



# **SNS COLLEGE OF TECHNOLOGY**

**Coimbatore-35  
An Autonomous Institution**

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A+' Grade  
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai



## **DEPARTMENT OF INFORMATION TECHNOLOGY**

### **23ITT101-PROGRAMMING IN C AND DATA STRUCTURES**

**I YEAR - II SEM**

#### **UNIT 1 – INTRODUCTION TO C**

#### **TOPIC 4 – Introduction to C Programming & Fundamental Rules**



# INTRODUCTION TO C PROGRAMMING



## □ History:

- C was originally developed in the 1970s, by **Dennis Ritchie** at **Bell Laboratories**.
- C is a High level , general – purpose **structured** programming language.
- It allows software developers to develop programs without worrying about the hardware platforms where they will be implemented.
- **Regulated by** American National Standards Institute (ANSI) & International Standards Organization (ISO).



# INTRODUCTION TO C PROGRAMMING

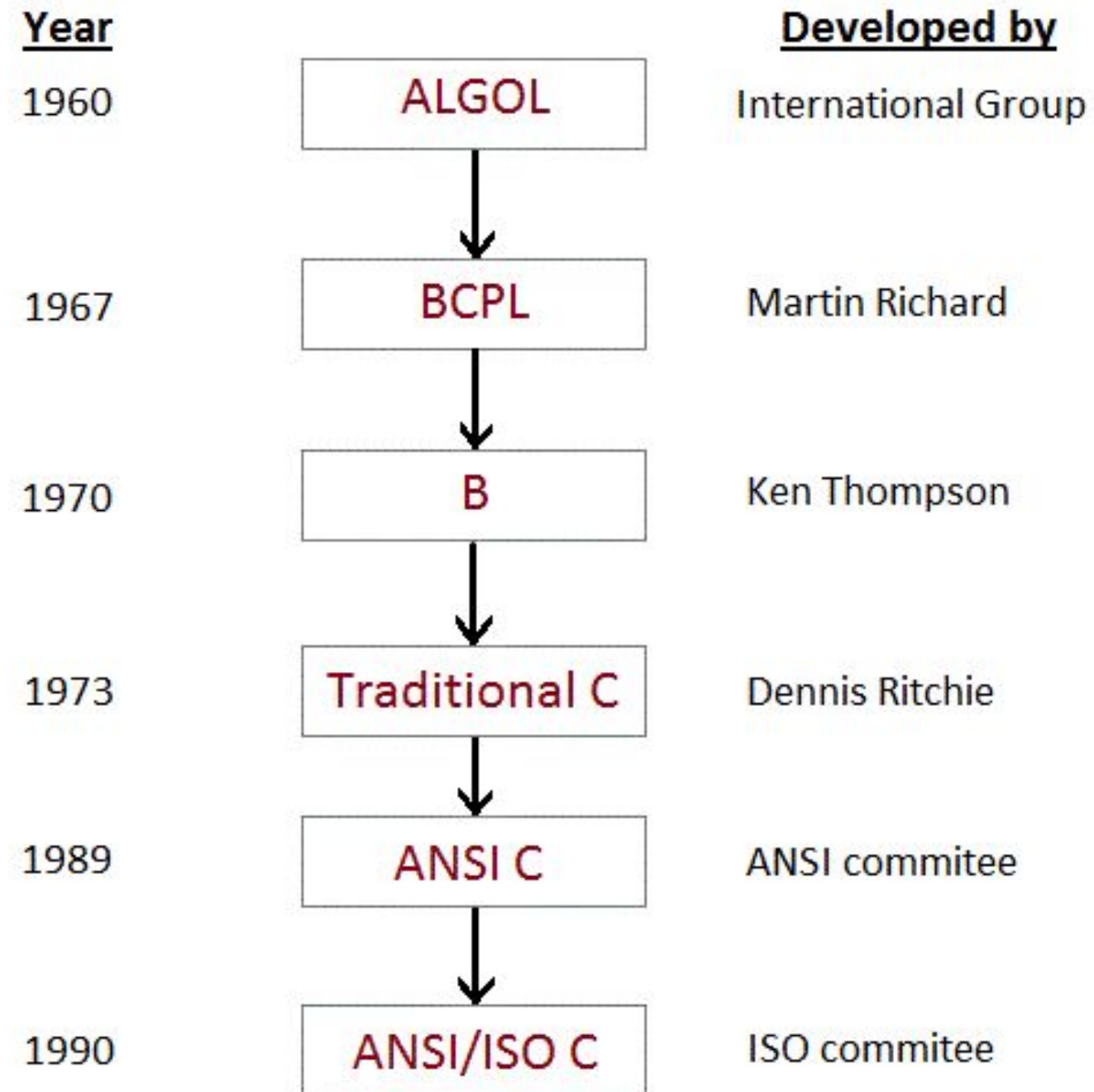


## □ History:

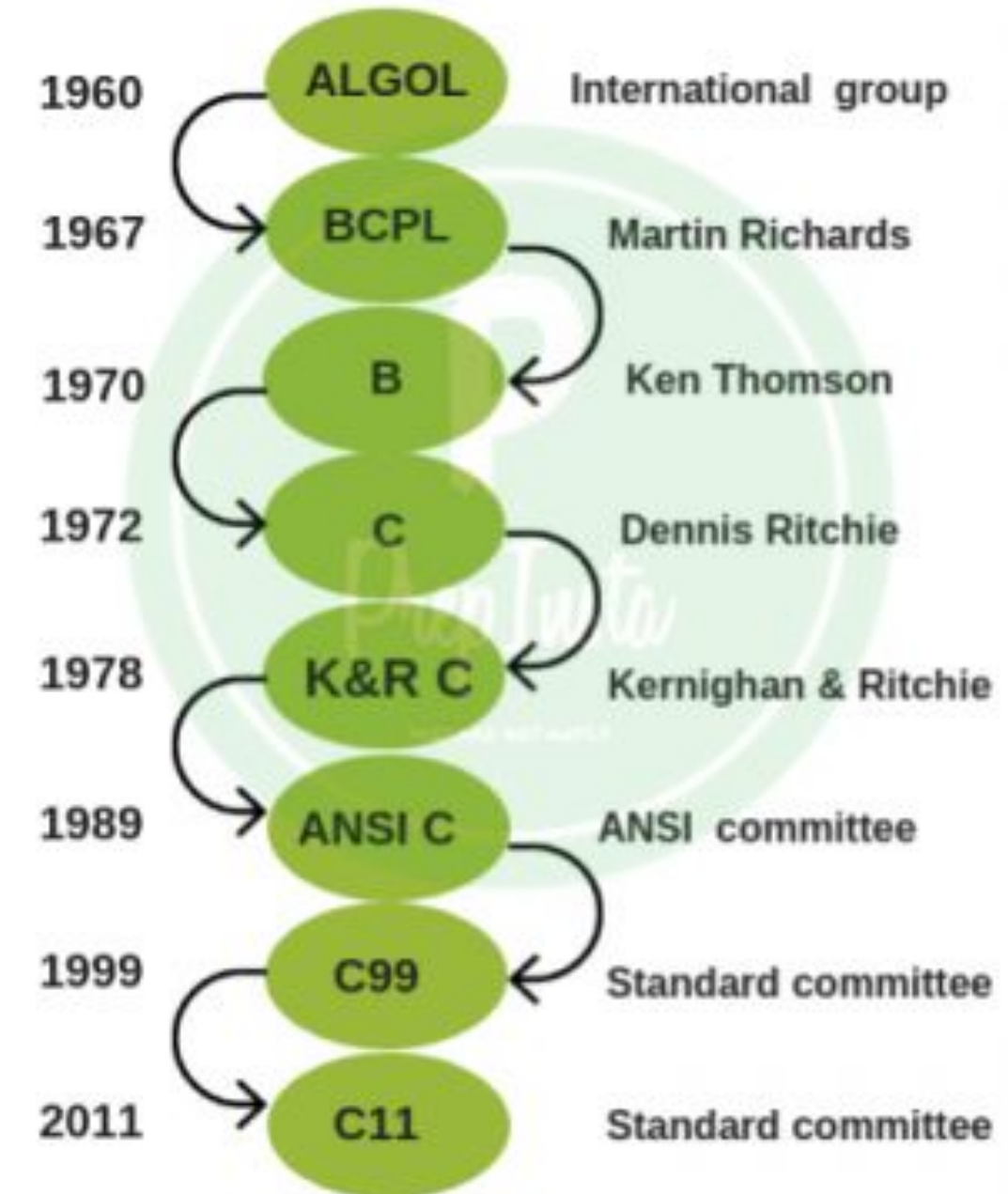
- Bell Labs developed C language based on “Basic Combined Programming Language” (BCPL).
- Instructions of C consists of terms that are very closely same to **algebraic expressions**, consisting of certain **English keywords** such as if, else, for ,do and while.
- C contains certain additional features that allows it to be used at a lower level, acting as **bridge** between machine language and the high level languages.
- This allows C to be used for **system programming** as well as for **applications programming**
- C Supports Both the low-level and High-Level programming features.



# HISTORY OF C DEVELOPMENT



## History of C







# IMPORTANCE & FEATURES OF C



## □ ROBUST LANGUAGE: (STRONG)

- C is a robust language, whose **rich set of built-in functions** and operators can be used to write any complex program.

## □ EFFICIENT AND FAST:

- Programs written in C are efficient and fast.
- This is due to its **variety of data types** and powerful operators.

