# CHAPTER 06

# डिजाइन टूल्स तथा प्रोग्रामिंग भाषाएँ

# (Design Tools and Programming Languages)

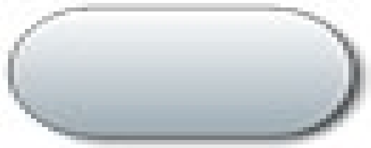
# (Design Tools)

- किसी प्रोग्राम को लिखने से पहले उसके अर्न्तगत होने वाले इनपुट, आउटपुट, डेटा के प्रवाह तथा लॉजीक का निर्धारण करना होता है। इसके लिए हमें डिजाइन टुल्स की आवश्यकता होती है। ये डिजाइन टुल्स निम्नलिखित है :

- **DFD-Data Flow Diagram**
- **DFD** किसी प्रोसेस या सिस्टम में डेटा के प्रवाह का चित्रात्मक प्रदर्शन है। इसमें सिस्टम में कंट्रोल का प्रवाह न दिखाकर डेटा का प्रवाह को चित्रित करते हैं।

- **DFD** बनाने के लिए कुछ चिह्नें और संकेतकों **(Symbols and Notations)** का उपयोग होता है। ये संकेतक निम्नलिखित हैं:
- **(Data Flow):** इसे तीर के चिह्न वाली रेखा से प्रदर्शित करते हैं। यह सिस्टम में डेटा के प्रवाह की दिशा बताता है।
- **(Process):** यह आने वाली डेटा के प्रवाह **(Incoming data flow)** को जाने वाली डेटा के प्रवाह **(outgoing data flow)** में बदल देता है। इसके अन्दर प्रोसेस के निर्देश होते हैं।
- **(Decision):** यह लॉजिकल प्रक्रिया को प्रदर्शित करता है जिसका परिणाम हाँ **(yes)** या ना **(No)** होता है। **DFD** में इसे से दर्शाया जाता है।

- **(Connector)** : विशाल प्रोग्राम के एक पृष्ठ से अधिक के फ्लोचार्ट को कनेक्टर के द्वारा जोड़ा जाता है।
- **(Input/output)** : यह प्रोग्राम में इनपुट तथा आउटपुट को दर्शाता ळें
- **(Data Store)** : यह डेटा के संग्रह को दर्शाता है।

| 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 ouptut. |
| | Process | A rectangle represents a process. |
| | Decision | A diamond indicates a decision. |

# Algorithm

- कम्प्यूटर की सहायता से किसी भी कार्य को सम्पन्न करने के लिए प्रोग्राम या निर्देशों के समूह की आवश्यकता होती है। प्रोग्राम लिखने के लिए हमें एक–एक कर बताना होता है, यह कैसे संपन्न होगा या प्रोग्राम किस लॉजिक पर कार्य करेगा यहाँ पर हमें कम्प्यूटर एल्गोरिदम की जरूरत होती है। एल्गोरिदम किसी कम्प्यूटर प्रोग्राम को पूरा करने के लिए बुनियादी तकनीक है। यह निर्देशों का समूह है जो कार्य सम्पन्न होने में सहायक है। एल्गोरिदम में कोई गलती **(error)** होने पर गलत रिजल्ट प्राप्त होता है, जिसे **Logic error** कहते हैं।

**Step 1:** Start

**Step 2:** Create a variable to receive the user's email address

**Step 3:** Clear the variable in case it's not empty

**Step 4:** Ask the user for an email address

**Step 5:** Store the response in the variable

**Step 6:** Check the stored response to see if it is a valid email address

**Step 7:** Not valid? Go back to Step 3.

**Step 8:** End

## (Flowchart)

- फ्लोचार्ट एल्गोरिदम या प्रोसेस का चित्रात्मक प्रदर्शन है| फ्लोचार्ट प्रोसेस या प्रोग्राम का विश्लेषण, डिजाइन करने, डाक्यूमेंट बनाने तथा प्रबंधन में उपयोग होता है| यह भी **DFD** के तरह चिह्नों तथा संकेतकों का प्रयोग कर बनाया जाता है|

## (Pseudocode)

- इसे प्रोग्राम डिजाइन लैंग्वेज **(PDL)** भी कहा जाता है जो फ्लोचार्ट का एक विकल्प है| सुडोकोड में लॉजिक को अंग्रेजी की तरह लिखा जाता है| बहुत सारे प्रोग्रामर सुडोकोड को वरीयता देते हैं क्योंकि इसमें परिवर्तन करना आसान है|

# Programming Languages

- प्रोग्रामिंग भाषा कम्प्यूटर को निर्देश देने तथा इच्छानुसार कार्य करवाने का एक माध्यम है। यह एक कृत्रिम भाषा है जिसे कम्प्यूटर को एक निश्चित क्रमानुसार चलाने या काम करने के लिए प्रयोग में लाया जाता है। यह की–बोर्ड, सिंवाल्य का एक सेट और स्टेटमेंट कन्स्ट्रक्ट करने के लिए नियमों का एक सेट है, जिसके द्वारा मानव कम्प्यूटर द्वारा निष्पादित किये जाने वाले अनुदेशों को संप्रेषित कर सकता है। मुख्यत: प्रोग्रामिंग भाषा दो प्रकार के होते हैं :
- निम्न स्तरीय भाषा **(Low Level Language)**
- उच्च स्तरीय भाषा **(High Level Language)**

