Introduction to Computing and Programming

Introduction to Programming, Identifiers and Constants

Content

- Quick Recap
- What do computer understand
- Introduction to Programming
- Types of Programming Language
- History of C

Recap

- Computer: **Electronic device** that stores, retrieves, and processes data and can be programmed with instructions.
- **Functions of Computer:**
  - Input,
  - Processing
  - Output
  - Storage
- **Components of Computer:**
  - Hardware
  - Software
  - Users
- **Types of Computers**: Microcomputer, Minicomputer, Personal Computer, Supercomputer, Laptop, Tablets

What do Computer understand?

Computers are only **Smart** because we **program** them to be.

What is Programming?

- Attempting to get a computer to complete a **specific task** without making mistakes.

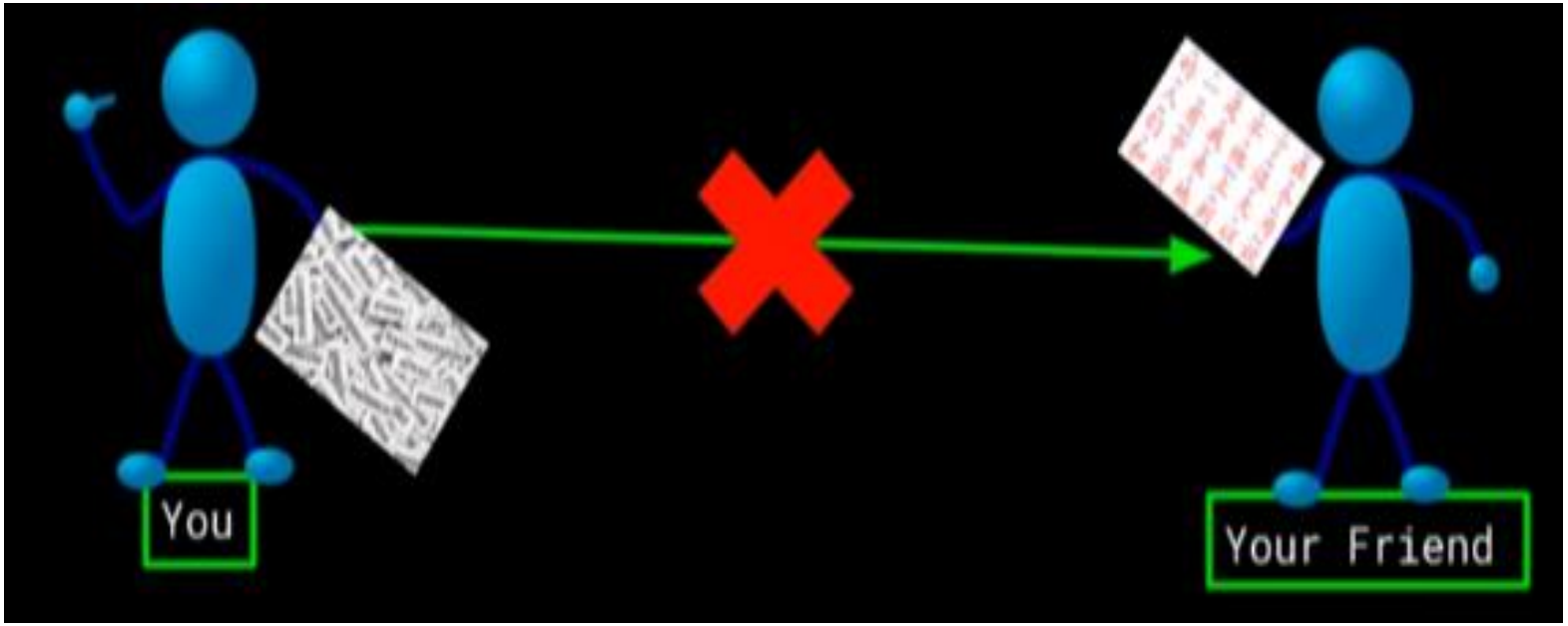
- **Computers are dumb**

What do computers understand Cont.?