## □ BUILT-IN FUNCTIONS

- There are only **32 keywords in ANSI C** and its strength lies in its built-in functions.
- Several standard functions are available which can be used for developing programs.

## □ PORTABILITY:

- C is highly portable.
- This means that C programs **written for one computer can be run on another** with little or no modification.

## □ STRUCTURED PROGRAMMING:

- C language is well suited for structured programming, thus requiring the user to think of a problem in terms of function **modules or blocks**.
- A proper collection of these modules would make a complete program.
- This “**modular structure**” makes program debugging, testing and maintenance easier.

## □ ABILITY TO EXTEND:

- Another important feature of C is its ability to extend itself.
- A C program is basically a collection of functions that are supported by the C library.
- We can continuously **add our own functions** to C library.



# FUNDAMENTAL RULES



- All C statements must **end with semi colon (;)**
- Semicolon (;) acts as a **terminator**.
- C is case sensitive i.e., upper case and lower case characters are different. (printf vs. PRINTF)
- Generally the **statements are typed in lower case**.
- All C statements can be written in one line or it can split into multiple lines.
- Braces must always **match upon pairs** i.e., every opening brace must have a matching closing brace ({...})
- **Every C program must contain a Main() function**
- Comments **can not** be nested. Example ( /\* welcome to c/\*programming\*/ \*/ )
- Blank spaces can be included between two words to improve the readability
- The variables must be **declared in the declaration section before they are used**



# SAMPLE PROGRAM 1: PRINTING A MESSAGE



```
main( )
{
/*.....printing begins.....*/
printf("I see, I remember");
/*.....printing ends.....*/
}
```

- The first line informs the system that the execution begins at this line.
- The **main( )** is a special function used by the C system to tell the computer where the program starts.
- **Every program must have exactly one main function.**
- If we use **more than one main** function, the compiler **cannot** understand which one marks the beginning of the program.
- The opening brace “{ ” in the second line marks the **beginning of the function main** and the closing brace “}” in the last line indicates the **end of the function**.
- The lines beginning with /\* and ending with \*/ are known as **comment lines**.
- Comment lines are **not executable statements** and therefore anything between /\* and \*/ is **ignored** by the compiler.



# SAMPLE PROGRAM 1: PRINTING A MESSAGE



```
main( )
{
/*.....printing begins.....*/
printf("I see, I remember");
/*.....printing ends.....*/
}
```

- The **printf( )** function is the only executable statement of the program.  
printf(“I see, I remember”);
- **printf** is a predefined standard C function for printing output.
- Predefined means that it is a function that has already been written and compiled, and linked together with our program at the time of linking.
- The **printf** function causes everything between the starting and the ending **quotation marks** to be printed out.
- In this case, the output will be:

I see, I remember





# SAMPLE PROGRAM 1: PRINTING A MESSAGE



```
main( )
{
/*.....printing begins.....*/
printf("I see, I remember");
/*.....printing ends.....*/
}
```

- C permits different forms of main statement. Following forms are allowed.
  - main()
  - **int main()**
  - **void main()**
  - int main(void)
- The keyword **void** means that the function does not return any information to the operating system
- The Keyword **int** means that the function returns an integer value to the operating system.
- When int is specified, the last statement in the program must be “return 0”.



## SAMPLE PROGRAM 2: PRINTING A MESSAGE



```

Including Header Files
#include<stdio.h>
#include<conio.h>
void main() ← main() Function Must Be There
{
  clrscr();
  printf("Welcome to DataFlair");
Single Line Comment → // helps to print the message "Welcome to DataFlair"
  getch(); ← Semicolon After Each Statement
}
Program Enclosed Within Curly Braces
```

- The first line states the inclusion of Header files.
- The **void main( )** is a special function used by the C system to tell the computer that the program starts and returns no value.
- Every program must have exactly one main function.
- If we use more than one main function, the compiler cannot understand which one marks the beginning of the program.
- The opening brace “{ ” in the fourth line marks the beginning of the function main and the closing brace “}” in the last line indicates the end of the function.



# SAMPLE PROGRAM 2: PRINTING A MESSAGE



```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    printf("Welcome to DataFlair");
    // helps to print the message "Welcome to DataFlair"
    getch();
}
```

Annotations in the image:

- Including Header Files (points to #include lines)
- main() Function Must Be There (points to void main())
- Program Enclosed Within Curly Braces (points to the curly braces)
- Single Line Comment (points to the // comment line)
- Semicolon After Each Statement (points to the semicolon after getch())

- The lines beginning with // are known as comment lines (Second Type of Comment).
- Comment lines are not executable statements and therefore anything starting with // is ignored by the compiler.
- getch() represents **getting the character from processing**.
- Clrscr() represents clear screen indication, which processes with **clearing the previous processed descriptive screen**.
- Statements end with Semicolon (;)