# Machine Language

- यह कम्प्यूटर की आधारभूत भाषा है। यह केवल 0 और 1 दो अंकों के प्रयोग से निर्मित श्रृंखला अर्थात् बाइनरी कोड से लिखी जाती है। यह एकमात्र कम्प्यूटर प्रोग्रामिंग भाषा है जो कि कम्प्यूटर द्वारा सीधे–सीधे समझी जाती है। इसे किसी अनुवादक **(Translator)** प्रोग्राम की आवश्यकता नहीं करनी होती है। इसे कम्प्यूटर का मशीन संकेत भी कहते हैं। प्रोग्रामिंग के शुरूआत के समय प्रोग्राम इसके प्रयोग से लिखे जाते थे।

# Machine Language

- मशीन भाषा में प्रत्येक निर्देश के दो भाग होते हैं पहला ऑपरेशन को या ऑपकोड **(Operation code or opcode)** और दूसरा लोकेशन कोड या ऑपरेण्ड **(Location code or oprand)A** ऑपकोड कम्प्यूटर को यह बताता है कि क्या करना है और ऑपरेण्ड यह बताता है कि आँकड़ें कहाँ से प्राप्त करना है, कहाँ संग्रहित करना है।
- मशीन भाषा में प्रोग्राम लिखना एक मुश्किल कार्य है। इस भाषा में प्रोग्राम लिखने के लिए प्रोग्रामर को मशीन निर्देशों या अनेकों संकेत संख्या के रूप में याद करना पड़ता है। इसमें गलती होने की संभावना अत्यधिक है तथा यह अत्यधिक समय लगने वाला कार्य है।

# Assembly Language

- मशीन भाषा में प्रोग्राम लिखने में आनेवाली कठिनाइयों को दूर करने के लिए एक अन्य असेम्बली भाषा का निर्माण किया गया। इसमें बाइनरी कोड (0 या 1) का इस्तेमाल न कर अक्षर अथवा चिह्नों का प्रयोग किया जाता है जिसे सिम्बॉल **(Symbol)** भाषा कहते हैं। इसमें न्यूमोनिक कोड का प्रयोग किया गया जिन्हें याद रखना आसान है। जैसे **LDA (load), TRAN (Translation), ADD (Adding)** तथा **SUB (Subtraction)** के लिए इत्यादि।

# ADDTION PROGRAM IN ASSEMBLY

```
.MODEL SMALL
.STACK 64
.DATA
.CODE

START: MOV AX, @DATA
       MOV DX,AX
       MOV AL,01H
       MOV BL,02H
       ADD AL,BL
       MOV SI,3000H
       MOV [SI], AL
       MOV AH,4CH
       INT 21H
       END START
       .END
```

# Assembler

- Assembler is the language translator that converts assembly language code in to the object code (machine code).

- इनमें से प्रत्येक के लिए एक मशीन कोड भी निर्धारित किया गया, पर असेम्बली कोड से मशीन कोड या ऑब्जेक्ट कोड में परिवर्तन का काम एक प्रोग्राम के द्वारा किया जाता है। जिसे असेम्बलर **(Assembler)** कहा गया।
- अतः असेम्बली भाषा में प्रोग्राम लिखना अपेक्षाकृत अधिक सरल तथा समय की बचत करने वाला है। इसमें गलतियों को सरलता से ढूँढ़ जा सकता है।
- उच्च स्तरीय भाषा
- **High Level Language**

- उच्च स्तरीय भाषा कम्प्यूटर में प्रयोग की जाने वाली वह भाषा है जिसमें अंग्रेजी अक्षरों, संख्याओं एवं चिह्नों का प्रयोग कर प्रोग्राम लिखा जाता है। यह मषीन पर निर्भर **(Machoine dependent)** नहीं हैं। इन प्रोग्रामिंग भाषाओं को कार्यानुसार चार वर्गों में विभाजित किया गया है :

1. **वैज्ञानिक प्रोग्रामिंग भाषाएँ (Scientific Programming Languages) :** इनका प्रयोग मुख्यत: वैज्ञानिक कार्यों के लिए होता है; जैसे– अल्गोल, बेसिक, फोरट्रॉन, पास्कल आदि।

| | |
|---|---|
| Algol | International Group |
| BCPL | Martin Richards |
| B | Ken Thomson |
| Traditional C | Dennis Ritchie |
| K&R C | kernighan & Ritchie |
| ANSI C | ANSI Commitee |
| ANSI/ISO C | ISO Commitee |
| C99 | Standerd Commitee |

2. **(Commercial Programming Languages)** :

व्यापार संबंधित कार्यों, जैसे– बही खाता, रोजानामचा, स्टॉक आदि का लेखा–जोखा आदि के लिए इनका उपयोग किया जाता है। जैसे– **PL1**, कोबोल, डीबेस आदि।

3. **(Special Purpose Programming Languages)** :

ये भाषायें विभिन्न कार्यों को विशेष क्षमता के साथ करने के लिए प्रयोग की जाती है। जैसे– **AP360**, लोगों आदि।

4. **(Multipurpose Programming Languages)** : जो भाषायें समान रूप से विभिन्न प्रकार के कार्यों को करने की क्षमता रखती है, उन्हें बहुउद्देषीय भाषाएँ कहते हैं। जैसे– बेसिक, पास्कल, **PL1** आदि।