- Computer does not understand the language as us like English or Hindi
- Only understands **machine code**
- **A series of 1's and 0's** is interpreted by the computer using **interpreter or Compiler**

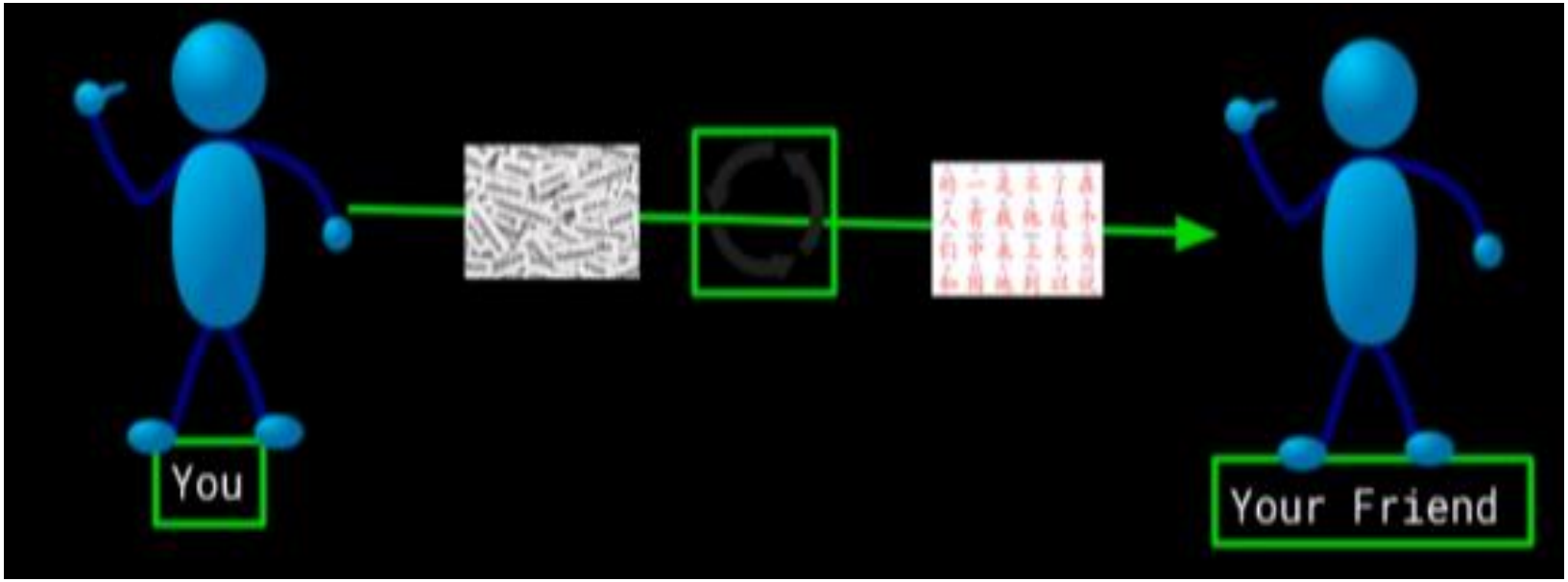
What do computers understand Cont.?

- Consider a friend of yours who only speaks **Mandarin** language and you only speaks **English**
- **Is direct communication possible?**
- Answer is **no.**



