“चला तर, Coding शिकू आपल्या भाषेत!”

Marathi Coding Shala

@JayeshKande-g6f1e · 125 subscribers · 7 videos

नमस्कार मित्रांनो! 👋 🙏 ...more

youtube.com/channel/UCUc5mUOo3xqzlgXFCaIFkbw and 2 more links

Customize channel | Manage videos | View channel stats

📘 C Language Syllabus

1. Introduction to C

- History of C language
- Features of C
- Structure of a C program
- Compilation and Execution process
- C Tokens: Keywords, Identifiers, Constants, Strings, Operators

2. Variables, Data Types & Operators

- Data Types (int, float, char, double, void)
- Type Modifiers (short, long, signed, unsigned)
- Variables and Constants (const, #define)
- Operators:
    - Arithmetic (+ - * / %)
    - Relational (== != < > <= >=)
    - Logical (&& || !)
    - Bitwise (& | ^ ~ << >>)
    - Assignment (= += -= *= /= %=)
    - Increment/Decrement (++ --)
    - Conditional / Ternary (?:)
    - Comma and sizeof

3. Input / Output

- printf() and scanf()
- getchar(), putchar()
- gets(), puts()

4. Control Statements

- Decision Making:
  - if, if-else, nested if
  - switch-case
- Looping:
  - for loop
  - while loop
  - do-while loop
- Jump Statements:
  - break, continue, goto

5. Functions
- Function Definition and Declaration
- Function Prototype
- Function Arguments (call by value & call by reference)
- Recursion
- Storage Classes: auto, static, extern, register

6. Arrays and string
- Single Dimensional Array
- Multi-Dimensional Array (2D array – matrix)
- String Handling in C:
    - Declaration and Initialization of Strings
    - Standard String Functions (strlen, strcpy, strcat, strcmp, etc.)

7. Pointers
- Introduction to Pointers
- Pointer Arithmetic
- Pointer to Pointer
- Array and Pointers
- Pointer with Functions (call by reference)
- Pointer to Strings
- Pointer to Structure
- Dynamic Memory Allocation (malloc, calloc, realloc, free)

8. Structures & Unions
- Defining and Using Structures
- Array of Structures
- Nested Structures
- Pointer to Structures
- Union (difference from structures)
- typedef keyword

9. File Handling

- File Operations:
  - fopen(), fclose()
  - fscanf(), fprintf(), fgets(), fputs()
  - fread(), fwrite()
- File Modes (r, w, a, r+, w+, a+)
- Random File Access: fseek(), ftell(), rewind()

AI with C – C language is used for implementing core AI algorithms, basic ML models, and high-performance applications like embedded systems and game AI.

10. Preprocessor & Advanced Concepts

- Macros (#define)
- File Inclusion (#include)
- Conditional Compilation (#ifdef, #ifndef, #endif)
- Command line arguments
- Enum (Enumerations)

📜 History of C Language

1.BCPL (1966)

1. Developed by **Martin Richards**.
2. Mainly used for writing **system software and compilers**.

2.B Language (1969)

1. Developed by **Ken Thompson** at **Bell Labs**, based on BCPL.
2. Used in the early development of the **UNIX operating system**.

3.C Language (1972)

1. Developed by **Dennis Ritchie** at **Bell Labs** as an improved version of B.
2. The **UNIX operating system** was rewritten in C, which made it **portable** across different machines.
3. This was the key reason behind C's success.

4.Standardization

1. In **1989**, ANSI (American National Standards Institute) standardized C as **ANSI C** (also called C89/C90).
2. Later versions were released: **C99, C11, C18** with new features.

✨ Why C Became Popular?

•Simple yet very powerful.

•Combines **low-level features** (close to hardware) with **high-level features** (easy to understand).

•Widely used in **system programming, compilers, operating systems, and embedded systems**.

❑ What is a Program?

•**Definition**: A program is a set of instructions that tells a computer what tasks to perform.

•**How It Works**: Similar to a recipe guiding a chef, a program guides a computer to complete actions step-by-step.

•**Examples**:
•**Smartphone App**: The app for a smart home system is a program that lets you control lights or the thermostat from your phone.

•**Remote Control for TV**: The remote is like a program that sends instructions to the TV to change channels, adjust volume, or turn on/off.

•**Calculator App**: A simple program that performs math operations when you input numbers.

