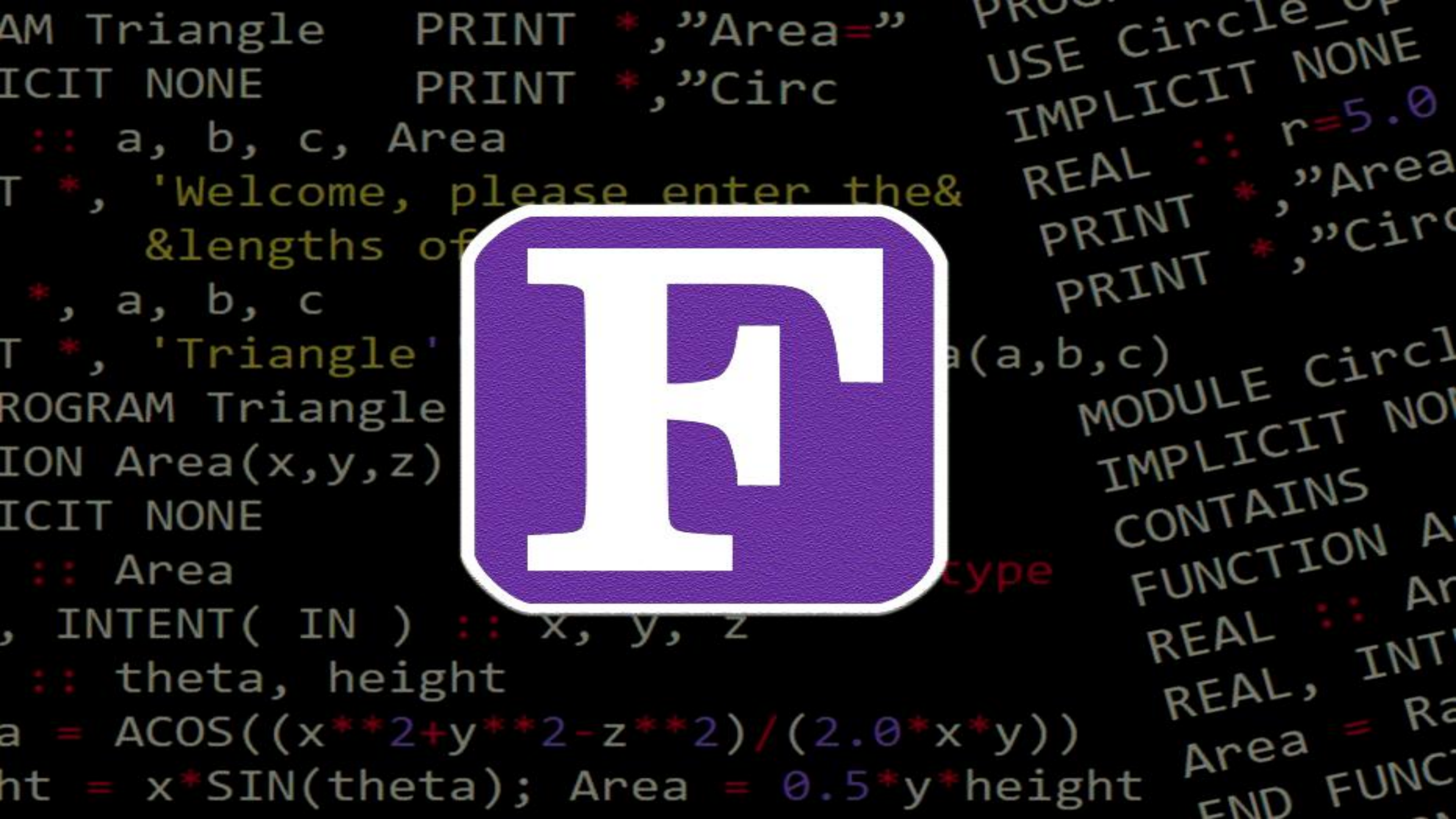
# Programming Languages

- 5. Command languages for operating system:
  - These Languages are used to control operation of a computer.
  - Most command languages are specific to the particular manufacturer's operating system.
  - Examples are DCL, SHELL, MS-DOS.

```
                         Welcome

Welcome to DOS         and        S         program.  SELECT
will install DOS          o       ix        disk or diskette
If you install                    a        te, the number of
blank diskettes                  e ds          the type and
capacity of you

   Drive Type                                  f Diskettes

   5.25-Inch D                                .25  (360KB)
   5.25-Inch D                                .25  (360KB)
   3.5-Inch Di                                 5   (720KB)
   3.5-Inch D                                  5   (720KB)
```

# Programming Languages

- 1. FORTRAN (Formula Translation):
  - FORTRAN was the first high level programming language invented by John Backus for IBM 704 in October 1956 but the first FORTRAN compiler delivered in April 1957.
  - The language was widely adopted by scientists and engineers for writing numerically intensive programs, which encouraged compiler writers to produce compilers that could generate faster and more efficient code.
  - Fortran is still used today for programming scientific and mathematical applications such as mathematical calculation, function and formula.



AM Triangle
ICIT NONE
:: a, b, c, Area
T *, 'Welcome, please enter the&
&lengths of
*, a, b, c
T *, 'Triangle'
ROGRAM Triangle
ION Area(x,y,z)
ICIT NONE
:: Area
, INTENT( IN ) :: x, y, z
:: theta, height
a = ACOS((x**2+y**2-z**2)/(2.0*x*y))
ht = x*SIN(theta); Area = 0.5*y*height
PRINT *,"Area="
PRINT *,"Circ
a(a,b,c)
type
USE Circle_op
IMPLICIT NONE
REAL :: r=5.0
PRINT *,"Area
PRINT *,"Circ
PRINT
MODULE Circl
IMPLICIT NON
CONTAINS
FUNCTION A
REAL :: Ar
REAL, INT
Area = Ra
END FUNC
F

# Programming Languages

- 2. ALGOL (Algorithmic Language):
  - It has originally developed by John Backus in 1958 known as ALGOL 58.
  - It was revised and expanded by Peter Naur in 1960 and known as ALGOL 60.
  - It used for scientific and engineering purpose and has powerful mathematical facilities.

# ALGOL

# Programming Languages

- 3. COBOL (Common Business Oriented Language):
  - It was one of the earliest high level programming languages.
  - It was developed in 1959 by Grace Hopper.
  - Its primary domain is business, finance, and administrative systems for companies and governments.
  - Group of sentences in this language is called paragraph.

# Programming Languages

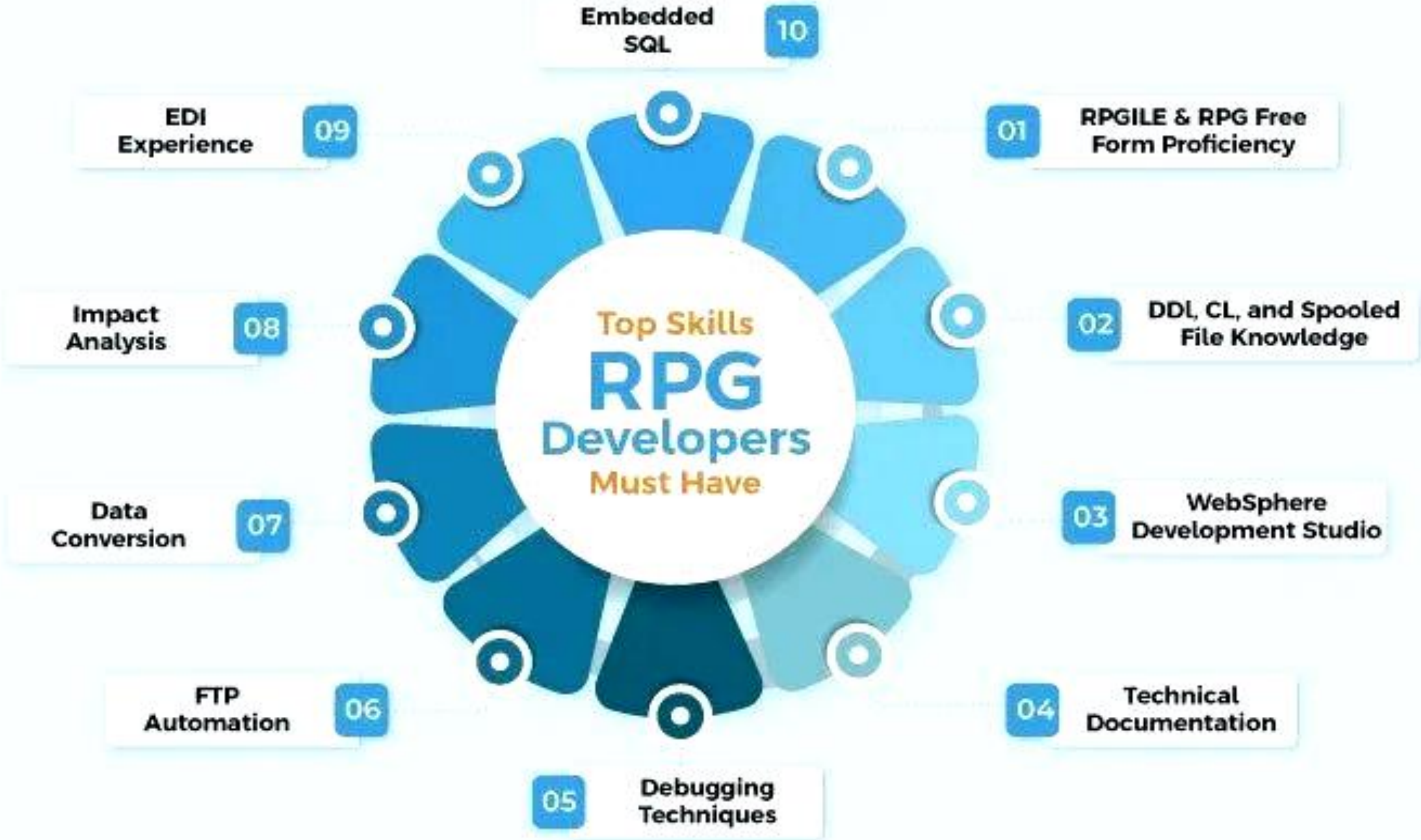
- All paragraphs together make a section and all sections make a division.
- For mathematical terms, COBOL uses ADD, SUBTRACT and MULTIPLY etc.
- It is Eriglish like language and provide much suitable documentation.

```
  Credit Limit:$     0    Finance Charge? Y    Area:        Sort C

                     BILLING                                     SHIPPING
    Name: A CLEAN WELL LIGHTED PLACE FOR      Name: A CLEAN WELL LIG
 Address: 601 VAN NESS AVENUE              Address: 601 VAN NESS AVE
        :                                         :
        :                                         :
    City: SAN FRANCISCO                       City: SAN FRANCISCO
   State: CA                                 State: CA
     Zip: 94102                                Zip: 94102
 Country: U.S.A                            Country:

   Phone:                                    Phone:
```

