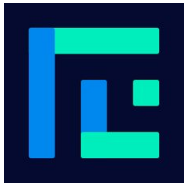


Introduction To Programming

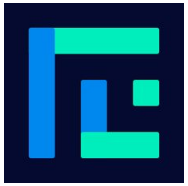
- Viraj Chandra

What is programming?

- You give a set of instructions to the computer
- The computer executes all instructions in a specific order
- There should be enough information for the computer to decide what to do in all situations
- These instructions or steps are called a program or code
- Does computer understand English?
  - **Programming Language**

Assembly → Compiler → computer

Programming Process: Step-by-Step Example



- **Define the Problem - Example**: "We want the computer to calculate the sum of two numbers."

- **Plan the Steps (Algorithm)**
  - Input two numbers.
  - Add the numbers together.
  - Display the result.

- **Choose a Programming Language**
  - Example language: C++

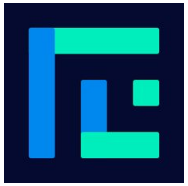
Programming Process: Step-by-Step Example

- **Write the Code (Program)**

```
int number1, number2;
cin >> number1 >> number2;       // input the numbers
int sum = number1 + number2;     // calculate sum
cout << "The sum is: " << sum << endl;     // Display the result
```

- **Execute the Code**
  - Compile the code using a C++ compiler (e.g., g++).
  - Run the program.
  - Output: **"The sum is: 8"**

Some Issues! ❌



What if you don’t give the instructions correctly?

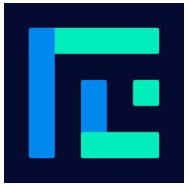
- **Issue**: Syntax errors or logic errors occur.

  Example: Forgetting to close a parenthesis or using the wrong operator.

  ```
  int sum = (number1 + number2)      // Missing opening parenthesis
  ```

- **Result**: The program won’t compile or will produce unexpected results.

Some Issues! ❌



What if some instruction fails?

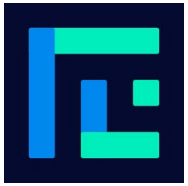
- **Issue**: Runtime errors (e.g., division by zero or file not found).

  Example:

  ```
  int result = 10 / 0;    // Division by zero causes runtime error
  ```

- **Result**: Program crashes or stops unexpectedly.

Some Issues! ❌



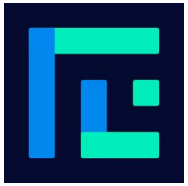
What if even after correct execution you get the wrong result?

- **Issue**: Logical errors or incorrect formulas.

  Example:
  int average = total / 2; // Wrong formula if `total` represents sum of 3 numbers

- **Result**: Output looks correct at first glance but is actually wrong.

Some Issues! ❌



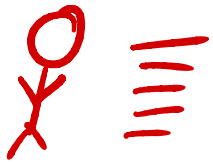
What if it takes too long to complete the instructions?

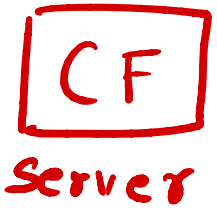
- **Issue**: Inefficient algorithms or poor design.

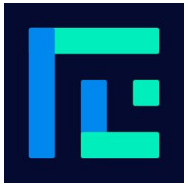
  Example: Searching for a file in folder with **100 subfolders** where each subfolder contains **10000 subfolders.** If we search in the parent folder it will take too much time but if **we know exactly in which folder** to search we do it faster.

- **Result**: Program takes too long to complete.

Some Issues!







What if it takes too many resources to complete the instructions?

- **Issue**: High memory or CPU usage.

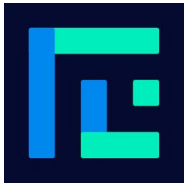
  Example:

  ```
  vector<int> large_array(1e9, 0);     // Allocating 1 billion integers
  ```

- **Result**: System runs out of memory or becomes unresponsive.

10^9

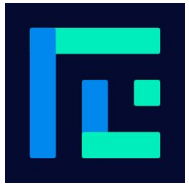
Choose and Learn Programming Languages



- Why so many programming languages?
- Does it matter which one you learn first?
- How **easy it is to switch** between languages?

{10 19 1 17 . . . . }

What is Competitive Programming?



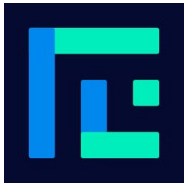
- **Many people competing to**
  - Write the code that gets executed fastest
  - Write the code that uses minimum resources
  - Do this faster than others
- **How is it different from normal programming?**
- **Best programming language CP**
  - C++
- **How is competitive programming useful?**

logic 1 logic 2

500 100000

1000

How is Competitive Programming Useful?



- **Enhances Problem-Solving** – Boosts logical and analytical thinking.
- **Deepens Algorithmic Knowledge** – Improves understanding of algorithms and data structures.
- **Builds Coding Efficiency** – Encourages writing clean, optimized code.
- **Prepares for Interviews** – Sharpens skills for tech interviews and real-world challenges.
- **Improves Debugging** – Trains you to fix issues quickly.
- **Encourages Networking** – Connect with like-minded coders.
- **Boosts Careers** – Opens doors to top tech roles and competitions.
- **Sharpens Mental Agility** – Speed and accuracy under pressure.