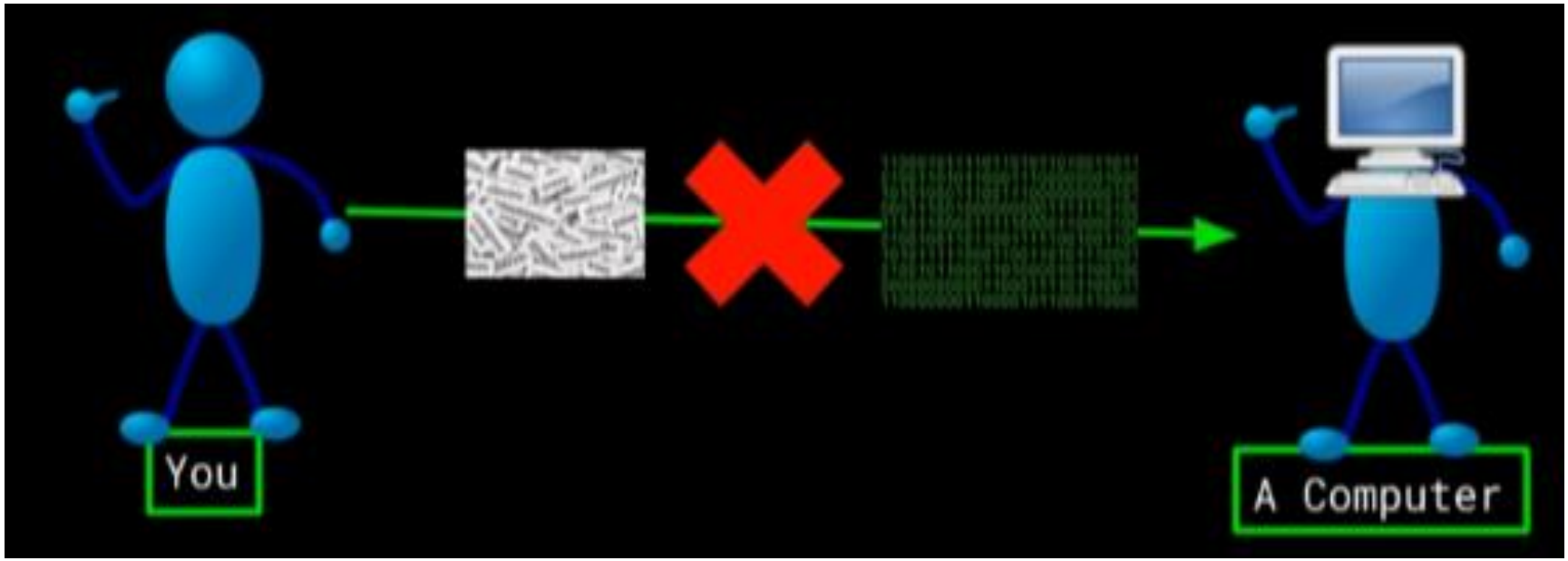
What do computers understand Cont.?

- Communication is only possible **using converting the instructions.**
- You need to **convert your instructions from English to Mandarin** in order for your friend to understand and vice versa.



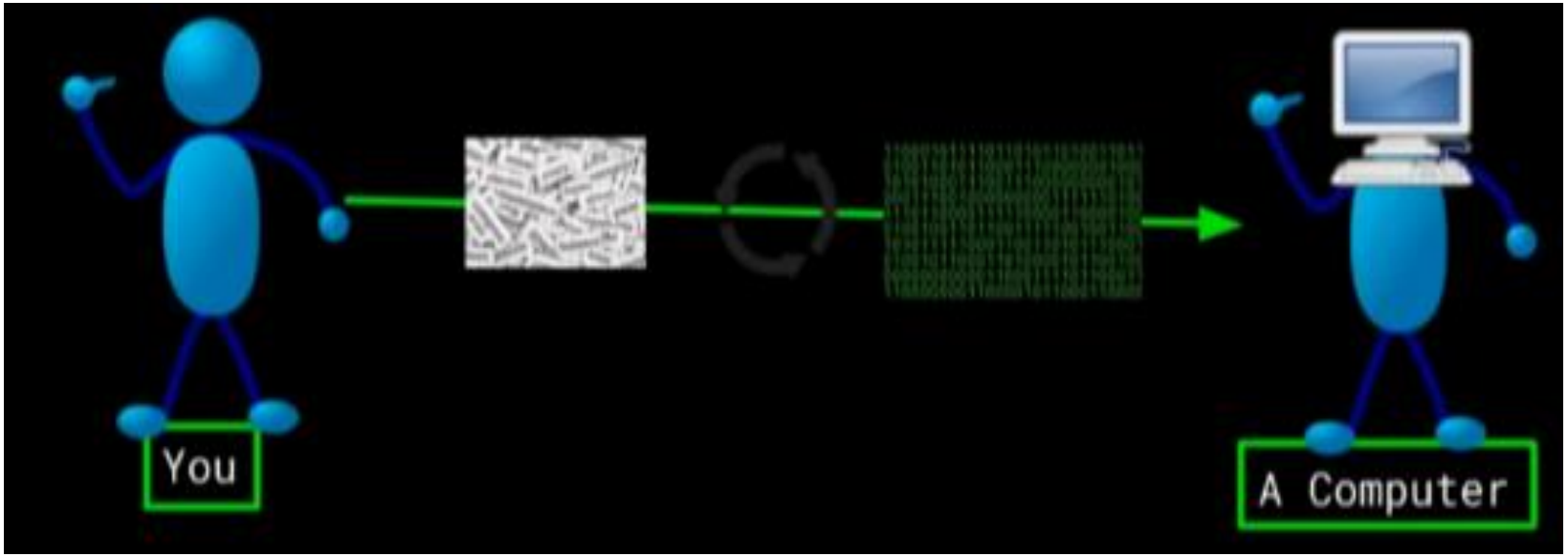
What do computers understand Cont.?