# COBOL

# Programming Languages

- 4. RPG (Report Program Generator):
  - It is a high level programming language for business applications, which generates report.
  - It is developed by IBM in 1961 and primary vendor of RPG is also IBM.



Top Skills

RPG Developers

Must Have

01 RPGILE & RPG Free Form Proficiency

02 DDl, CL, and Spooled File Knowledge

03 WebSphere Development Studio

04 Technical Documentation

05 Debugging Techniques

06 FTP Automation

07 Data Conversion

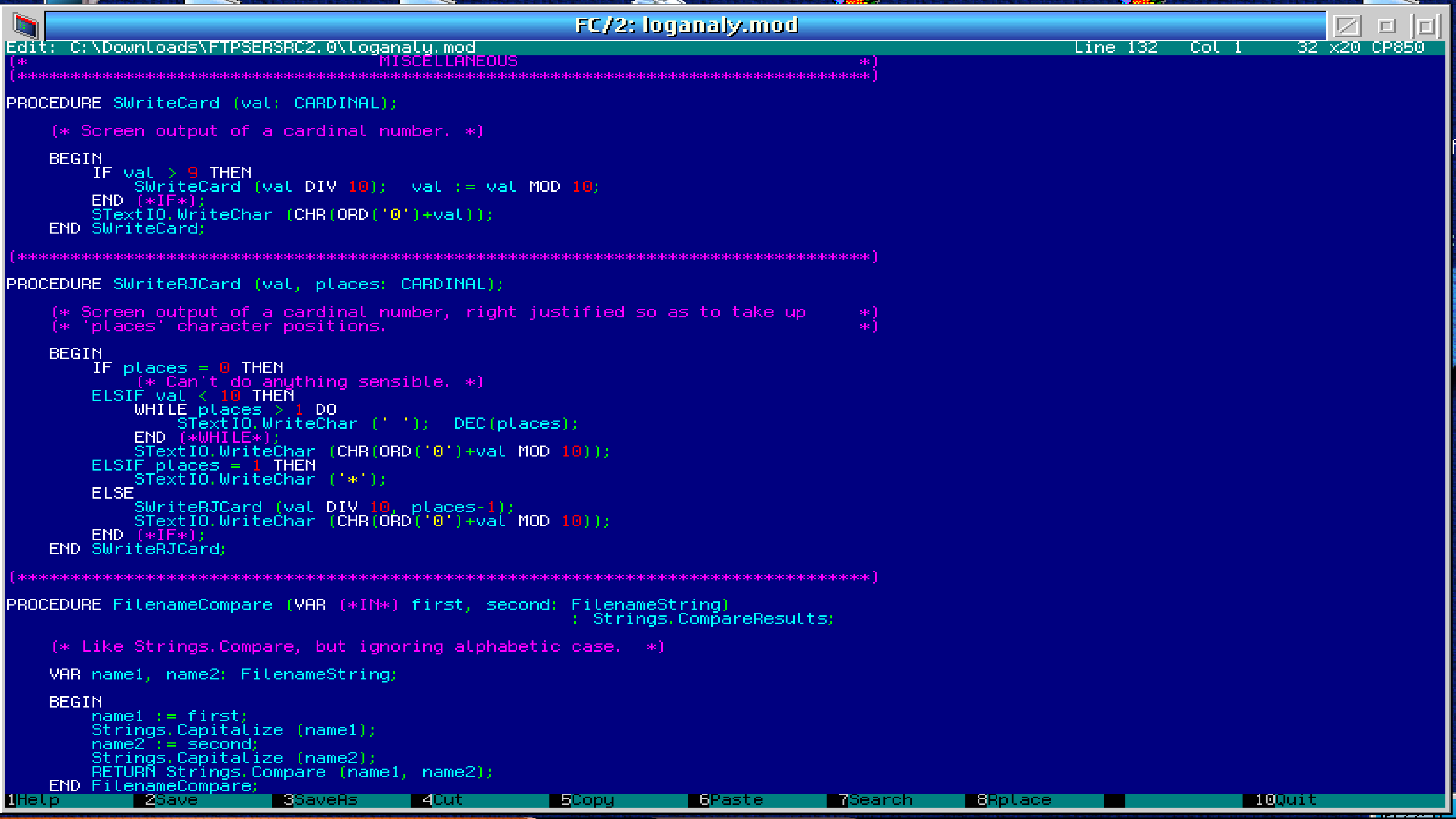
08 Impact Analysis

09 EDI Experience

10 Embedded SQL

# Programming Languages

- 5. Modula and Modula 2:
  - Modula is a descendant of the programming language Pascal.
  - It was developed in Switzerland in the late 1970s by Nicklaus Wirth.
  - The main innovation of Modula over Pascal is a module system, used for grouping sets of related declarations into program units.
  - Modula 2 is a computer programming language invented by Niklaus Wirth around 1978, as a successor to Modula.
  - It is specially suited to computer systems development work.



```
(*                                  MISCELLANEOUS                                  *)
(********************************************************************************)

PROCEDURE SWriteCard (val: CARDINAL);

    (* Screen output of a cardinal number. *)

    BEGIN
        IF val > 9 THEN
            SWriteCard (val DIV 10);  val := val MOD 10;
        END (*IF*);
        STextIO.WriteChar (CHR(ORD('0')+val));
    END SWriteCard;

(********************************************************************************)

PROCEDURE SWriteRJCard (val, places: CARDINAL);

    (* Screen output of a cardinal number, right justified so as to take up     *)
    (* 'places' character positions.                                            *)

    BEGIN
        IF places = 0 THEN
            (* Can't do anything sensible. *)
        ELSIF val < 10 THEN
            WHILE places > 1 DO
                STextIO.WriteChar (' ');  DEC(places);
            END (*WHILE*);
            STextIO.WriteChar (CHR(ORD('0')+val MOD 10));
        ELSIF places = 1 THEN
            STextIO.WriteChar ('*');
        ELSE
            SWriteRJCard (val DIV 10, places-1);
            STextIO.WriteChar (CHR(ORD('0')+val MOD 10));
        END (*IF*);
    END SWriteRJCard;

(********************************************************************************)

PROCEDURE FilenameCompare (VAR (*IN*) first, second: FilenameString)
                                                     : Strings.CompareResults;

    (* Like Strings.Compare, but ignoring alphabetic case.  *)

    VAR name1, name2: FilenameString;

    BEGIN
        name1 := first;
        Strings.Capitalize (name1);
        name2 := second;
        Strings.Capitalize (name2);
        RETURN Strings.Compare (name1, name2);
    END FilenameCompare;
```

# Programming Languages

- 6. SQL (Structured Query Language), QUEL (Query Language) are examples of database query language.

