

Welcome to Programming Fundamentals with C++

This course is designed to teach you the fundamental concepts of programming using C++. We'll explore problem-solving methodologies and build a solid foundation for your programming journey. Get ready to design, analyze, and decompose complex problems!



Course Overview: Mastering C++ Fundamentals

This course aims to master fundamental programming concepts using C++. We'll cover key topics, grading, and the schedule. There are helpful resources such as textbooks and online platforms like tutorials point and w3schools.

Versatility

Highly versatile language

Performance

Excellent performance

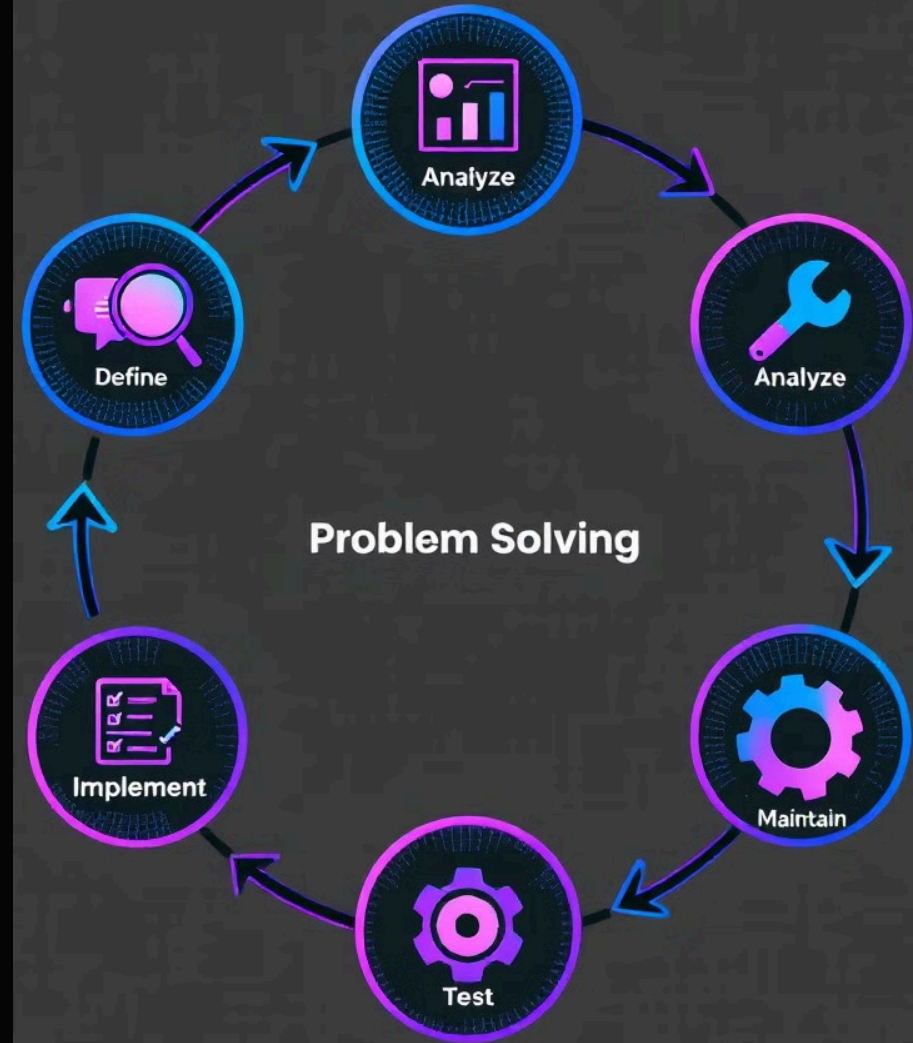
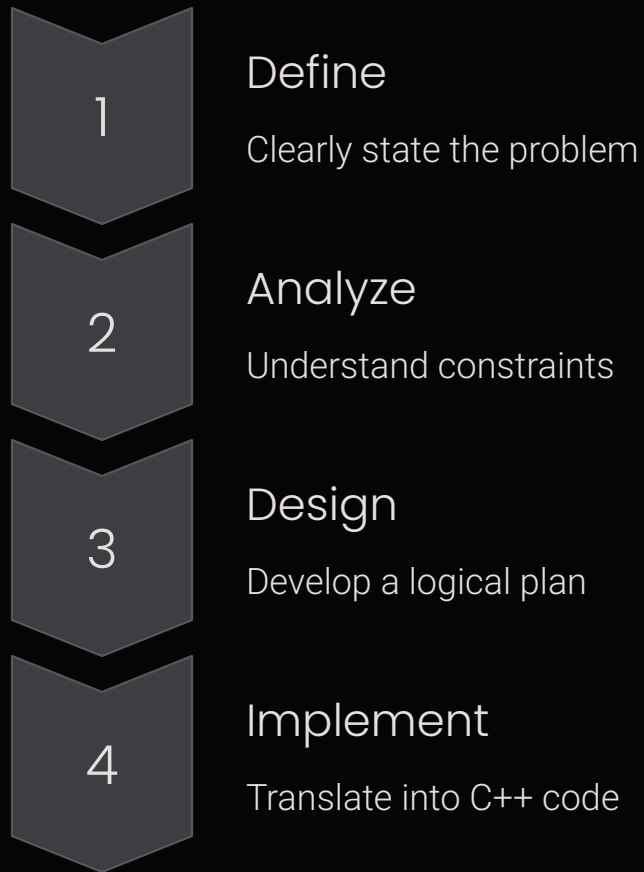
Relevance

