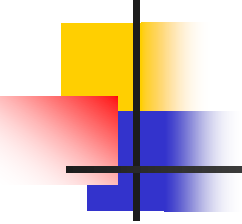




# COMPUTER

---

- A Computer is an electronic device designed to accept and store data, process them and produce meaningful results under the direction of detailed step by step instructions.

- 
- 
- Software : The software of the computer constitutes of the programs and instructions to be executed.
  - Hardware : Actual machine that includes the mechanical, electrical and electronic equipments.
  - The electronic circuits used in building the computer that execute the software is known as the Hardware of the computer.



# How Computer Works?

---

- A Computer by itself cannot do any manipulation as it does not have “Brain” or thinking power.
- It has to therefore be instructed to perform a task and supply with necessary information in detailed form in a ‘Language’ which computer can understand.
- Computer process information using the instructions from the user to produce the desired results.
- The set of instructions given to the computer is called a “Program” which directs the computer to function in the way specified by the instructions.



# Computer Languages

---

- The programmer makes use of a computer language to communicate with the computer, which being a machine, is receptive only to simple, unambiguous facts.
- A PL therefore uses a limited or restricted vocabulary.
- A PL consists of all symbols, characters and usage rules that permit people to communicate with computers.



# Computer Languages

---

- Low Level Languages
  - Machine Level Language
  - Assembly Language
- High Level Languages



# Machine Level Language

---

- First Generation of computer languages, used by Technicians who designed the computers.
- In ML, the instructions are written using 0's and 1's.
- The ML is the only one that the processor directly “understands”.
- An instruction prepared in any ML will have at least two parts:
  - “Command” or “Operation” which tells the computer what functions to perform.
  - “Operand” which tells the computer where to find or store the data and instructions that are to be manipulated.
- The number of operands in any instruction, and the option code varies from one computer to another.



# Disadvantages of ML

---

- Writing Programs in ML proved to be extremely difficult as well as tedious.
- Apart from remembering the dozen of code numbers for the commands in the machine's instruction set, the programmer had to keep track of the storage locations of data and instructions.
- The initial coding (writing the program) often took months.
- Error-checking and rectifications proved to be time-consuming.



# Assembly Language

---

- 2<sup>nd</sup> Generation of Languages which were essentially as tedious and detailed as the ML, but used abbreviations as commands in their instruction instead of 0's and 1's.
- i.e to ease the programmer's burden, mnemonic operation codes and symbolic addresses were developed in the early 1950's.
- The word mnemonic refers to a memory aid. Letter symbols or mnemonics were substituted for the numeric machine language operation codes.
- For egs: ADD, MOVE etc
- However the mnemonic codes used, vary from one computer to another, depending upon the make or models of the machine. Therefore, AL too is machine dependent.





# High Level Languages

---

- E.g COBOL, PASCAL, BASIC, etc.
- Easier to learn than AL & ML
- Require less time to write as the instructions are simple English like instructions.
- Provide better documentation.
- Easier to maintain i.e modifications and updating can be done without much difficulty.
- A Program written in HLL or AL is known as the “Source code” which has to be translated to ML that the computer understands.
- This translated version is known as the Object Code.
- Specialized software like compilers, interpreters, assemblers etc, are used to achieve translation.

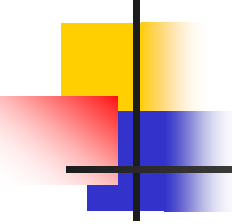


# What is C

---

- C is a programming language
- Developed in 1972 by Dennis Ritchie at AT&T Bell Labs
- C is a general-purpose programming language.
- C is a high-level language that has the advantages of readability, maintainability, and portability.

# The Computer's Brain

- 
- 
- Central Processing Unit (CPU)
  - A CPU has millions of transistors that make use of electronic switches
  - The electronic switches have only two states: off and on. (Symbolically, 0 and 1 are used to represent the two states.)
  - Therefore, a computer can only understand instructions consisting of series of 0s and 1s. In other words, machine-readable instructions have to be in binary format.