- In order to talk to computer you must first translate your **English instructions to Binary.**
- It is **extremely unpractical** to convert every programming instructions you have into Binary by hand



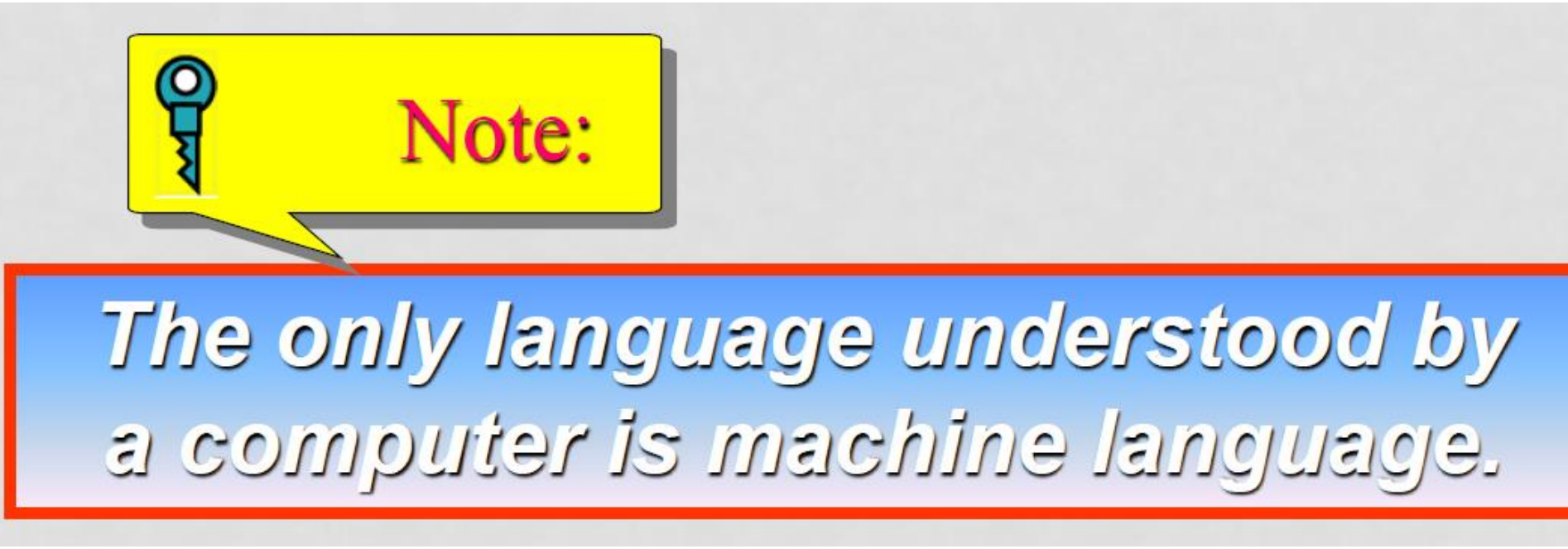
What do computers understand Cont.?