## कुछ उच्चस्तरीय भाषाएँ

1. **(FORTRAN-Formula Translation)** : इसका विकास सन् 1957 में **IBM 704** कम्प्यूटर के लिए जॉन बेकस के नेतृत्व में हुआ था। यह गणितीय कार्यों, सूत्रों तथा गणनाओं को करने में पूर्णतः सक्षम है। इसका उपयोग वैज्ञानिकों तथा इंजीनियरों द्वारा किया जाता है। यह प्रोग्रामिंग के लिए विकसित की गई सर्वप्रथम भाषा है।

2. **(ALGOL-Algorithmic Language)** : अल्गॉल का विकास सन् 1958 में अल्गॉल 58 के नाम से हुआ था। 1960 में इसमें थोड़ा परिवर्तन कर अल्गॉल 60 लाया गया। इसका उपयोग वैज्ञानिक और इंजीनियरिंग उद्देष्य से किया जाता है, तथा यह गणितीय गणना करने में पूर्ण रूप से सक्षमहै ।

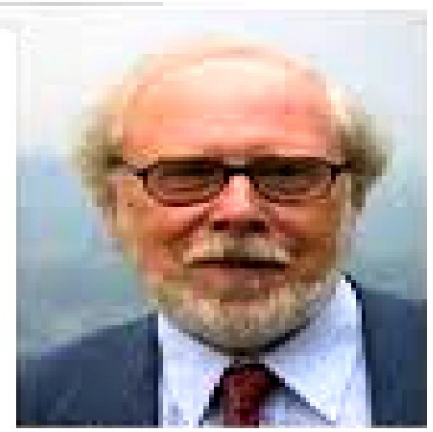
**3 -PL1 (Programming Language 1)** : **PL1** का विकास सन् 1960 में **IBM** के द्वारा व्यावसायिक तथा वैज्ञानिक अनुप्रयोगों के लिए किया गया था। यह एक सफल प्रोग्रामिंग भाषा है, सिवाय इसके कि यह बहुउद्देषीय प्रसाधन देने के कारण छोटे मषीनों के लिए बहुत बड़ा है।

5 . **(Pascal)** : सन् 1971 में निकलॉस विर्थ द्वारा पास्कल भाषा का विकास किया गया। इस समय अन्य भाषाओं में जो कमी थी उसे पास्कल में प्रदान करने की कोशिश की गई। इस भाषा में संरचित प्रोग्रामिंग तकनीकों **(Structured Programming technique)** की सुविधा प्रदान की गई। इसे विससित करने का मूल प्रयोजन छात्रों को प्रोग्रामिंग के मूलभूत तत्वों से अवगत कराना था। यह शिक्षण कार्यों के लिए विकसित किया गया था।ए

## History of Pascal

- Developed in the early 1970s by Niklaus Wirth at
- The Technical University in Zurich, Switzerland