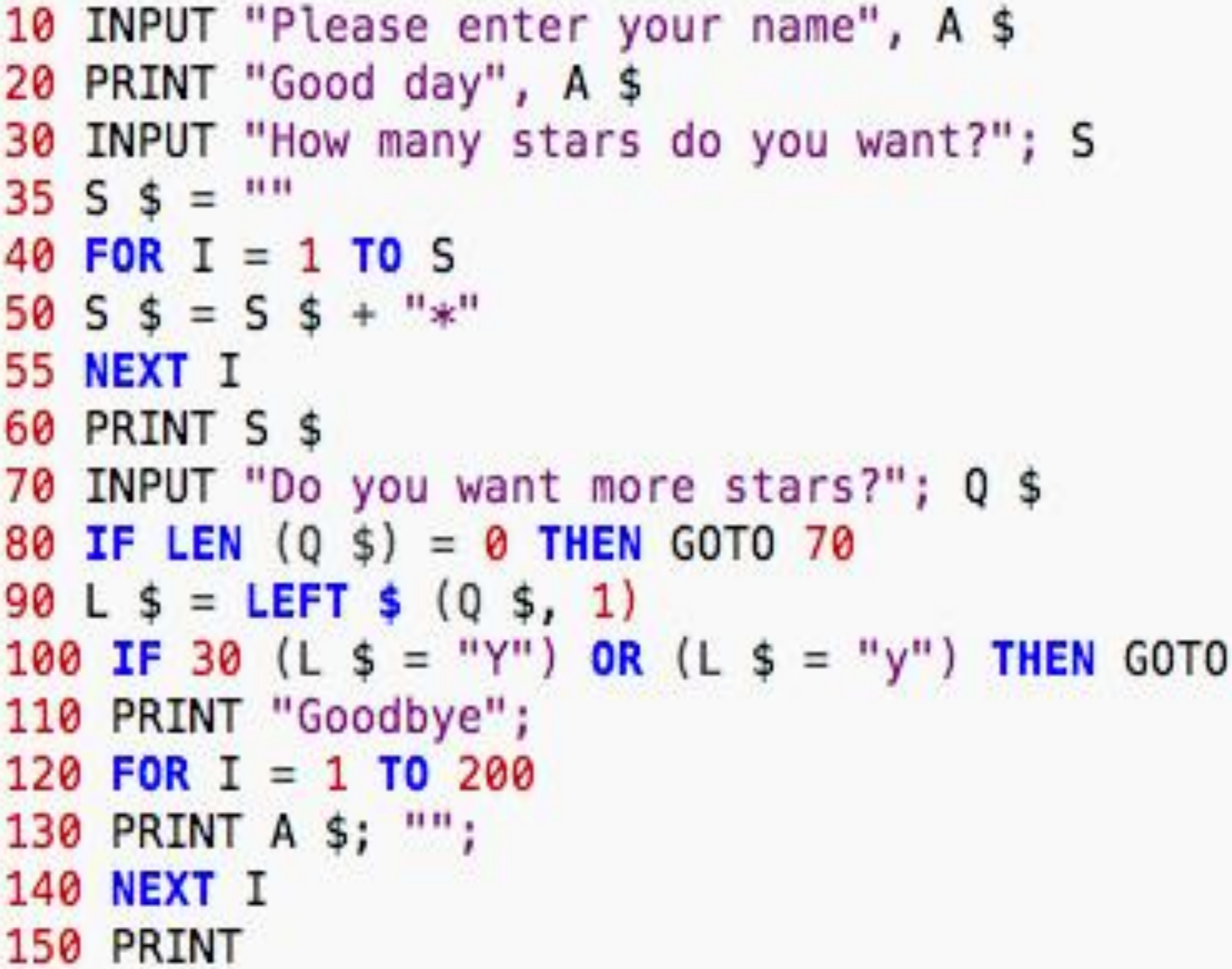
# Programming Languages

- 7. APL (A Programming Language):
  - It was developed in 1964 by Kenneth E. Iverson.
  - It is an oriented interactive language for algorithmic processing which is available from a number of commercial and non-commercial vendors for most computer platforms.
  - It is a specially powerful language in defining vectors and matrices.

```
        ∇DET[⎕]∇
      ∇ Z←DET A;B;P;I
[1]     I←⎕IO
[2]     Z←1
[3]   L:P←(|A[;I])⍳⌈/|A[;I]
[4]     →(P=I)/LL
[5]     A[I,P;]←A[P,I;]
[6]     Z←-Z
[7]   LL:Z←Z×B←A[I;I]
[8]     →(0 1 ∨.=Z,1↑⍴A)/0
[9]     A←1 1 ↓A-(A[;I]÷B)∘.×A[I;]
[10]    →L
[11]  ⍝EVALUATES A DETERMINANT
      ∇
```

# Programming Languages

- 8. BASIC (Beginner's All purpose Symbolic Instruction Code):
  - It was designed in 1964 by John George Kemeny and Thomas Eugene Kurtz to provide computer access to non-science students.
  - It is simple, powerful and interactive language for beginners and provides clear error message.
  - It allows advanced features to be added for experts so it is used by both scientists and businessmen.



```
10 INPUT "Please enter your name", A $
20 PRINT "Good day", A $
30 INPUT "How many stars do you want?"; S
35 S $ = ""
40 FOR I = 1 TO S
50 S $ = S $ + "*"
55 NEXT I
60 PRINT S $
70 INPUT "Do you want more stars?"; Q $
80 IF LEN (Q $) = 0 THEN GOTO 70
90 L $ = LEFT $ (Q $, 1)
100 IF 30 (L $ = "Y") OR (L $ = "y") THEN GOTO
110 PRINT "Goodbye";
120 FOR I = 1 TO 200
130 PRINT A $; "";
140 NEXT I
150 PRINT
```

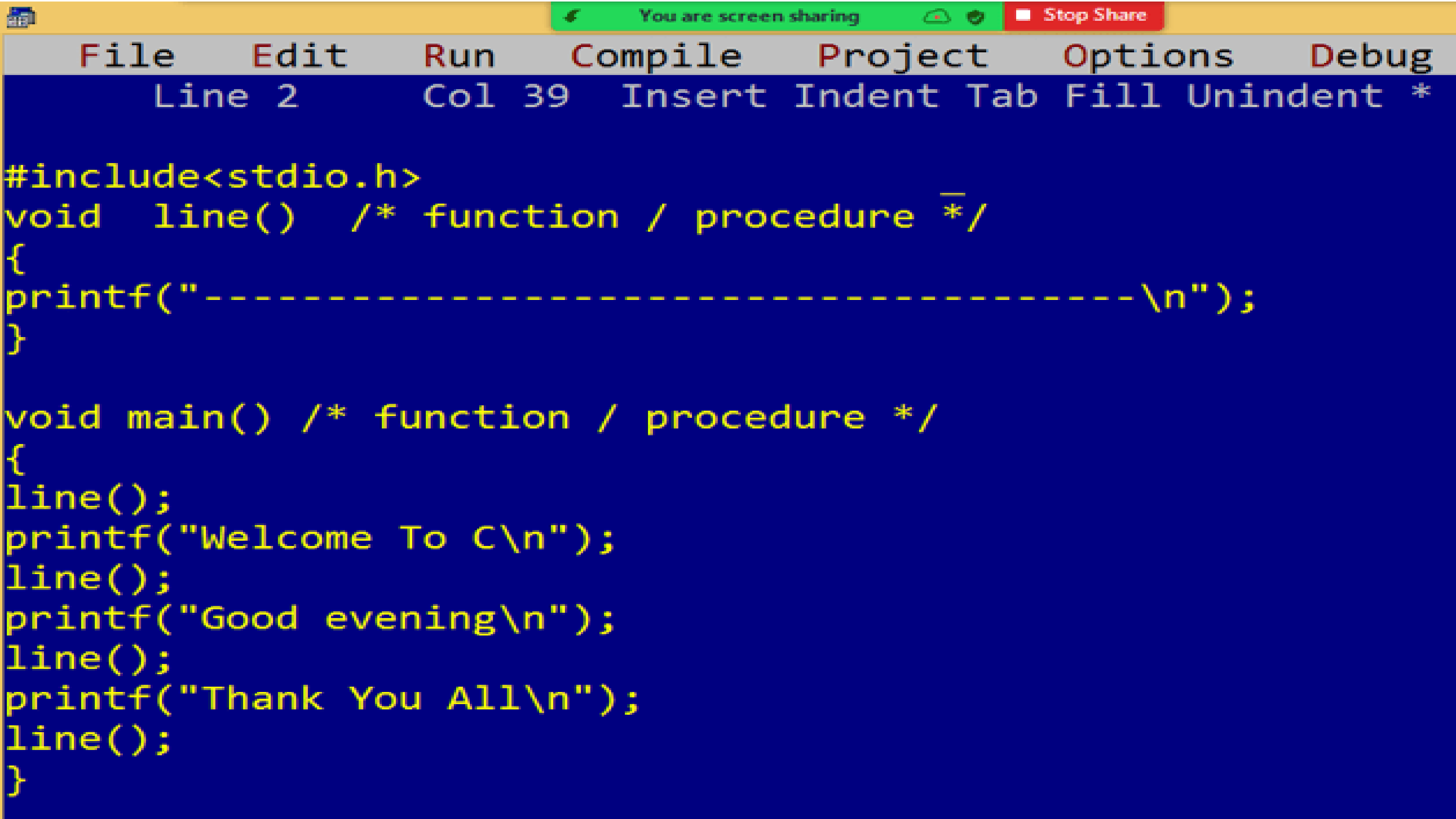
# Programming Languages

- 9. PL1 (Programming Language One):
  - It was developed by IBM in the early 1960s, and is still actively used.
  - It is designed for scientific, engineering, and business applications.
  - It has been used by various academic, commercial and industrial users.
  - It is a very successful language except that its multipurpose facilities made it too large for use on small machines.

```
@RCCHECK_HELPER: PROC(P,RC) RETURNS(FIXED BIN(31));
  DCL P POINTER;
  DCL RETURNDATA CHAR(8) BASED(P);
  DCL RC CHAR(8);

  IF RETURNDATA = RC THEN DO;
    RETURN(1);
  END;

  RETURN(0);
END @RCCHECK_HELPER;

%ACTIVATE RCCHECK;
%RCCHECK: PROC(ARG1,ARG2) RETURNS (CHAR);
  DCL ARG1 CHAR;
  DCL ARG2 CHAR;
  RETURN (' @RCCHECK_HELPER(ADDR(' || ARG1 || '),' || ARG2 || ')');
%END RCCHECK;
```