- **Programming languages serves as a middle man** to translate the instructions.



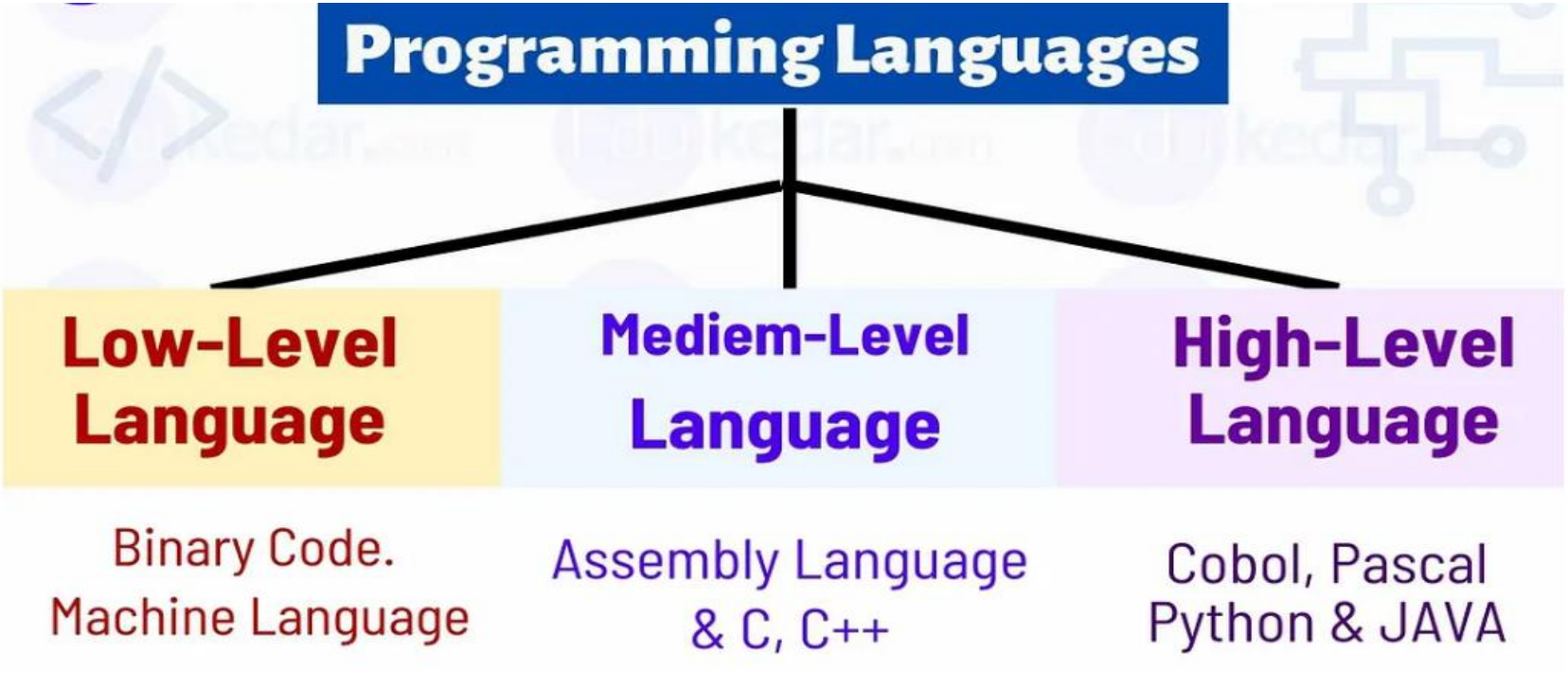
Why programming is important for all of us?