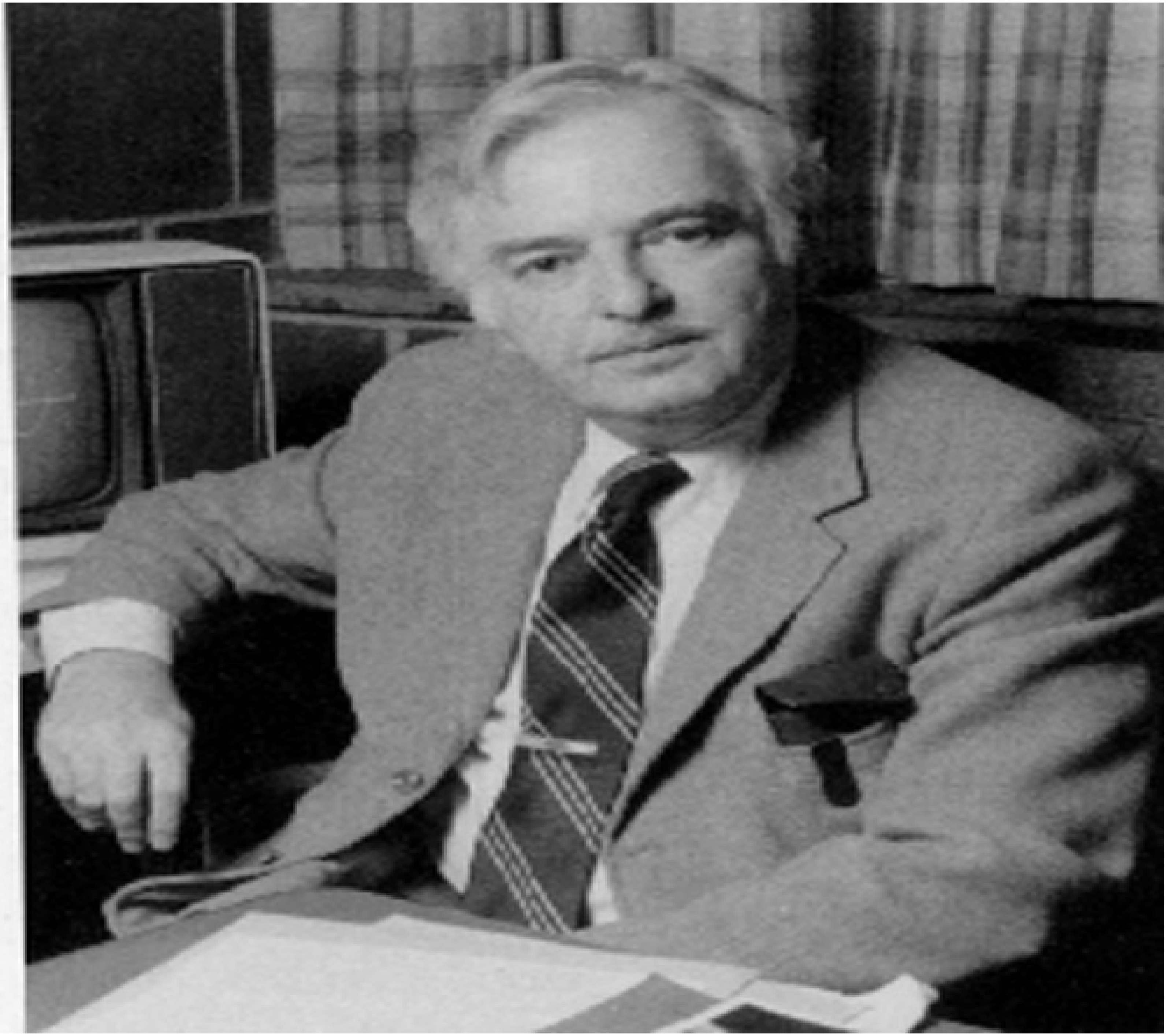
Pascal



Niklaus Wirth

## 6 . (BASIC-Beginner's ALL-purpose Symbolic Instruction Code) :

- 1964 में जॉन जार्ज कैमी और थॉमस यूजीन कुर्टज ने बेसिक भाषा का विकास किया।
- नये प्रोग्रामरों के लिए यह सरल तथा शक्तिशाली भाषा है। यह इनट्रैक्टिव उपयोग के लिए डिजाइन किया गया है।
- इसका उपयोग वैज्ञानिकों तथा व्यवसायियों दोनों द्वारा किया जाता है।



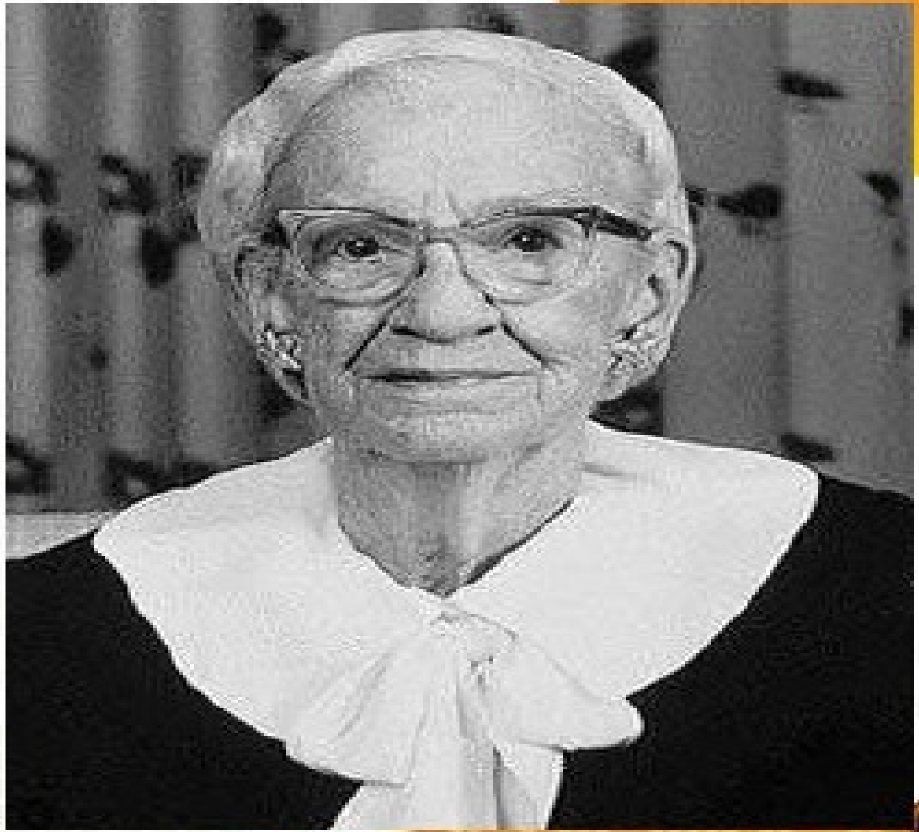
John George Kemeny

## 7 (COBOL-Common Business Oriented Language) :

इस भाषा का विकास व्यावसायिक हितों के लिए किया गया। इस भाषा में लिखे गये वाक्यों के समूह को पैराग्राफ कहते हैं। सभी पैराग्राफ मिलकर एक सेक्षन बनाते हैं और सेक्षनों से मिलकर डिविजन बनता है। कोबोल में गणितीय शब्दावली के लिए **ADD, SUBTRACT** और **MULTIPLY** का उपयोग होता है। यह अंग्रेजी भाषा की तरह है तथा इसमें सर्वाधिक उपयुक्त डाक्यूमेंटेसन संभव है।

