



C PROGRAMING

by Ketan Kore @Sunbeam Infotech



About Trainer

- Mr. Ketan Kore
 - Trainer at Sunbeam Infotech
 - M.Sc. Computer science
- Training
 - PreCAT(Training + lab)
 - DAC course(LAB support)
- Contact
 - Email: ketan.kore@sunbeaminfo.com
 - Mobile: 8805822402



Computer and Program

- What is Computer?
 - It is a machine/hardware/digital device which does various tasks for the user efficiently and effectively.
- • What is Program?
 - Set of instructions given to the machine to do specific task.



Classification of Languages

- **The low-level language** is a programming language that provides no abstraction from the hardware, and it is represented in 0 or 1 forms, which are the machine instructions.
- **The high-level language** is a programming language that allows a programmer to write the programs which are independent of a particular type of computer. The high-level languages are considered as high-level because they are closer to human languages than machine-level languages.



Low level Languages

- Machine-level language

- **The machine-level language** is a language that consists of a set of instructions that are in the binary form 0 or 1. As we know that computers can understand only machine instructions, which are in binary digits, i.e., 0 and 1, so the instructions given to the computer can be only in binary codes.
- **Advantages:**
 - Performance is good as we are directly writing the program on machine
- **Disadvantages:**
 - Machine dependent
 - Difficult to program
 - Error prone

- Assembly Language

- The assembly language contains some human-readable commands Alphanumeric(Alphabet+numbers) codes, The language was introduced in 1952.
- As we know that computers can only understand the machine-level instructions, so we require a translator that converts the assembly code into machine code. The translator used for translating the code is known as an assembler.
- **Advantages**
 - Easier to understand and use and to locate errors
 - Easier to modify
- **Disadvantages**
 - Machine dependent
 - Knowledge of hardware required
 - Machine level coding required



High Level Languages

- The high-level language is a programming language that allows a programmer to write the programs which are independent of a particular type of computer. The high-level languages are considered as high-level because they are closer to human languages than machine-level languages.
- A compiler is required to translate a high-level language into a low-level language.
- **Advantages:**
 - They are machine independent
 - They do not require programmer to know anything about hardware
 - They do not deal with machine level coding
- **Disadvantages:**
 - It takes additional translation times to translate the source to machine code.
 - High level programs are comparatively slower than low level programs.



C programming Language

- C programming Language is an High level Language
- C is a general-purpose programming language that is extremely popular, simple, and flexible to use.
- Machine Independent or Portable
- C language include low-level access to memory has simple set of keywords.
- C language is the most widely used language in operating systems and embedded system development today.



History

- C language was developed by Dennis Ritchie in 1972 at AT & T Bell Labs on PDP-11 machine.
- It was developed while porting UNIX from PDP-7 to PDP-11.
- Many features of C are inspired from B (Ken Thompson) and BCPL (Martin Richards).
- Initial release of C is referred as K & R C.



Standardization

- C was standardized by ANSI in 1989. This is referred as C89.
- Standardization ensures C code to remain portable.
- C standard is revised multiple times to add new features in the language.
 - C89 – First ANSI standard
 - C90 – ANSI standard adopted by ISO
 - C99 – Added few C++ features like `bool`, `inline`, etc.
 - C11 – Added multi-threading feature.
 - C17 – Few technical corrections.



Introduction

- High-level
- Compiled
- Procedural
- Block-Structured (control structures).
- Typed
- Library Functions



Features

- Data types
- Operators
- Control structures
- Functions
- Storage classes
- Pointers
- Arrays
- Strings
- Dynamic memory allocation
- Structures
- Unions
- Enums
- File IO
- Preprocessor directives



Strengths

- Low level memory access (pointers, data structures)
- Effective memory access (bitwise operators, bit-fields, unions)
- Can access OS features (functions/commands)
- Extensive library functions (math, strings, file IO, ...)
- Compilers for different platforms & architectures
- Highly Readable (macros, enum, functions, ...)



Applications

- System programming
 - OS development
 - Device drivers
 - System utilities
- Language development
 - Compiler development
- Achievements (tiobe.com)
 - In top-2 languages in last 40 years.
 - Language of year: 2019, 2017, 2008.



Toolchain & IDE

- Toolchain is set of tools to convert high level language program to machine level code.
 - Preprocessor
 - Compiler
 - Assembler
 - Linker
 - Debugger
 - Utilities
- Popular compiler (toolchains)
 - GCC
 - Visual Studio
- IDE – Integrated development environment
 - Visual Studio
 - Eclipse
 - VS Code (+ gcc)
 - Turbo C
 - Anjuta, KDevelop, Codeblocks, Dev C++, etc.



Software installation

- Installations
 - GCC (MinGW)
 - VS Code



Hello World

- Source Code

```
// Hello World program
#include <stdio.h>
int main() {
    printf("Hello World\n");
    return 0;
}
```

- Commands

- cmd> gcc hello.c
- cmd> ./a.exe





Thank you!

Ketan Kore <ketan.kore@sunbeaminfo.com>