# Translators

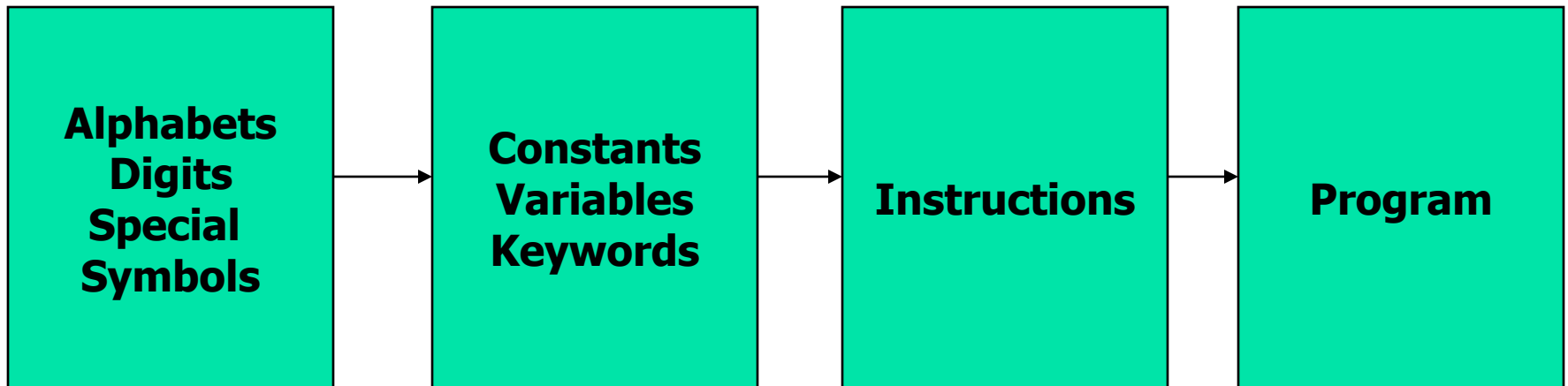
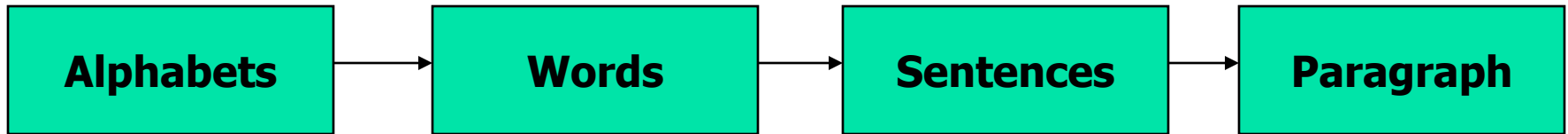
---

- A computer program written in a high-level language, such as C, Java, or Perl, is just a text file, consisting of English-like characters and words.
- Need of special programs, called compilers or interpreters, to translate such a program into a machine-readable code
- The text format of all instructions written in a HLL has to be converted into the binary format.
- The code obtained after the translation is called binary code.
- Prior to the translation, a program in text format is called source code.



# Getting started with C

---





# The Essentials of C Programs

---

- A constant in C is a value that never changes.
- This quantity can be stored at a locations in the memory of the computer.
- A variable, on the other hand, can present different values.
- A variable can be considered as a name given to the location in memory where this constant is stored.
- A combination of constants, variables, and operators is called an expression in the C language. An expression is used to denote different computations.
- A statement consists of a complete expression suffixed with a semicolon.



# Keywords

---

- Words whose meaning has already been explained to the C Compiler.
- Cannot be used as variable names.
- Reserved words
- 32 key words available in C.



# Keywords

---

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while





# Operators in C

---

- Arithmetic Operators.
- Relational Operators.
- Logical Operators.
- Bitwise Operators.
- Unary Operators.