## Grace Murray Hopper

- Invented the first Computer Language Cobol



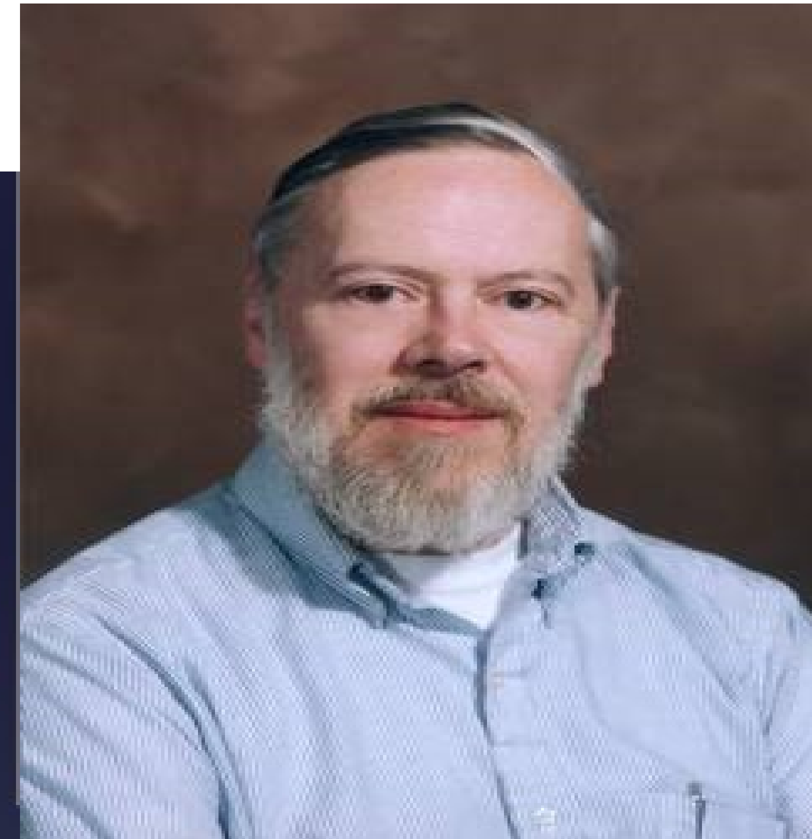
**8 .(LOGO) :** इस भाषा का विकास कम्प्यूटर षिक्षा को सरल बनाने हेतु किया गया। इसमें चित्रण इतना सरल है किय छोटे बच्चे भी चित्रण कर सकते हैं। लोगों भाषा में चित्रण के लिए एक विशेष प्रकार की त्रिकोणाकार आकृति होती है जिसे टरटल **(turtle)** कहते हैं। यह टरटल निर्देशों द्वारा किसी भी तरफ घूम सकता है। जब टरटल चलता है तो पीछे एक रेखा बनाता जाता है जिससे अनेक प्रकार के चित्रों को सरलता से बनाया जा सकता है।

9 .**C Language**: सी प्रोग्रामिंग भाषा 1970 के दषक में डेनिस रिची द्वारा विकसित किया गया था। सी कम्पाईलर सारे मषीनों/कम्प्यूटरों पर कार्य करने में सक्षम है। अतः इसका उपयोग बहुत ही व्यापक रूप से होता है। यह सामान्य उद्देषीय **(General Purpose)** प्रोग्रामिंग भाषा है। इसका डिजाइन तो सिस्टम सॉफ्टवेयर बनाने के लिए हुआ था, पर इसका उपयोग अनुप्रयोग सॉफ्टवेयर **(Application Software)** बनाने में भी काफी होता है।