❑ Why We Need to Learn Programming Languages?

Definition: Programming is the way humans communicate with computers, using special languages to give them instructions. Just like we use words to communicate with each other, we use programming languages (like Python, Java, or C++) to communicate with machines.

Example:

•**Human-to-Human Communication**: When we talk or text, we use language to share information and ask each other to do things.

•**Human-to-Machine Communication**: With programming, we “talk” to computers, telling them what to do, like instructing a robot to move or telling an app to send a message.

❑ Why We Need to Communicate with Machine?

•Machines are like dumb devices; they require clear instructions to function.

•By giving them commands, we can make them work efficiently and accurately.

•Communication with machines helps automate repetitive tasks, saving us time.

•It reduces the chance of human error, leading to better results.

•Machines can analyze data and solve complex problems faster than humans.

❑ Can You Give a Program Directly to a Machine?

```c
#include <stdio.h>

int main() {
    // Print Hello, World! to the console
    printf("Hello, World!\n");
    return 0;
}
```

NO

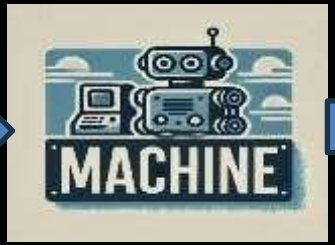
❑ Can You Give a Program Directly to a Machine?

```c
#include <stdio.h>

int main() {
    // Print Hello, World! to the console
    printf("Hello, World!\n");
    return 0;
}
```

Translator /complier

Machine readable (0101010110)



output

❑ Learning a Programming Language:

- **Definition**: Understanding how to write code in a specific language.

- **Key Aspects:**

1. **Syntax**: Code structure and rules.
2. **Semantics**: Meaning of the code.
3. **Data Structures**: Ways to store data.
4. **Control Structures**: Managing code flow (loops, conditionals).
5. **Problem-Solving**: Implementing solutions with code.

- **Benefits:**

- **Automation**: Automate tasks.
- **Software Development**: Build applications.
- **Career Opportunities**: Jobs in tech fields.

ATM Withdrawal

1. Insert Card

2. Enter PIN

3. Select Amount

4. Dispense Cash

Insufficient Balance

❑ Why is Learning Programming Important?

Example: ATM User vs. ATM Programmer

1. ATM User:

•**What They Do**: Withdraw cash, check balance.

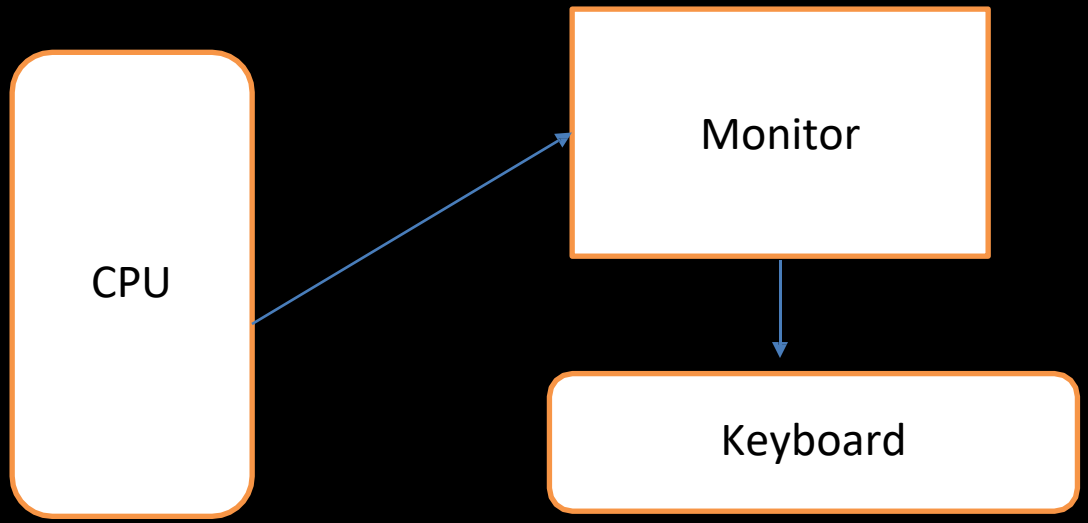
•**Skills**: Basic operations; no need for programming knowledge.

2. ATM Programmer:

•**What They Do**: Develop ATM software, design functions and security features.