# Arithmetic Operators

---

/	integer division
*	multiplication
%	modulus remainder
+	addition
-	subtraction
/=	division assignment
*=	multiplication assignment
%=	remainder assignment
+=	addition assignment
-=	subtraction assignment
++	increment
--	decrement



# Relational Operators

---

<

less than

<=

less than or equal to

>

greater than

>=

greater than or equal to

==

equal to

!=

not equal to



# Logical Operators

---

! not

&& and

|| or

?: conditional (two separate symbols )



# Bitwise Operators

---

~	not
&	and
^	xor
	or
<<	shift left
>>	shift right
~=	not assignment
&=	and assignment
^=	xor assignment
=	or assignment
<<=	shift left assignment
>>=	shift right assignment



# Unary Operators

---

~

bitwise complement

!

logical not

-

unary minus

\*

pointer indirection

&

address of

+

unary plus



# Precedence And Associativity Of Operators

---

1. `() [] -> .`
- \*2. `! ~ ++ -- + - & (typecast) sizeof`
3. `* / %`
4. `+ -`
5. `<< >>`
6. `< <= > >=`
7. `== !=`
8. `&`
9. `^`
10. `|`
11. `&&`
12. `||`
- \*13. `?:`
- \*14. `= += -= *= /= %= &= ^= |= <<= >>=`
15. `'`

Associativity of the operators is left to right except in case of 2, 13, and 14, where it associates right to left.



# CHARACTER SETS

---

1. C character set.
2. Representable character set.
3. Identifier character set.





# Character Sets

---

- **C character set.**

It is made up of letters, digits, and punctuation marks which are needed to construct a C program.

- **Representable character set.**

It is a super set of the C character set. It is made up of letters, digits and symbols which can be represented by the given systems characters.

- **Identifier character set.**

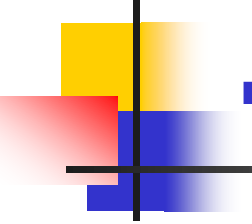
It is made up of letters, digits and underscores( ' \_ ' ) and can be used to construct an identifier.



# Writing Your First C Program

---

- `/* This is my first C Program */`
- `#include<stdio.h>`
- `void main()`
- `{`
- `printf("Hello World !!! ");`
- `}`

- 
- Comments in your C programs are needed to help you document your programs.
  - a comment starts with the opening comment mark, `/*`, and ends with the closing comment mark, `*/`
  - When you read someone's code, the comments in the code help you to understand what the code does, or at least what the code intends to do.
  - Some header files should be included at the beginning of your C program.
  - Header files, such as `stdio.h` and `conio.h`, contain the declarations for functions used in your C program; for example, the `printf()`, `clrscr()`, etc functions.
  - Every C program should have one but only one `main()` function. The program execution starts and ends with the `main()` function.



# C instruction

---

- Type Declaration Instruction
  - to declare the type of variables used in a C Program.
- Input/Output Instruction
  - To perform the function of supplying input data to a program and obtaining the output results from it.
- Arithmetic Instruction
  - To perform arithmetic operations between constants and variables.
- Control Instruction
  - To control the sequence of execution of various statements in a C program.

# Data Type

Sr. No	Data type	Bytes	Format Specifier
1	signed char	1	%c
2	unsigned char	1	%c
3	short signed int	2	%d
4	short unsigned int	2	%u
5	long signed int	4	%ld
6	long unsigned int	4	%lu
7	float	4	%f
8	double	8	%lf
9	long double	10	%Lf



# Control Instruction

---

- Sequence control Instruction
  - Ensures that the instructions are executed in the same order in which they appear in the program.
- Selection or Decision control Instruction
  - Allow the computer to take a decision as to which instruction is to be executed next.
- Repitition or Loop control Instruction
  - Helps computer to execute a group of statements repeatedly.