# Dennis Ritchie

Američan, 1941

*"I am not now, nor have I ever been, a member of the demigodic party."*



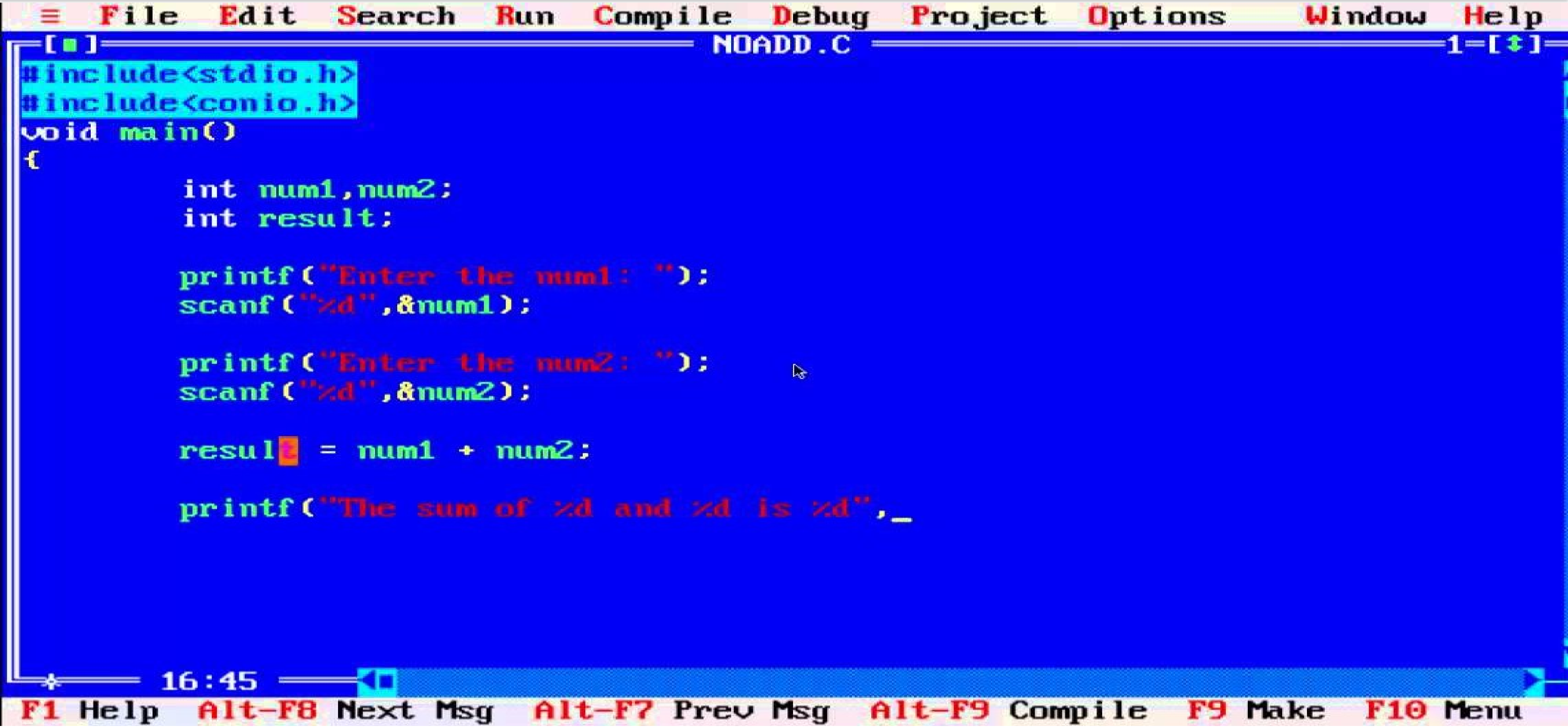
The C Programming Language

Harvard

Bell Labs

UNIX

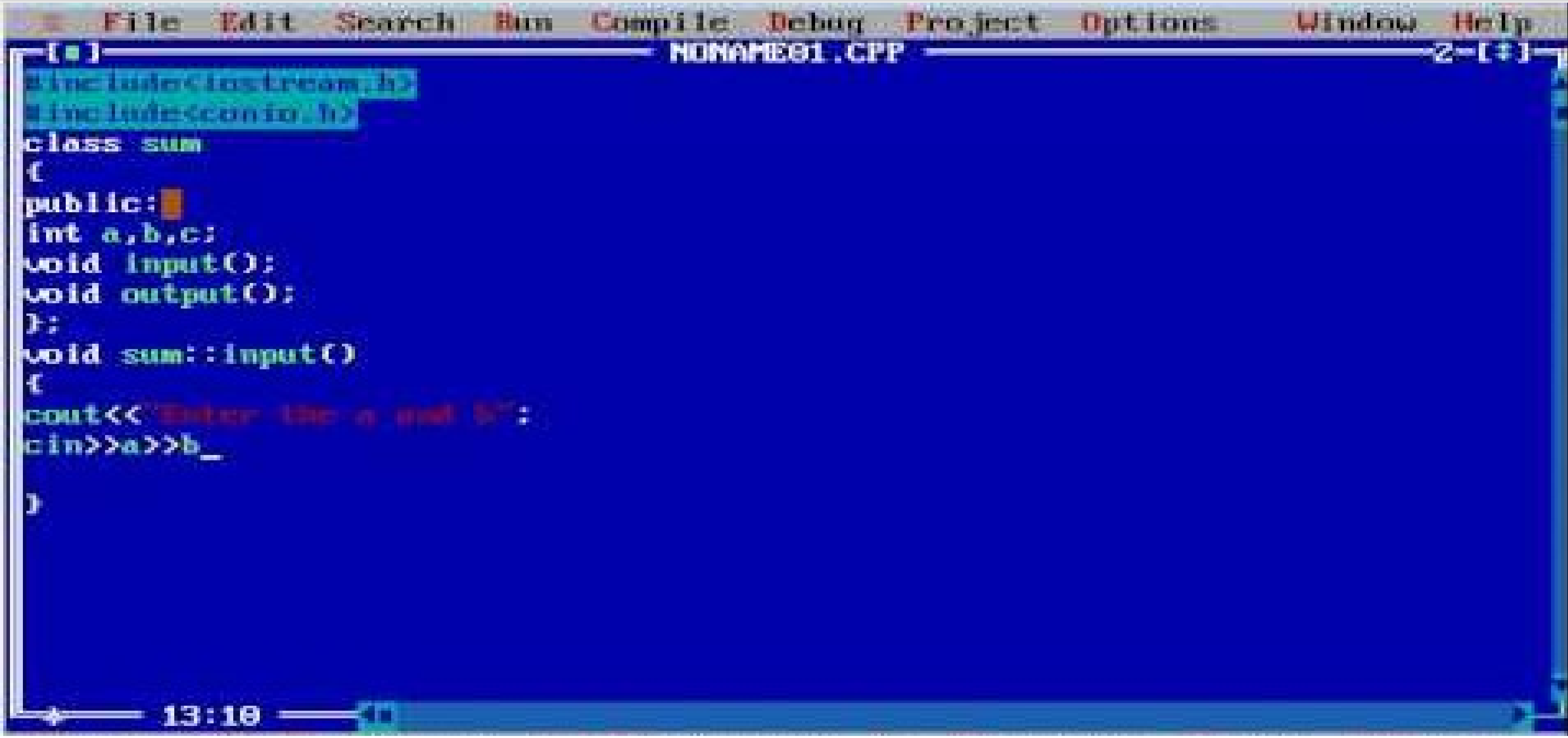
# C Program


≡ File Edit Search Run Compile Debug Project Options Window Help
[■] NOADD.C 1=[↕]
#include<stdio.h>
#include<conio.h>
void main()
{
int num1,num2;
int result;
printf("Enter the num1: ");
scanf("%d",&num1);
printf("Enter the num2: ");
scanf("%d",&num2);
result = num1 + num2;
printf("The sum of %d and %d is %d",_
16:45
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu

10 .**(C++)** : यह सिस्टम प्रोग्रामिंग के साथ–साथ सामान्य उद्देष्य **(General purpose)** प्रोग्रामिंग भाषा है। यह सी से थोड़ा बेहतर है तथा ऑब्जेक्ट उन्मुख **(Object oriented)** प्रोग्रामिंग भाषा है। **C++, C** की अपेक्षा कठिन प्रोग्रामिंग भाषा है।

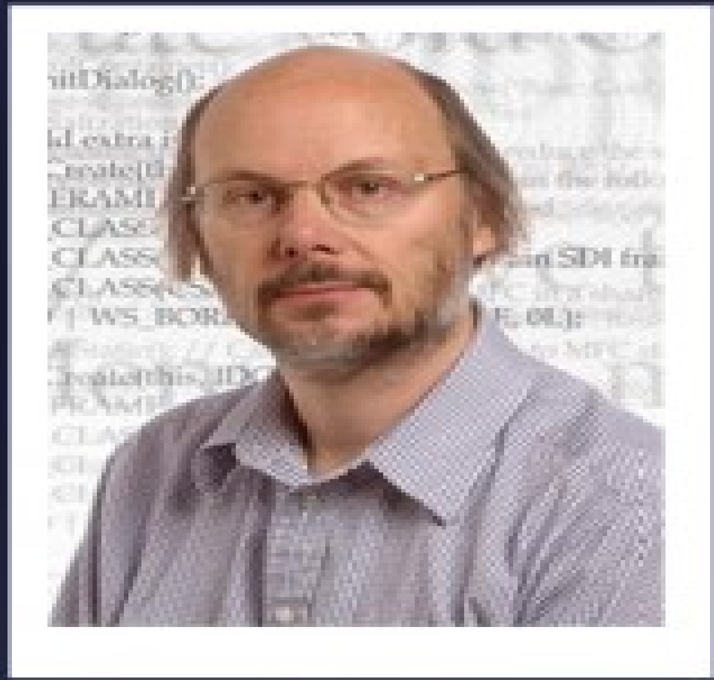
11 .**(COMAL-Common Algorithmic Language)** : यह सन् 1973 में डेनमार्क के बेनेडिक्ट लॉफस्टड और ब्रौज क्रिस्टनसन के द्वारा विकास किया गया था। कोमल, बेसिक और पास्कल भाषा का मिला–जुला रूप है जो छात्रों को षिक्षा देने के लिए डिजाइन किया गया था।

# ADDITION PROGRAM IN C++



BJARNE STROUSTRUP

Dán, 1950



Texas A&M U

# The C++ Programming Language

12 .**(Prolog)** : यह प्रोग्रामिंग इन लॉजिक **(Programming in Logic)** का संक्षिप्त है। यह डाटा स्ट्रक्चर का धनी संग्रह है। इसका उपयोग बुद्धिमान सिस्टम **(Intelligent System)]** विशेषज्ञ सिस्टम **(Expert System)** को विकसित करने में किया जाता है, जो तार्किक और भावनात्मक प्रोग्रामिंग में संभव है।

13 .**(RPG-Report Program Generator)** : यह 1961 में **IBM** द्वारा विकसित किया गया था। यह व्यावसायिक अनुप्रयोगों के लिए एक प्रोग्रामिंग भाषा है जो रिपोर्ट बनाकर देता है।

14 .**(C Sharp)C#** : सी शार्प को ब्रु भी लिखा जाता है। **C#** एक कम्प्यूटर भाषा है, जो माइक्रोसॉफ्ट द्वारा विकसित की गई है। यह एक बहु कार्यात्मक तथा ऑब्जेक्ट ओरिएन्टेड प्रोग्रामिंग भाषा है।

15 .**(Java)** : जावा मूल रूप से सन माइक्रोसिस्टम द्वारा विकसित किया गया है और 1995 में इसे जारी किया गया। जावा, सिन्टैक्स सी तथा **C** का डेरिभेटीव **(Derivative)** है। यह ऑब्जेक्ट ओरिएन्टेड भाषा है। यह सामान्य उद्देष्यीय प्रोग्रामिंग भाषा है जो विभिन्न विषेषताओं के कारण इंटरनेट या वर्ल्ड वाइड वेव के लिए उपयुक्त भाषा है।

# James Gosling

- James Gosling is generally credited as the inventor of the Java programming language