# Programming Languages

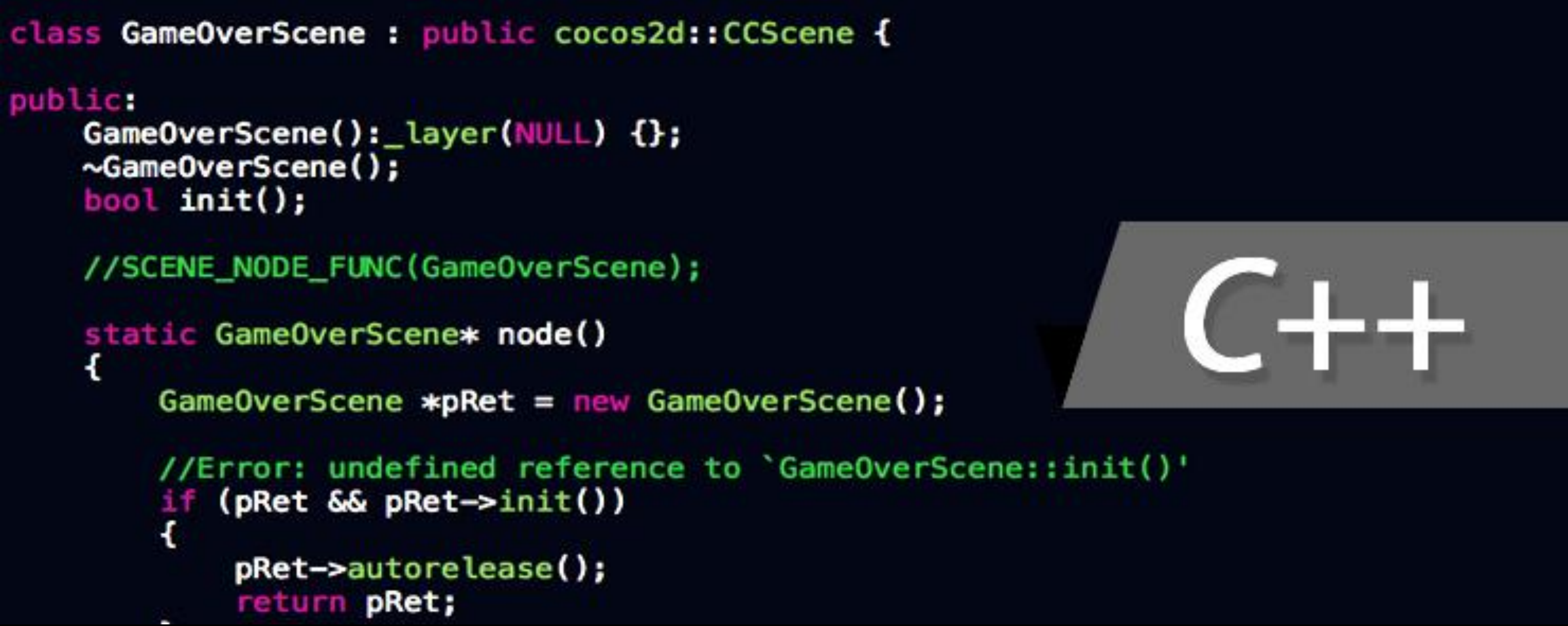
- 10. C:
  - C is a general purpose computer language developed in 1972 by Dennis Ritchie at the Bell Laboratories for use on the Unix Operating System.
  - Although C was designed for implementing system software, it is also widely used for developing portable application software.
  - It is one of the most popular programming language and it is widely used on many different platforms.



```
#include<stdio.h>
void  line()  /* function / procedure */
{
printf("--------------------------------------\n");
}

void main() /* function / procedure */
{
line();
printf("Welcome To C\n");
line();
printf("Good evening\n");
line();
printf("Thank You All\n");
line();
}
```

# Programming Languages

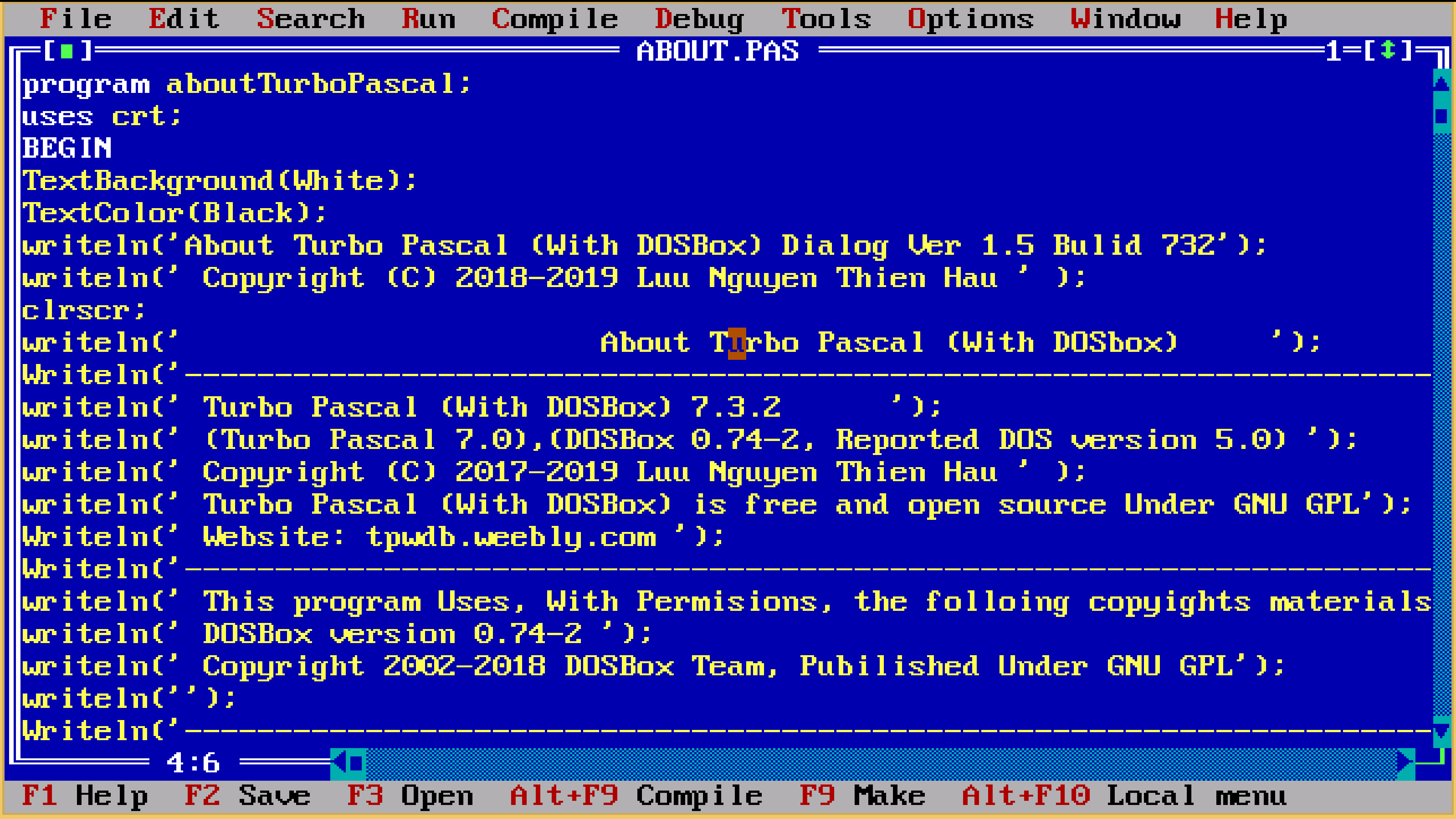
- 11. C++:
- It is object oriented general purpose programming language.
- It is regarded as a middle level language, as it comprises a combination of both high level and low level language features.
- It is better than C programming language but tough to code.



```cpp
class GameOverScene : public cocos2d::CCScene {

public:
    GameOverScene():_layer(NULL) {};
    ~GameOverScene();
    bool init();

    //SCENE_NODE_FUNC(GameOverScene);

    static GameOverScene* node()
    {
        GameOverScene *pRet = new GameOverScene();

        //Error: undefined reference to `GameOverScene::init()'
        if (pRet && pRet->init())
        {
            pRet->autorelease();
            return pRet;
```

# Programming Languages

- 12. PASCAL:
  - Pascal is a procedure programming language developed in 1970 by Niklaus Wirth.
  - It supports structured programming than many older languages such as COBOL or FORTRAN.
  - It is based on the ALGOL programming language and named in honor of the French mathematician and philosopher Blaise Pascal.
  - Initially, Pascal was developed to teach students structured programming and teaching purpose.