•**Skills**: Knowledge of programming languages, database management, and technical skills.

❑ Every Programming Language to must have Feature



1) To read Fetch Input Coming From KeyBoard

2) Display Output On Screen

❑ Operating System:

Definition: Software that manages computer hardware and software resources, providing common services for computer programs.

Platform Dependent OS

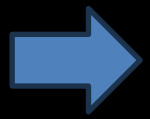
An operating system that is designed to work only on a specific type of hardware or platform.

Examples:
Windows (x86/x64), macOS (Apple hardware), and Android (ARM devices)
Linux.

❑ Windows

```c
#include <stdio.h>

int main() {
    // Print Hello, World! to the console
    printf("Hello, World!\n");
    return 0;
}
```



Translator
/complier

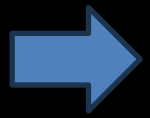
.exe
(runnable File)

The C program will compile to .exe file that can be run on Windows.

❑ Linux

```c
#include <stdio.h>

int main() {
    // Print Hello, World! to the console
    printf("Hello, World!\n");
    return 0;
}
```



Translator
/complier

.out
(runnable File)

The C program will compile to .out file that can be run on Linux.

- list of common file types along with their extensions and associated software

File Extension	File Type	Software
.exe	Executable file	Windows OS
.txt	Text file	Notepad, any text editor
.jpg	Image file	Windows Photo Viewer
.pdf	Portable Document Format	Adobe Reader
.mp3	Audio file	Windows Media Player
.c	C source file	Visual Studio, Code::Blocks

❑ What is Algorithm:

An **algorithm** is a step-by-step set of instructions designed to perform a specific task or solve a particular problem. In computer science, algorithms are foundational for programming, guiding computers to execute tasks efficiently and accurately.

❖ Example: Algorithm for Adding Two Numbers

•Start
•Take two numbers as input (A and B)
•Add the numbers: Sum = A + B
•Output the result: Sum
•End

❑ What is Syntax:

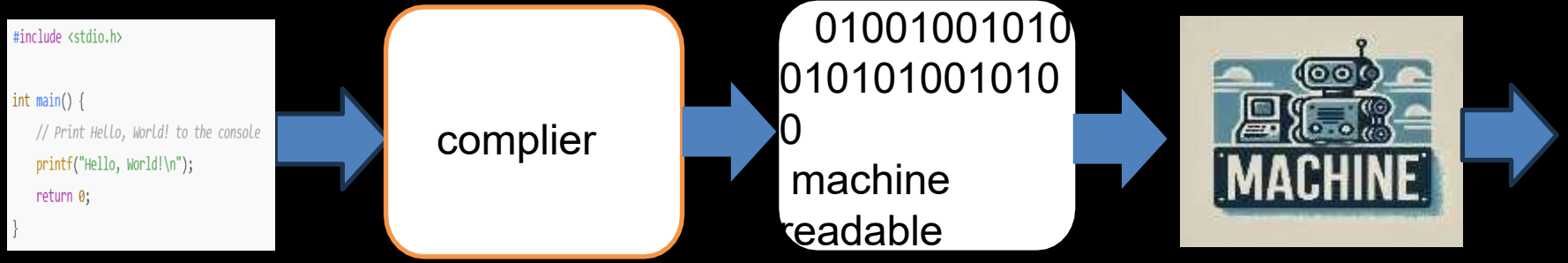
Syntax is the set of rules that defines the correct structure of code in a programming language. It specifies how commands, keywords, and symbols should be arranged.

❖ Example: In C, to print text

• printf("Hello, World!");

❑ What is Complier:

A **compiler** is a program that translates high-level code (like C, Java) into machine code or binary code that a computer can understand and execute.



Kickstart Your Coding: Installation and Setup Guide

- What’s an IDE and Why use one?
- why an IDE Makes Coding Easier
- How to set it up on windows
- How to get started on mac

❑ What's an IDE and Why use one?

Point	Description
1.	**Definition:** An IDE (Integrated Development Environment) is a software suite that provides tools for software development.
2.	**Core Components:** Includes a code editor, compiler, debugger, and build automation tools.
3.	**Functionality:** Centralizes coding, testing, and debugging to enhance workflow and efficiency.
4.	**Benefits:** Increases developer productivity by reducing time spent on switching tools and streamlining tasks.
5.	**Examples:** Popular IDEs include Visual Studio, Eclipse, IntelliJ IDEA, and PyCharm.