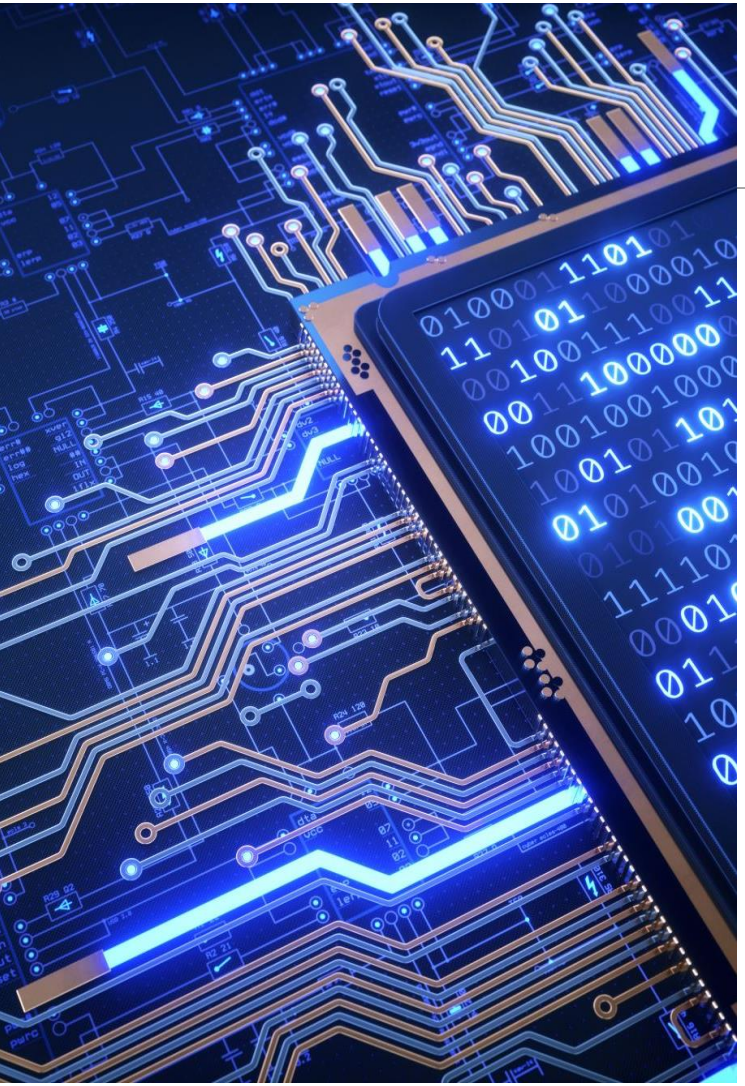
- **Automating tasks: D*ata processing, simulation, and optimization***, which helps programmer to save time and focus on more complex tasks.
- **Problem-solving:** Provide ability to solve complex problems using algorithms that would be difficult to solve using traditional methods.
- **Collaboration:** Allows engineers to work collaboratively on projects which saves time and ensures that the code is error-free.
- **Job opportunities:** Knowledge of programming languages can increase an engineering student's job opportunities.
- **Innovation:** **A**llow engineers to develop new products and technologies.



Computer Language - A Way to Talk to a Computer!

Types of Programming Language





Low Level Language : Machine Language

Very detailed instructions that controls computer's internal circuits.

Advantages:

- Allows user to address computer memory/addresses directly.
- User learns how a computer works.

Disadvantages:

- Many lines of code for a simple program.
- More difficult to locate errors and modify the code.
- Instructions are machine dependent/specific.

Program	Program in machine language
1	00000000 00000100 0000000000000000
2	01011110 00001100 11000010 0000000000000010
3	11101111 00010110 0000000000000101
4	11101111 10011110 0000000000001011
5	11111000 10101101 11011111 0000000000010010
6	01100010 11011111 0000000000010101
7	11101111 00000010 11111011 0000000000010111
8	11110100 10101101 11011111 0000000000011110
9	00000011 10100010 11011111 0000000000100001
10	11101111 00000010 11111011 0000000000100100
11	01111110 11110100 10101101
12	11111000 10101110 11000101 0000000000101011
13	00000110 10100010 11111011 0000000000110001
14	11101111 00000010 11111011 0000000000110100
15	00000100 0000000000111101
16	00000100 0000000000111101