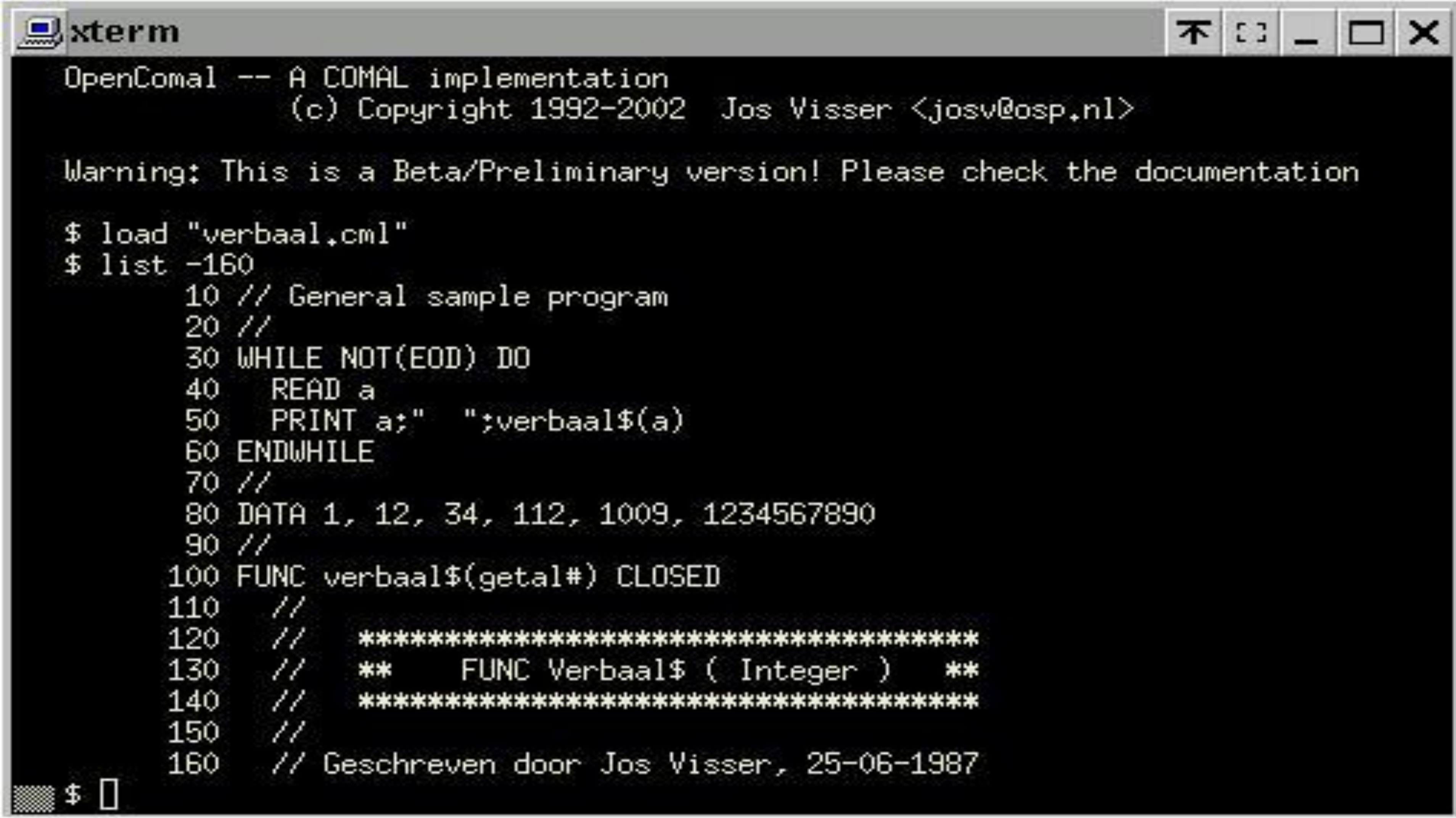


File Edit Search Run Compile Debug Tools Options Window Help
[■]
ABOUT.PAS
1=[↕]
4:6
F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make Alt+F10 Local menu

```
program aboutTurboPascal;
uses crt;
BEGIN
TextBackground(White);
TextColor(Black);
writeln('About Turbo Pascal (With DOSBox) Dialog Ver 1.5 Bulid 732');
writeln(' Copyright (C) 2018-2019 Luu Nguyen Thien Hau ' );
clrscr;
writeln('                       About Turbo Pascal (With DOSbox)     ');
Writeln('---------------------------------------------------------------------
writeln(' Turbo Pascal (With DOSBox) 7.3.2      ');
writeln(' (Turbo Pascal 7.0),(DOSBox 0.74-2, Reported DOS version 5.0) ');
writeln(' Copyright (C) 2017-2019 Luu Nguyen Thien Hau ' );
writeln(' Turbo Pascal (With DOSBox) is free and open source Under GNU GPL');
Writeln(' Website: tpwdb.weebly.com ');
Writeln('---------------------------------------------------------------------
writeln(' This program Uses, With Permisions, the folloing copyights materials
writeln(' DOSBox version 0.74-2 ');
writeln(' Copyright 2002-2018 DOSBox Team, Pubilished Under GNU GPL');
writeln('');
Writeln('---------------------------------------------------------------------
```

# Programming Languages

- 13. COMAL (Common Algorithmic Language):
  - It was developed by Benedict Lofstedt and Borge Christensen in 1973.
  - It was a mixed form of the prevalent educational programming languages namely BASIC and Pascal It was designed to educate students.



```
OpenComal -- A COMAL implementation
             (c) Copyright 1992-2002  Jos Visser <josv@osp.nl>

Warning: This is a Beta/Preliminary version! Please check the documentation

$ load "verbaal.cml"
$ list -160
       10 // General sample program
       20 //
       30 WHILE NOT(EOD) DO
       40   READ a
       50   PRINT a;"  ";verbaal$(a)
       60 ENDWHILE
       70 //
       80 DATA 1, 12, 34, 112, 1009, 1234567890
       90 //
      100 FUNC verbaal$(getal#) CLOSED
      110   //
      120   //   ************************************
      130   //   **    FUNC Verbaal$ ( Integer )   **
      140   //   ************************************
      150   //
      160   // Geschreven door Jos Visser, 25-06-1987
$ 
```

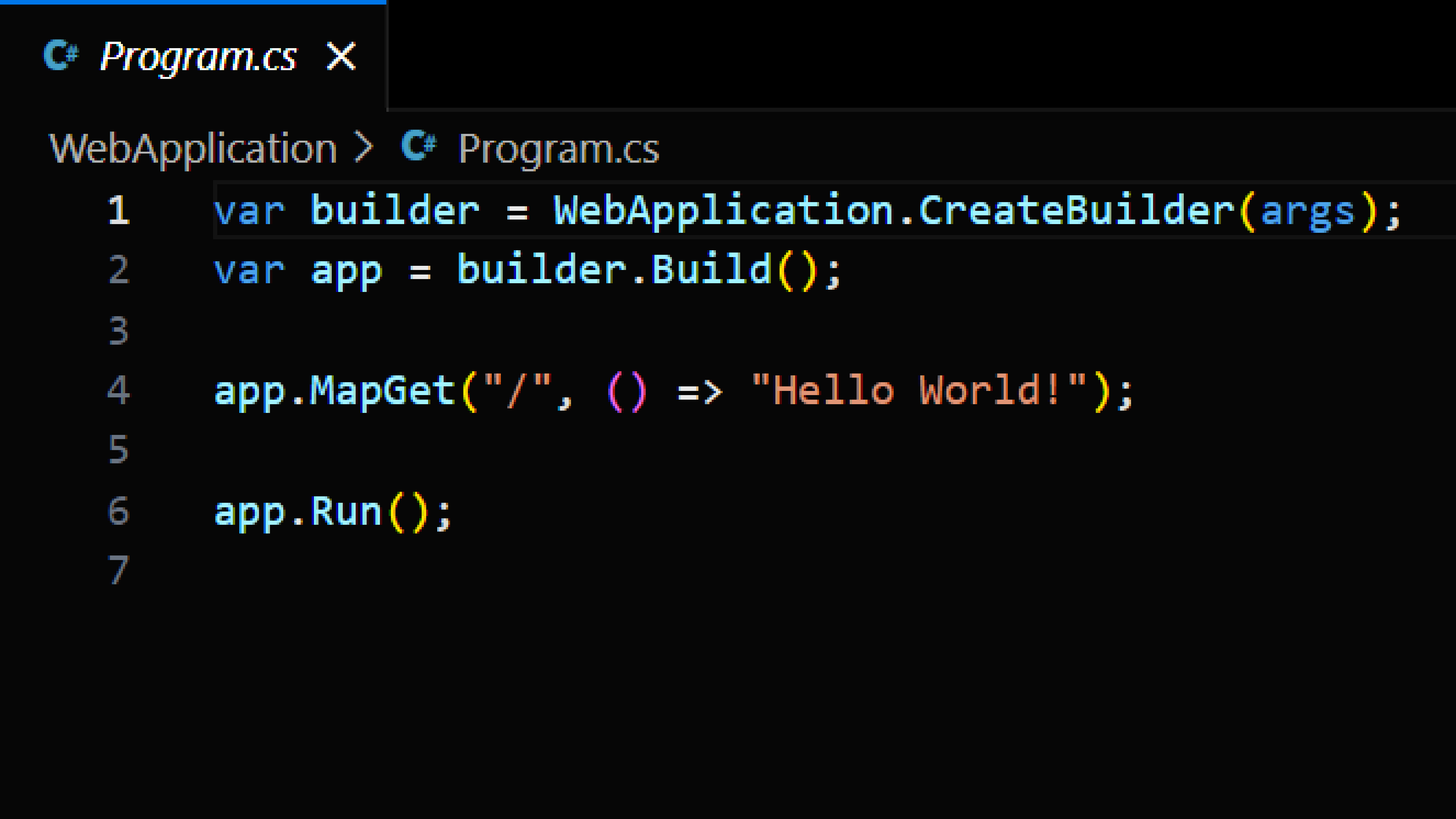
# Programming Languages

- 14. PROLOG (Programming in Logic):
  - It is a general purpose programing language.
  - It has a rich collection of data structure.
  - It is used to develop artificial intelligence.

```
% neutrality(+Matrix,+Exprs,-Exprs): the function $\mathcal{N}$(X)
neutrality(AttM, X, Y) :-
    mv_mult(AttM, X, Z),        % $\mathcal{R}^{+}$(X)
    maplist(bnot, Z, Y).

% innocuousity(+Matrix,+Exprs,-Exprs): the function $\mathcal{I}$(X)
innocuousity(AttM, X, Y) :-
    transpose(AttM, AttM_t),  % transpose operation
    mv_mult(AttM_t, X, Z).      % $\mathcal{R}^{-}$(X)
    maplist(bnot, Z, Y).

% defense(+Matrix,+Exprs,-Exprs): the function $\mathcal{F}$(X)
defense(AttM, X, Y) :-
    neutrality(AttM, X, Z),
    neutrality(AttM, Z, Y).
```

# Programming Languages

- 15. C Sharp:
  - It is a programming language which also expressed as C#.
  - It was developed by Microsoft.
  - It is a simple, modern, general purpose, object oriented programming language.