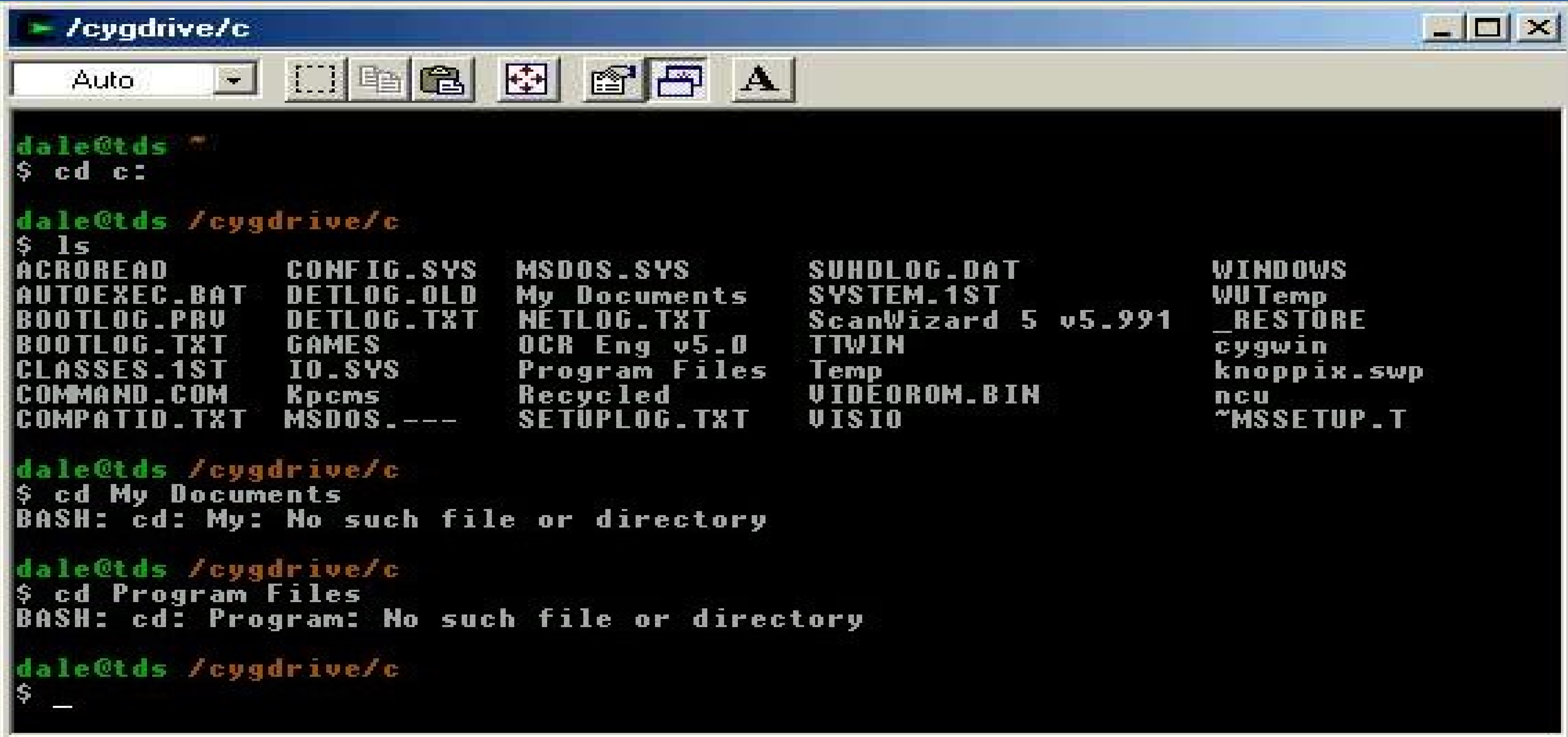
- He was the first designer of Java and implemented its original compiler and virtual machine

- He is also known as the Father of Java.

2 .**(Shell) :** शेल कमांड भाषा यूनिक्स ऑपरेटिंग सिस्टम के साथ प्रयुक्त होता है। यूनिक्स का अधिकतर वेब सर्वर या सर्वर में होता है।

3 .**(MS-DOS-Microsoft Disc Operating System) :** यह **IBM** के साथ प्रयुक्त होता है। इसके साथ अधिकतर डेटाबेस पैकेज डीबेस प्रयुक्त होता है।

# Shell Command in Linux



# MS - DOS

```
Displays a list of files and subdirectories in a directory.

DIR [drive:][path][filename] [/P] [/W] [/A[[:]attribs]] [/O[[:]sortord]]
    [/S] [/B] [/L] [/C[H]]

  [drive:][path][filename]   Specifies drive, directory, and/or files to list.
  /P       Pauses after each screenful of information.
  /W       Uses wide list format.
  /A       Displays files with specified attributes.
  attribs    D  Directories    R  Read-only files         H  Hidden files
             S  System files   A  Files ready to archive  -  Prefix meaning "not"
  /O       List by files in sorted order.
  sortord    N  By name (alphabetic)       S  By size (smallest first)
             E  By extension (alphabetic)  D  By date & time (earliest first)
             G  Group directories first    -  Prefix to reverse order
             C  By compression ratio (smallest first)
  /S       Displays files in specified directory and all subdirectories.
  /B       Uses bare format (no heading information or summary).
  /L       Uses lowercase.
  /C[H]    Displays file compression ratio; /CH uses host allocation unit size.

Switches may be preset in the DIRCMD environment variable.  Override
preset switches by prefixing any switch with - (hyphen)--for example, /-W.

C:\>_
```

- चौथी पीढ़ी की भाषा
- **4th GL**
- तीसरी पीढ़ी की भाषाओं में प्रोग्राम लिखने के लिए बहुत सारे कोड लिखने होते हैं। इनमें त्रुटि ढूँढ़ना तथा कोई परिवर्तन करना कठिन होता है। परन्तु 4 थी पीढ़ी की भाषा/**4th GL** में निर्देशों की संख्या कम होती है अतः प्रोग्राम लिखना आसान होता है।