

# Structured Programming Language

---

Gourab Saha  
Lecturer  
CSE, UIU

Ajwad Akil  
Lecturer  
CSE, UIU

# Problem

1. What is a problem?
2. Fields of problems
3. Most problems can be defined mathematically
4. Use of computer in solving mathematical problems

# Program

1. What is a program?
2. Use of program
3. Programming language

# Programming Language

1. Computer cannot understand human language
2. It can only understand bits (1/0)
3. Bridge between humans and a representation that computer can understand

# Types of Programming Language

## 1. High-Level Language:

- a. C
- b. C++
- c. Java
- d. Python
- e. Javascript
- f. Ruby
- g. Go etc.

## 2. Low-Level Language

- a. Assembly
- b. Machine Code

# Examples ( High-Level Language )

- C

```
#include <stdio.h>
```

```
int main() {  
    printf("Hello, World!");  
    return 0;  
}
```

- Python

```
print('Hello, World!')
```

# Examples ( Low-Level Language )

- Assembly Language

```
name "hi"
org 100h
jmp start
msg:  db    "Hello, World!", 0Dh,0Ah, 24h
start: mov    dx, msg
      mov    ah, 09h
      int    21h
      mov    ah, 0
      int    16h
ret
```

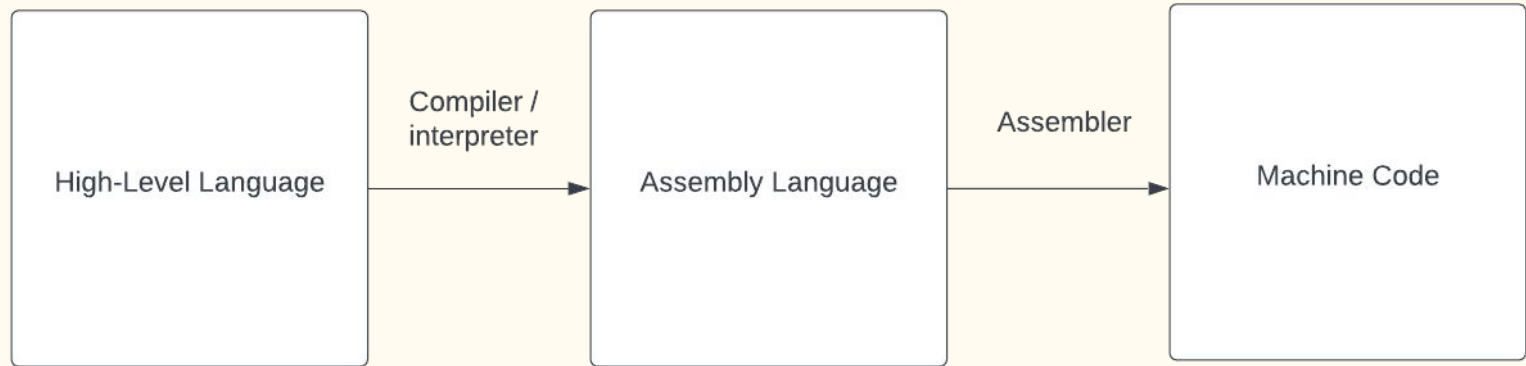
# Examples ( Low-Level Language )

- Machine Code

```
b8  21 0a 00 00
a3  0c 10 00 06
b8  6f 72 6c 64
a3  08 10 00 06
b8  6f 2c 20 57
a3  04 10 00 06
b8  48 65 6c 6c
a3  00 10 00 06
b9  00 10 00 06
ba  10 00 00 00
bb  01 00 00 00
b8  04 00 00 00
cd  80
b8  01 00 00 00
cd  80
```



# Process of Code Compilation



# Why high-level language?

1. Human Readable
2. Easier to
  - a. Understand
  - b. Write
  - c. Debug etc.

# A Bit of History of C language

1. Developed in **1972** by **Dennis Ritchie** at Bell laboratories at **AT&T**
2. Developed to overcome problems of previous languages such as B, BCPL etc
3. Developed to be used in the **UNIX** operating system

## Sources:

1. [History of C Language - javatpoint](#)
2. [C \(programming language\) - Wikipedia](#)

# Relevance of C

1. Mother of languages - directly/indirectly influenced other languages
2. Operating Systems (Window, Linux, Mac OS etc)
3. Databases ( MySQL, PostgreSQL etc)
4. Language Compilers and Core Parts of Language itself (Python)
5. Device Drivers, Game Drivers, Embedded Programs and many more ...

# Why Learn C?

1. Why not simpler languages like Python?
2. C gives you the most important tool to learn to program : To think (somewhat) like a computer
3. Easier to pick up languages when comfortable with C
4. Performance oriented
5. High level of control to the programmer
6. Still relevant to this day

# Structured Programming

## 1. Definition:

- a. Structured programming (sometimes known as modular programming) is a programming paradigm that facilitates the creation of programs with readable code and reusable components ([ref](#))
- b. Structured programming is a programming paradigm aimed at improving the clarity, quality, and development time of a computer program by making extensive use of the structured control flow constructs of selection (if/then/else) and repetition (while and for), block structures, and subroutines. ([ref](#))

## 2. Properties / Features:

- a. Clear
- b. Modular
- c. Reduces coding / development time etc.

# Structure of a C program

Source - [Scaler Topics](#)

```
/**                               //Documentation
 * file: age.c
 * author: you
 * description: program to find our age.
 */

#include <stdio.h>                //Link

#define BORN 2000                 //Definition

int age(int current);            //Global Declaration

int main(void)                   //Main() Function
{
    int current = 2021;
    printf("Age: %d", age(current));
    return 0;
}

int age(int current) {           //Subprograms
    return current - BORN;
}
```

# Structure of a C program

1. Documentation
2. Link / Preprocessor
3. Definition
4. Global Declaration
5. Main Function
6. User Defined Function / Subprogram

## Sources:

1. [Structure of C Program - Scaler Topics](#)
2. [Structure of a C program - javatpoint](#)



# IDEs to use

1. What is an IDE?
2. What is a code editor?
3. Difference between IDE and code editor
4. Examples of IDE for C programming
  - a. CodeBlocks
  - b. CLion
  - c. Visual Studio
  - d. VSCode etc.

# Problem Solving

1. What is problem solving?
2. Why is it important to solve problems
3. Places to solve problems from
  - a. Geekkforgeeks
  - b. HackerRank
  - c. TopCoder
  - d. CodeChef
  - e. CodeForces