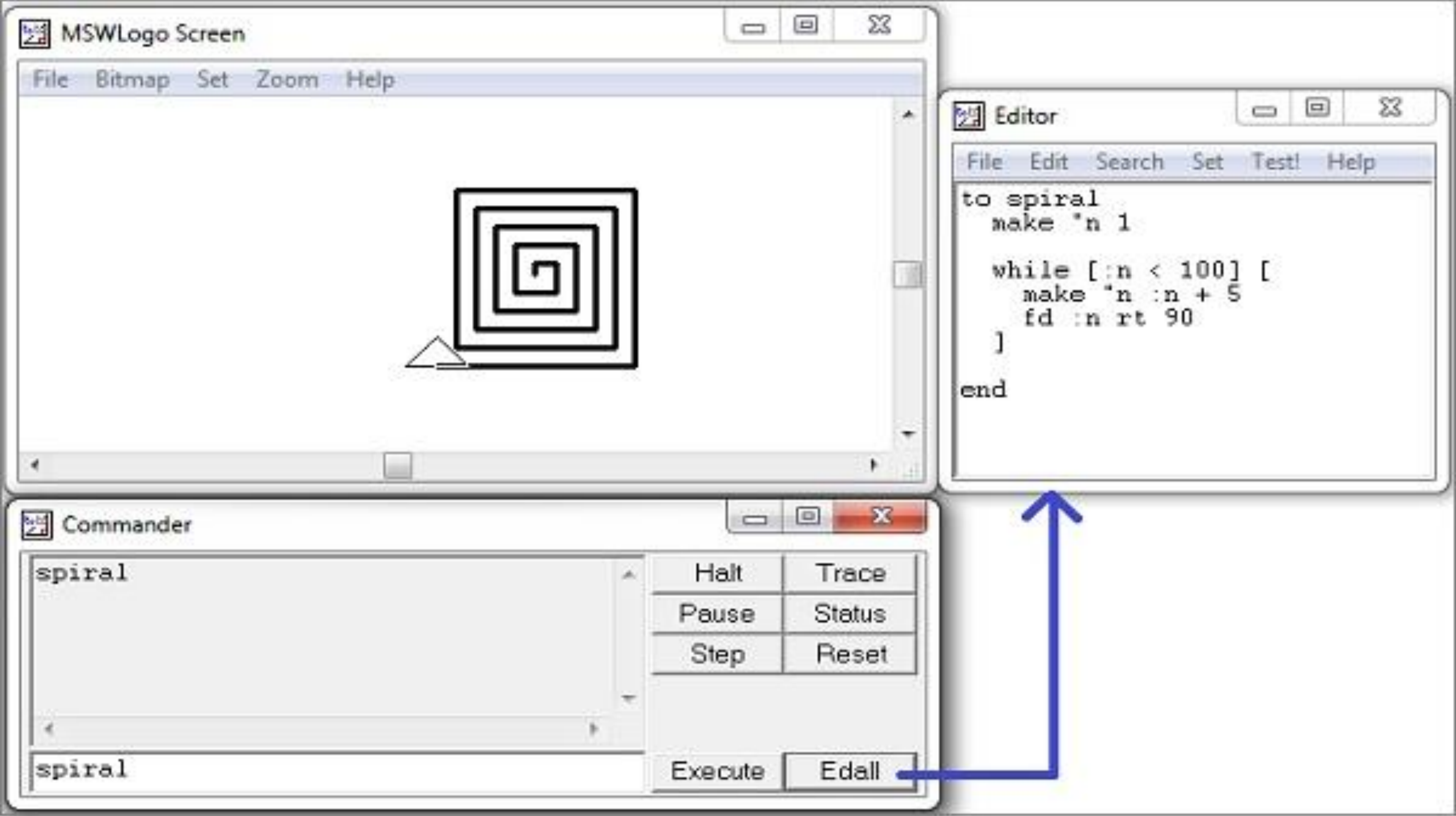
```
var builder = WebApplication.CreateBuilder(args);
var app = builder.Build();

app.MapGet("/", () => "Hello World!");

app.Run();

```

# Programming Languages

- 16. Java:
  - Java is a programming language originally developed by James Gosling at Sun Microsystems and released in 1995 as a core component of Sun Microsystems' Java platform.
  - It is object oriented programming language.
  - It derives much of its syntax from C and C++.
  - It is primarily used in the form of client side JavaScript, implemented as an integrated component of the web browser, allowing the development of enhanced user interfaces and dynamic websites.

# Programming Languages

- 17. LOGO (Logic Oriented Graphic Oriented):
  - LOGO was created in 1967 for educational use and constructive teaching.
  - It is known mainly for its turtle.
  - The turtle moves with commands that are relative to its own position.
  - When turtle moves it draw a line.
  - To teach a child to draw a line and shapes we use logo.
  - It is so easy for a child that they can use it to draw shapes and lines.



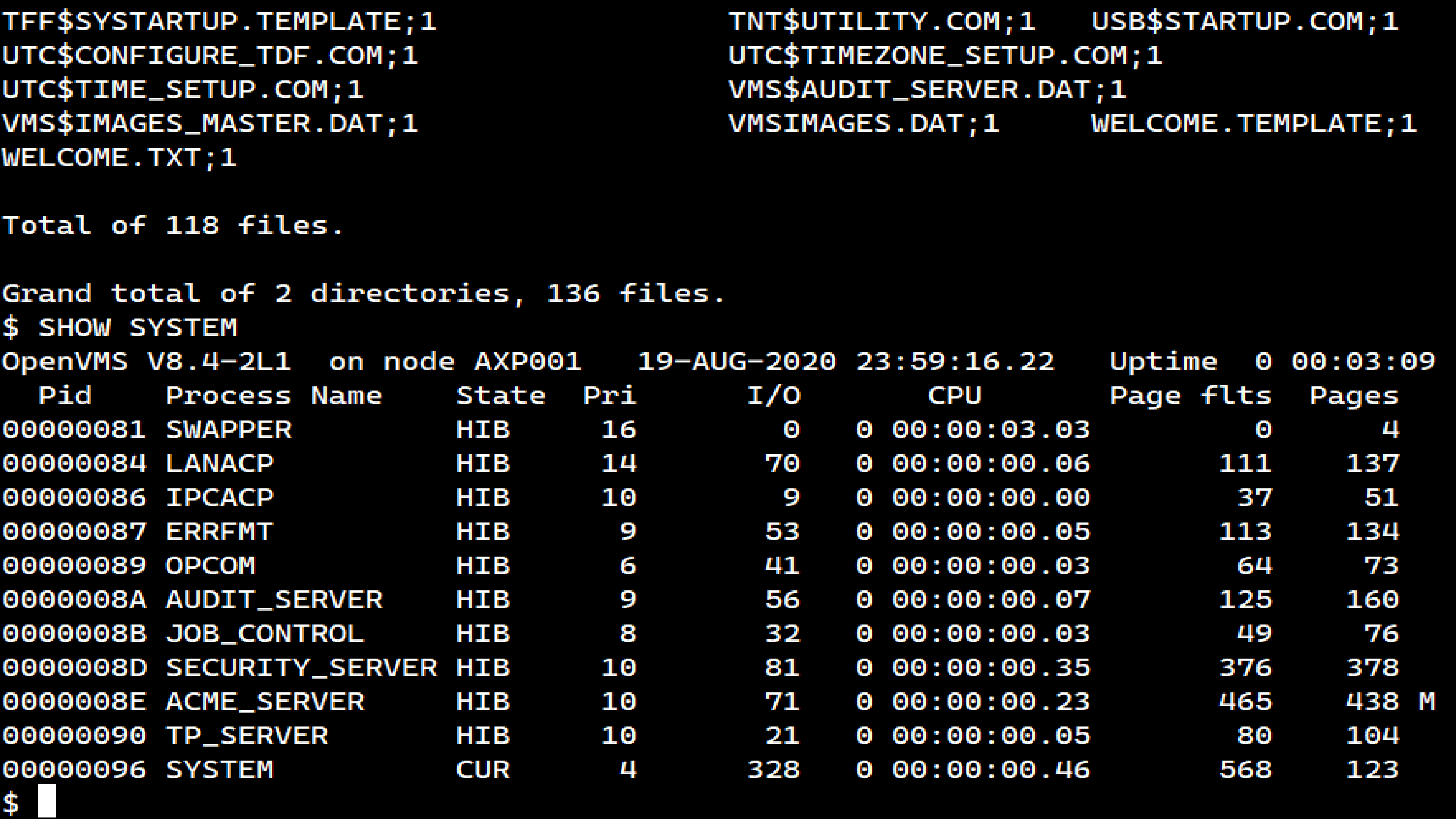
```
to spiral
  make "n 1

  while [:n < 100] [
    make "n :n + 5
    fd :n rt 90
  ]

end
```

```
spiral
```

```
spiral
```