❑ why an IDE Makes Coding Easier

Feature	Description
Streamlines Development	Facilitates faster coding by integrating essential tools.
Increases Productivity	Reduces time spent on repetitive tasks and tool switching.
Simplifies Complex Tasks	Breaks down challenging processes into manageable steps.
Offers a Unified Workspace	Combines all development activities in one interface.
Code Autocomplete	Suggests code completions to speed up writing.
Syntax Highlighting	Enhances readability by color-coding code elements.
Version Control	Manages code changes and collaboration efficiently.
Error Checking	Identifies and highlights errors in real-time.

❑ Basic Structure of a C Program

```c
#include <stdio.h>   // Preprocessor directives (header files)

int main()           // Main function – entry point of program
{
    // Variable declaration
    int a, b, sum;

    // Input
    printf("Enter two numbers: ");
    scanf("%d %d", &a, &b);

    // Processing
    sum = a + b;

    // Output
    printf("Sum = %d\n", sum);

    return 0;   // Exit status
}
```

❑ .c vs .h extensions

Feature	.c Files	.h Files
Purpose	Contains function definitions and main logic.	Contains declarations and prototypes.
Compilation	Compiled into object code.	Not compiled; included in .c files.
Content	Has variable definitions and implementations.	Contains function prototypes and constants.
Usage	Implements program functionality.	Shared declarations among multiple files.

❑ Recap:

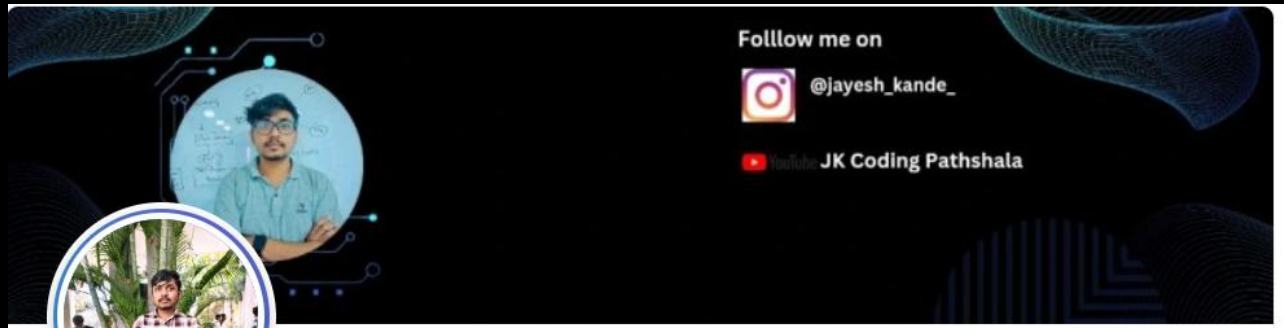
- ❖ What is a program?
- ❖ Why do we need to learn programming languages?
- ❖ Why is it necessary to communicate with machines?
- ❖ Can we give a program directly to a machine?
- ❖ What are the components of learning a programming language?
- ❖ Why is learning programming important?
- ❖ What is an Operating System (OS)?
- ❖ What are platform-dependent operating systems?
- ❖ What's an IDE, and why use one?
- ❖ What file extensions are generated when compiling a C program for different operating systems?

jayesh_kande_ Follow Message •••

33 posts 438 followers 358 following

Jayesh Kande
👩‍💻 Faculty @ CoDing SeeKho
📚 Coding | College Exams | Placement Prep
🎥 YouTuber – JK Coding Pathshala
🚀 Guiding Future... more
🔗 yt.openinapp.co/0y0qd + 1

LinkedIn



Jayesh Kande

Faculty at Coding Seekho (Offline + Online)|IT Engineering | Aspiring Web Developer | Java Enthusiast | Data Structures & Algorithms | Proficient in C, C++, Java, and MERN Stack | AI + Web Dev

Nashik, Maharashtra, India · Contact Info

725 followers · 500+ connections

Kbt engineering college nashik

JK Coding Pathshala

@jayeshkande9215 · 667 subscribers · 149 videos

🎓 Welcome to JK Coding Pathshala! ...more

youtube.com/channel/UC474QOAov9dTP9Y1nZwupqw and 3 more links

Customize channel | Manage videos | View channel stats