Program

Program in symbolic language

```
1     Entry    main,    ^m<r2>
2     subl2    #12,sp
3     jsb      C$MAIN_ARGS
4     movab    $CHAR_STRING_CON
5
6     pushal   -8(fp)
7     pushal   (r2)
8     calls    #2,read
9     pushal   -12(fp)
10    pushal   3(r2)
11    calls    #2,read
12    mull3    -8(fp),-12(fp),-
13    pushal   6(r2)
14    calls    #2,print
15    clrl     r0
16    ret
```

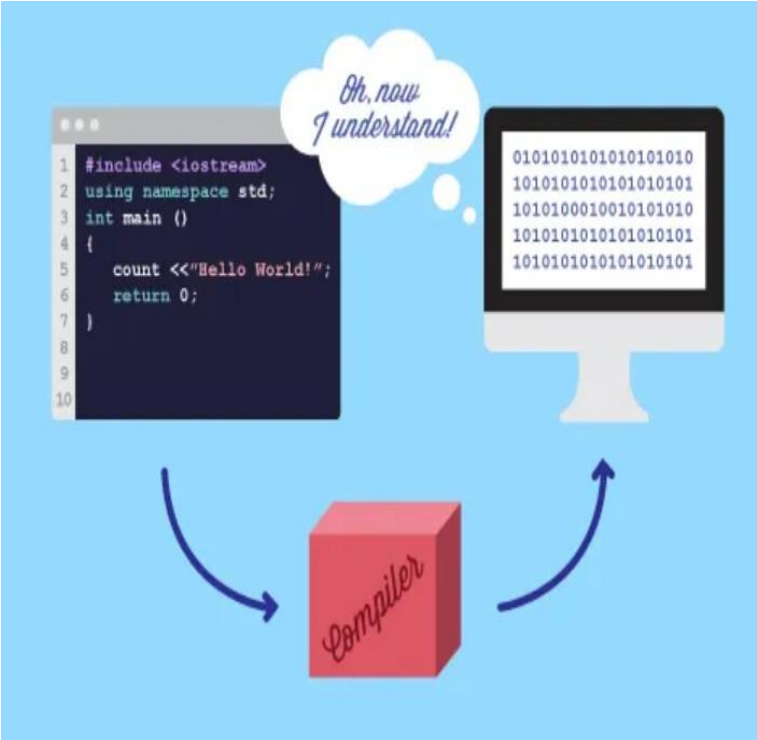
Medium Level Language

- Makes use of symbols such as letters, digits and special characters to create instructions.

Advantages	Disadvantages
• Works as a bridge between high- and low-level languages. • Static typing. • Extremely commonplace. • Easy to reason about program flow.	• Lack of object orientation. • Inefficient memory management. • No garbage collection. • Run-time checking. • Absence of namespace. • Absence of exception handling. • Lacks constructor and destructor.

High Level Language

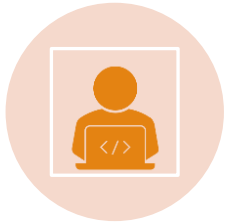
- Written in human understandable form and independent of a particular type of computer.



Advantages	Disadvantages
• Easy to use and understand. • Machine independent. • Debugging is easy. • Easy to maintain and modify program.	• Poor control on hardware. • Slow execution.

Introduction to C

C is a general purpose and structured programming language.



C can be used for system programming and for application programming.

It can be used to write very concise source program due to the availability of extensive libraries.