# Programming Languages

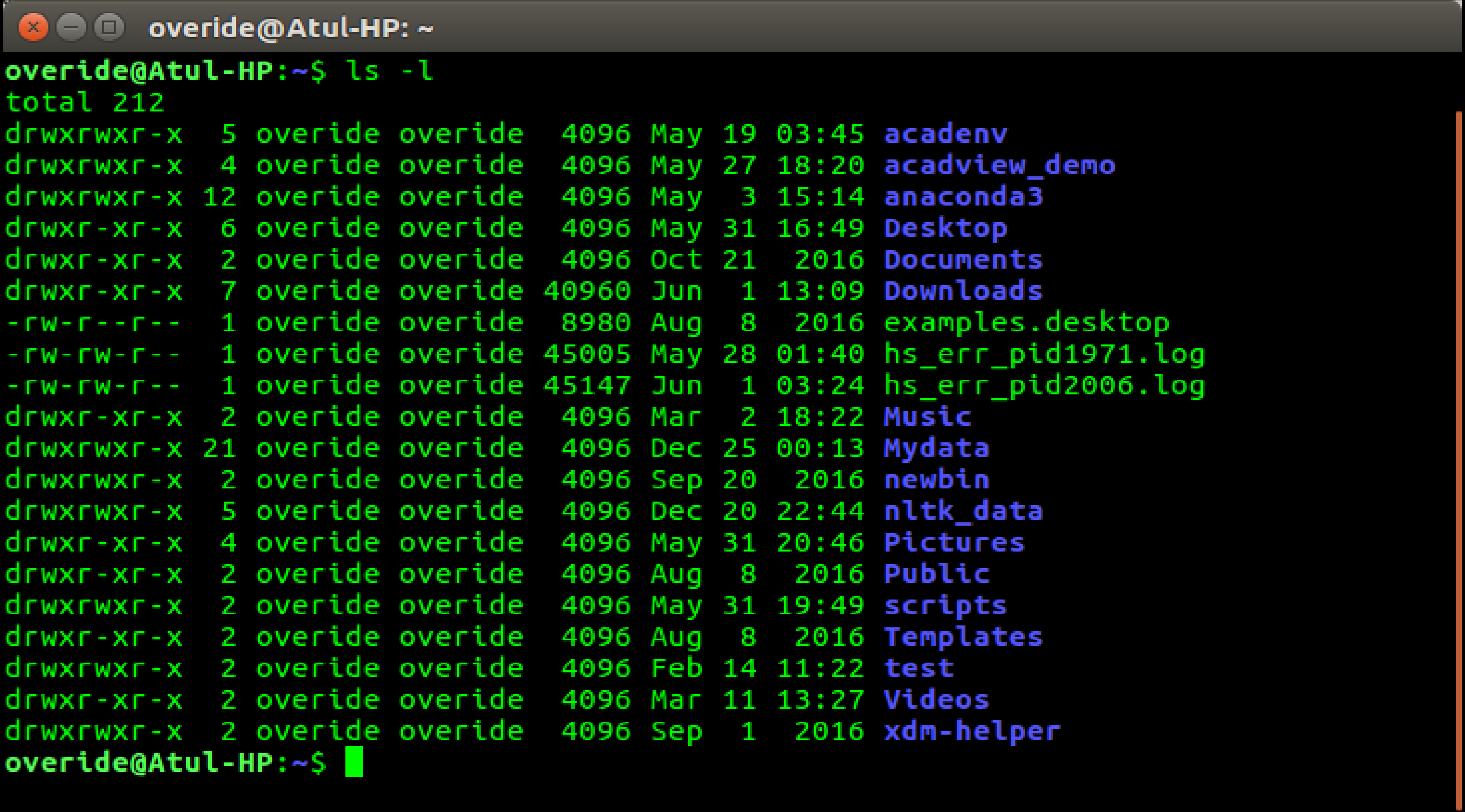
- 18. DCL:
  - It is a command language and used on DEC VAX/VMS operating system.



```
TFF$SYSTARTUP.TEMPLATE;1                TNT$UTILITY.COM;1   USB$STARTUP.COM;1
UTC$CONFIGURE_TDF.COM;1                 UTC$TIMEZONE_SETUP.COM;1
UTC$TIME_SETUP.COM;1                    VMS$AUDIT_SERVER.DAT;1
VMS$IMAGES_MASTER.DAT;1                 VMSIMAGES.DAT;1     WELCOME.TEMPLATE;1
WELCOME.TXT;1

Total of 118 files.

Grand total of 2 directories, 136 files.
$ SHOW SYSTEM
OpenVMS V8.4-2L1  on node AXP001   19-AUG-2020 23:59:16.22   Uptime  0 00:03:09
  Pid    Process Name    State  Pri      I/O       CPU       Page flts  Pages
00000081 SWAPPER         HIB     16        0   0 00:00:03.03         0      4
00000084 LANACP          HIB     14       70   0 00:00:00.06       111    137
00000086 IPCACP          HIB     10        9   0 00:00:00.00        37     51
00000087 ERRFMT          HIB      9       53   0 00:00:00.05       113    134
00000089 OPCOM           HIB      6       41   0 00:00:00.03        64     73
0000008A AUDIT_SERVER    HIB      9       56   0 00:00:00.07       125    160
0000008B JOB_CONTROL     HIB      8       32   0 00:00:00.03        49     76
0000008D SECURITY_SERVER HIB     10       81   0 00:00:00.35       376    378
0000008E ACME_SERVER     HIB     10       71   0 00:00:00.23       465    438 M
00000090 TP_SERVER       HIB     10       21   0 00:00:00.05        80    104
00000096 SYSTEM          CUR      4      328   0 00:00:00.46       568    123
$ 
```

# Programming Languages

- 19. SHELL:
  - It is also a command language and used with Unix operating system.
  - Unix is mostly used for and web servers.



```
overide@Atul-HP:~$ ls -l
total 212
drwxrwxr-x  5 overide overide  4096 May 19 03:45 acadenv
drwxrwxr-x  4 overide overide  4096 May 27 18:20 acadview_demo
drwxrwxr-x 12 overide overide  4096 May  3 15:14 anaconda3
drwxr-xr-x  6 overide overide  4096 May 31 16:49 Desktop
drwxr-xr-x  2 overide overide  4096 Oct 21  2016 Documents
drwxr-xr-x  7 overide overide 40960 Jun  1 13:09 Downloads
-rw-r--r--  1 overide overide  8980 Aug  8  2016 examples.desktop
-rw-rw-r--  1 overide overide 45005 May 28 01:40 hs_err_pid1971.log
-rw-rw-r--  1 overide overide 45147 Jun  1 03:24 hs_err_pid2006.log
drwxr-xr-x  2 overide overide  4096 Mar  2 18:22 Music
drwxrwxr-x 21 overide overide  4096 Dec 25 00:13 Mydata
drwxrwxr-x  2 overide overide  4096 Sep 20  2016 newbin
drwxrwxr-x  5 overide overide  4096 Dec 20 22:44 nltk_data
drwxr-xr-x  4 overide overide  4096 May 31 20:46 Pictures
drwxr-xr-x  2 overide overide  4096 Aug  8  2016 Public
drwxrwxr-x  2 overide overide  4096 May 31 19:49 scripts
drwxr-xr-x  2 overide overide  4096 Aug  8  2016 Templates
drwxrwxr-x  2 overide overide  4096 Feb 14 11:22 test
drwxr-xr-x  2 overide overide  4096 Mar 11 13:27 Videos
drwxrwxr-x  2 overide overide  4096 Sep  1  2016 xdm-helper
overide@Atul-HP:~$ 
```

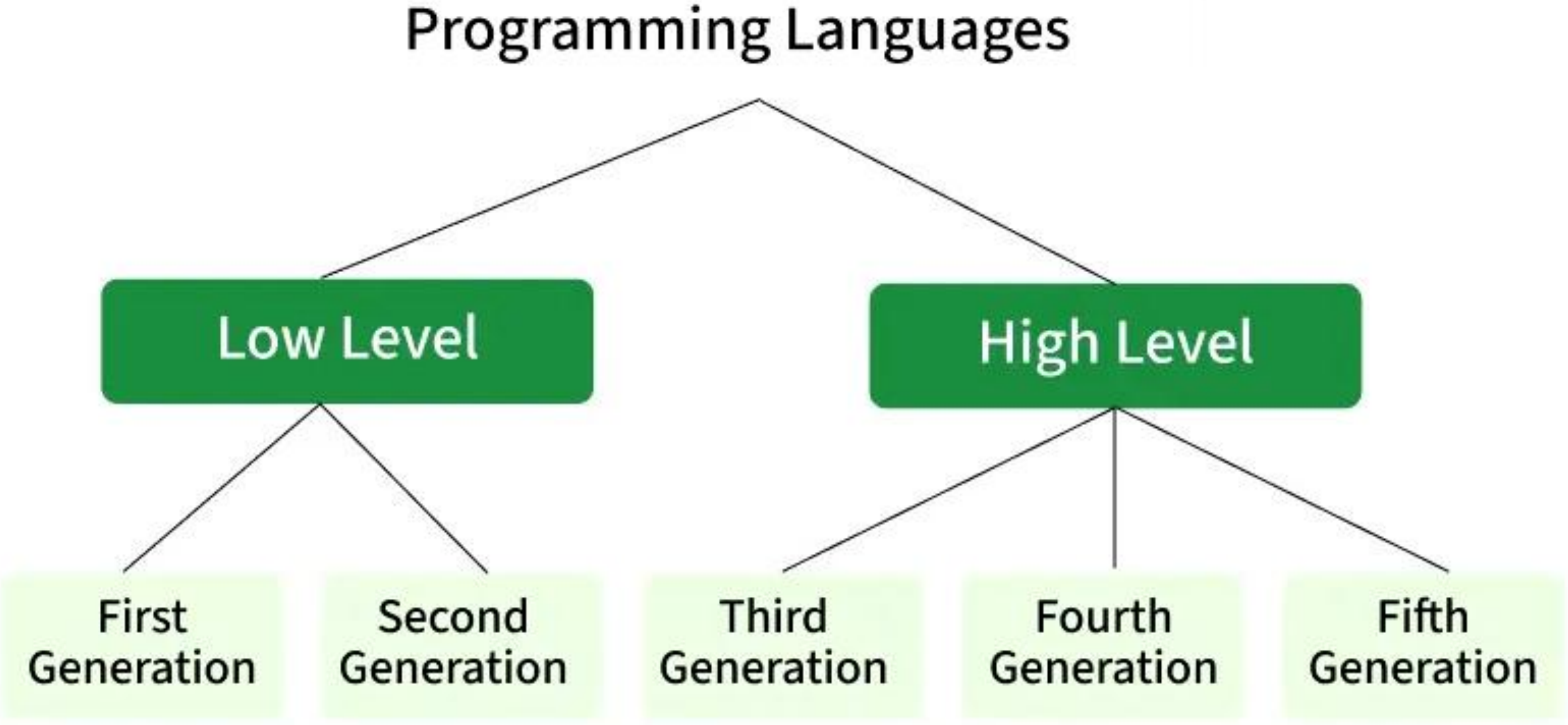
# Programming Languages

- 20. MS DOS:
  - It is one of the most popular operating systems developed by Microsoft.

Welcome
Welcome to DOS and program. SELECT
will install DOS disk or diskette
If you install te, the number of
blank diskettes the type and
capacity of you
Drive Type Diskettes
5.25-Inch 25 (360KB)
5.25-Inch 25 (360KB)
3.5-Inch (720KB)
3.5-Inch (720KB)
MS
DOS

# Programming Languages

- Fourth Generation Language (4th GL):
  - The third generation language needed a large number of codes for typical commercial system.
  - It is time consuming to debug, and the modification of complex system is very difficult.
  - It is a 4th generation language developed by the software vendors in various application tools offering further improvement in productivity in programing.
  - A fourth generation programming language is designed with a specific purpose in mind, such as the development of commercial business software.
  - All 4GLs are designed to reduce programming effort, the time it takes to develop software, and the cost of software development.



Programming Languages

Low Level

High Level

First Generation

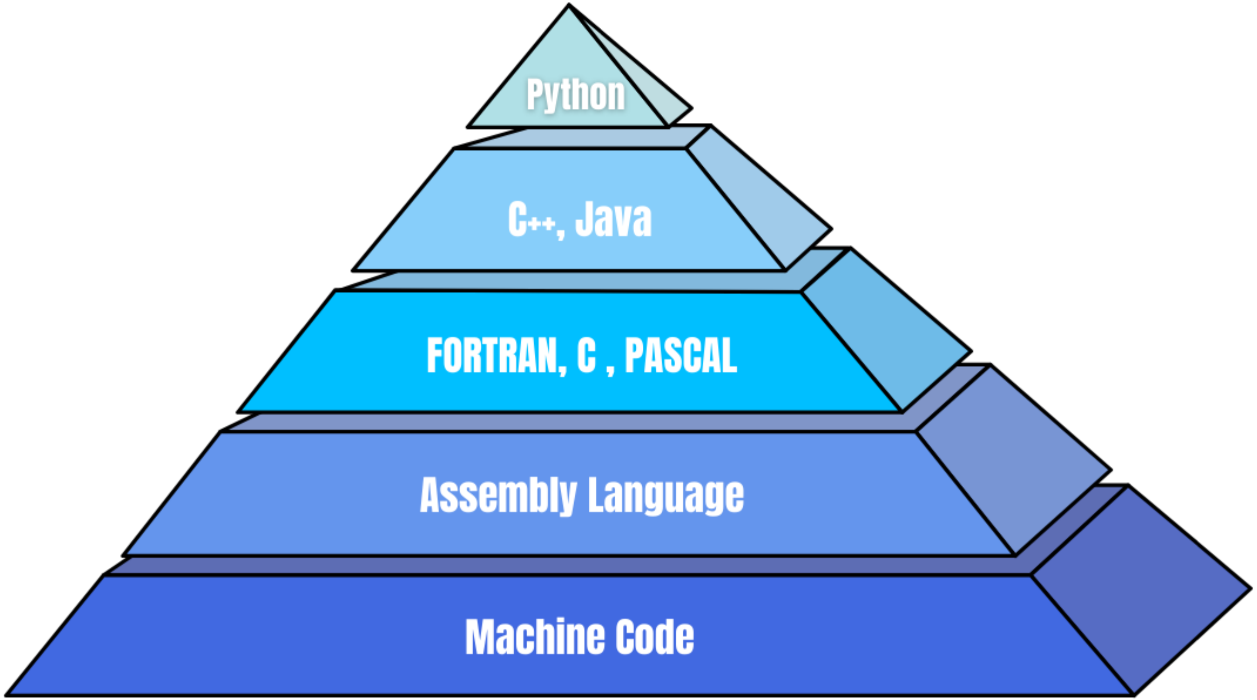
Second Generation

Third Generation

Fourth Generation

Fifth Generation

High Level Language



Python

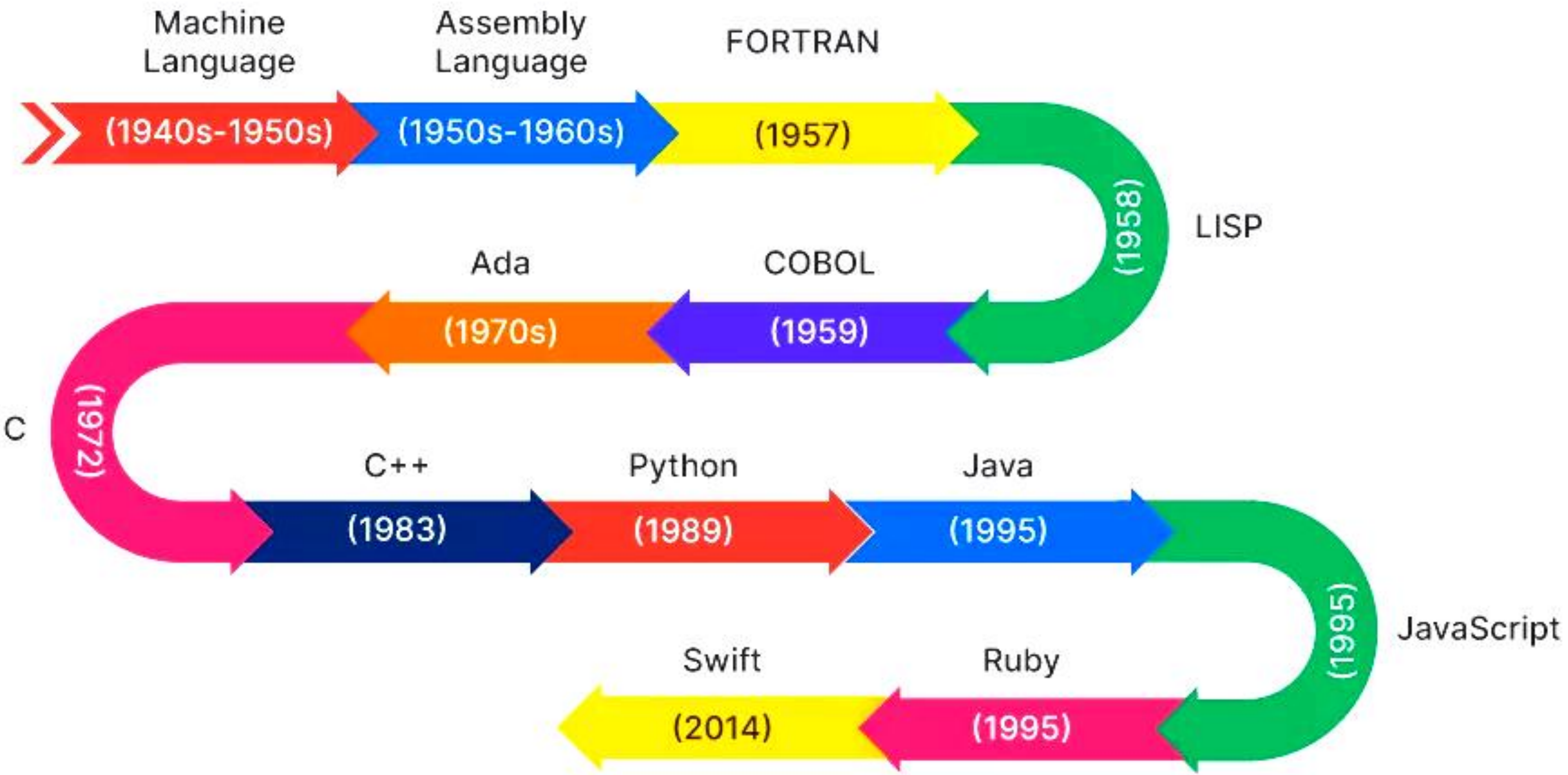
C++, Java

FORTRAN, C , PASCAL

Assembly Language

Machine Code

Low Level Language


Machine Language
(1940s-1950s)
Assembly Language
(1950s-1960s)
FORTRAN
(1957)
LISP
(1958)
COBOL
(1959)
Ada
(1970s)
C
(1972)
C++
(1983)
Python
(1989)
Java
(1995)
JavaScript
(1995)
Ruby
(1995)
Swift
(2014)

# timeline of programming languages:

- Lisp – 1958
- C – 1972
- C++ – 1980
- Objective-C – 1986
- Erlang – 1986
- Perl – 1987
- Haskell – 1990
- Python – 1990
- HTML – 1991
- Ruby – 1995
- PHP – 1995
- JavaScript – 1995
- Java – 1995
- CSS – 1996
- C# – 2001
- Scala – 2003
- Clojure – 2007
- Go – 2009
- Node.js – 2009
- Rust – 2010
- Kotlin – 2011
- TypeScript – 2012
- Swift – 2015

THANK
YOU