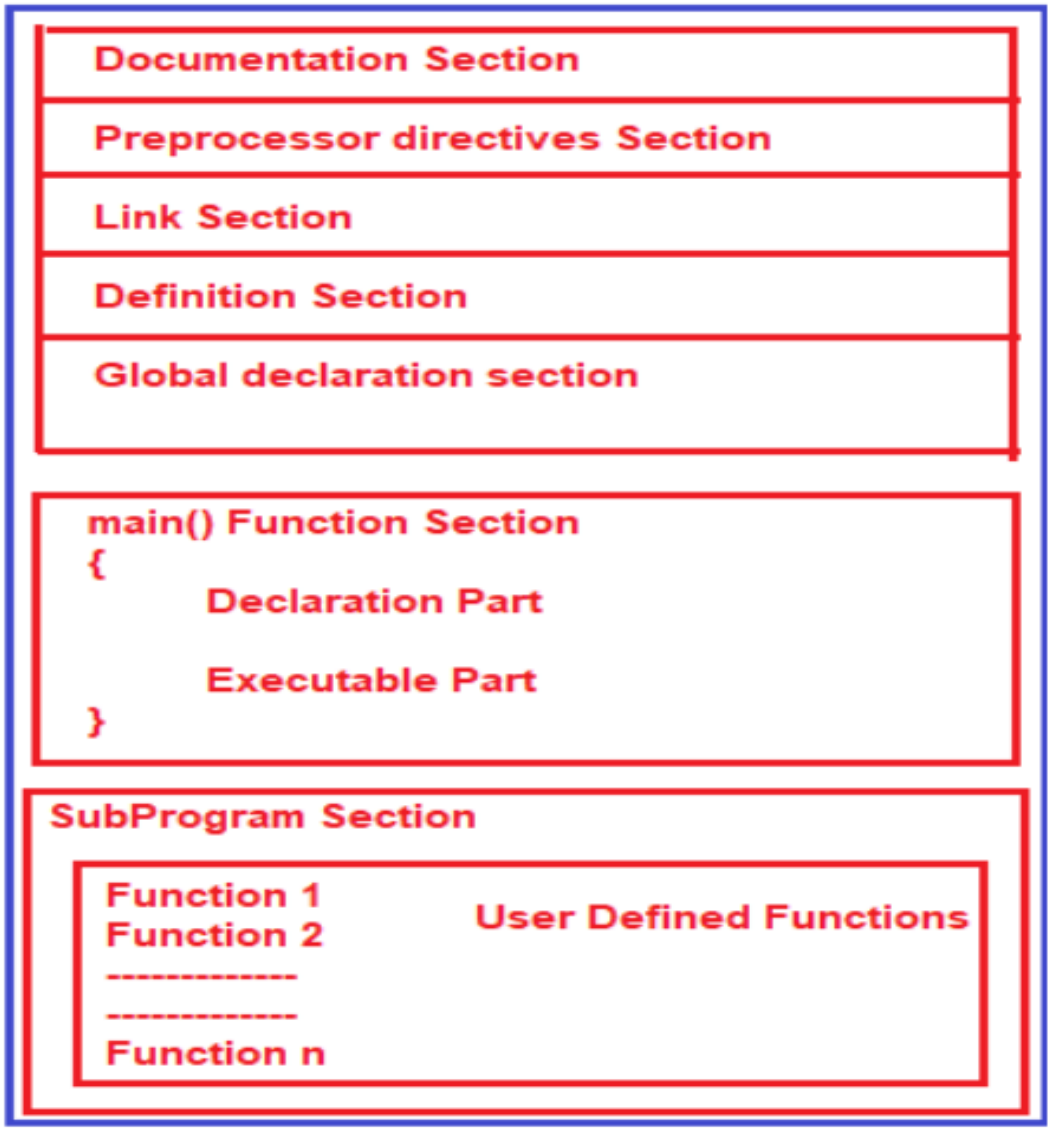
It is highly portable.

History of C Language

- Developed in **1970's by Dennis Ritchie at AT & T Bell Laboratories.**
- It became popular in mid 1980's with the availability of compilers for various platforms.
- Some standardization has been made for C implementation
  - ANSI
  - GNU

Language	Year	Developed By
Algol	1960	International Group
BCPL	1967	Martin Richard
B	1970	Ken Thompson
Traditional C	1972	Dennis Ritchie
K & R C	1978	Kernighan & Dennis Ritchie
ANSI C	1989	ANSI Committee
ANSI/ISO C	1990	ISO Committee
C99	1999	Standardization Committee

Structure of C Program



Upcoming topics

- Number System
- Variables
- Keywords