Industry relevance

```
mut fon are(enedby) {  
    conarcts wbiel; // stiction;  
    chlide.;  
    beffen is = cat+ cnuch. + client);  
    cattns, instrual {  
        caaut clnobor+{  
            costrunt ();  
            dedfing, = del(lar + is (low, ened));  
            facutr();  
            ++= ct ++ . alerplution convention)  
            vald.(lt + lx l);  
            detplynt ar, fallen));  
        )  
        c+++ l);  
        compoutions is, = (lowl){ + blac nuch(enerade  
  
        citini);  
        /+++ _nait ();  
        contract (lon) si result)) |  
  
        d))  
        callention(( = r contract+ #lye  
        at y);  
  
        claualle, neccentional);  
        blup()  
        iif, sdat(lar, or constan))(  
            clatent(estion'(last low enerate, elude)  
            ndeterit, (lawt Lohari lae)  
            /enatlenpstll, = choalic int zilu + lawne + huclyre
```

The Problem-Solving Methodology

A step-by-step guide to solve any problem.



A close-up, low-angle shot of a desk at night. On the desk is a dark-colored mug on a saucer, a small lit candle, and several potted plants. In the foreground, there are two papers. One paper is titled 'Algorithms' and has some diagrams and text on it. The other paper is titled 'DESIGNING' and has some text on it. The background is dark and out of focus, showing more plants and a window with some light coming through.

Design: Crafting Solutions with Algorithms

An algorithm is a step-by-step procedure for solving a problem. A good algorithm needs clarity, efficiency, and correctness. Here's the algorithm for calculating the area of a triangle given its base and height.

1

Base Input

Get the base of a triangle

2

Height Input

Get the height of a triangle

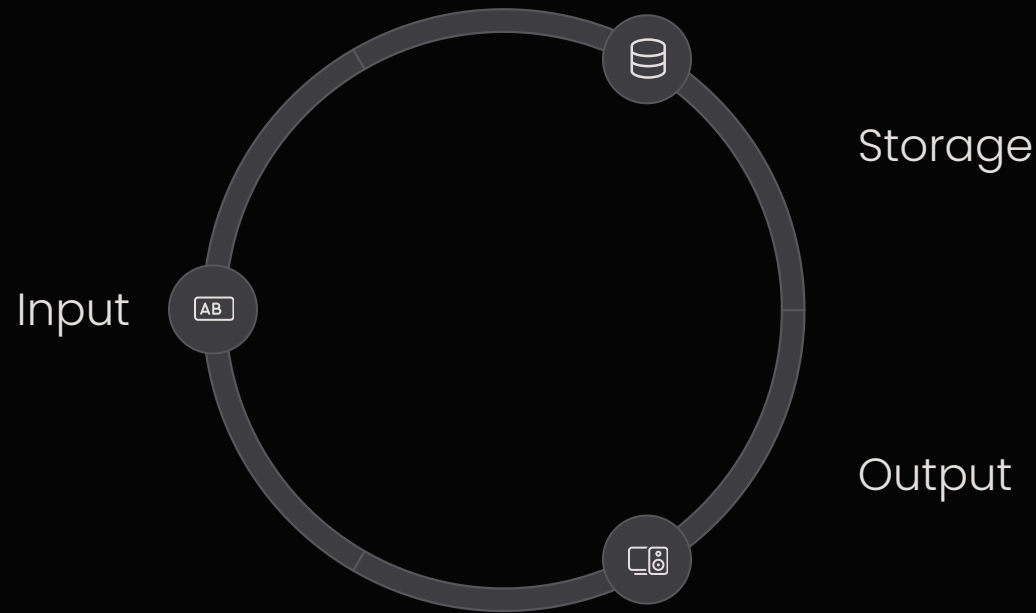
3

Calculate

Multiply base by height then divide by 2

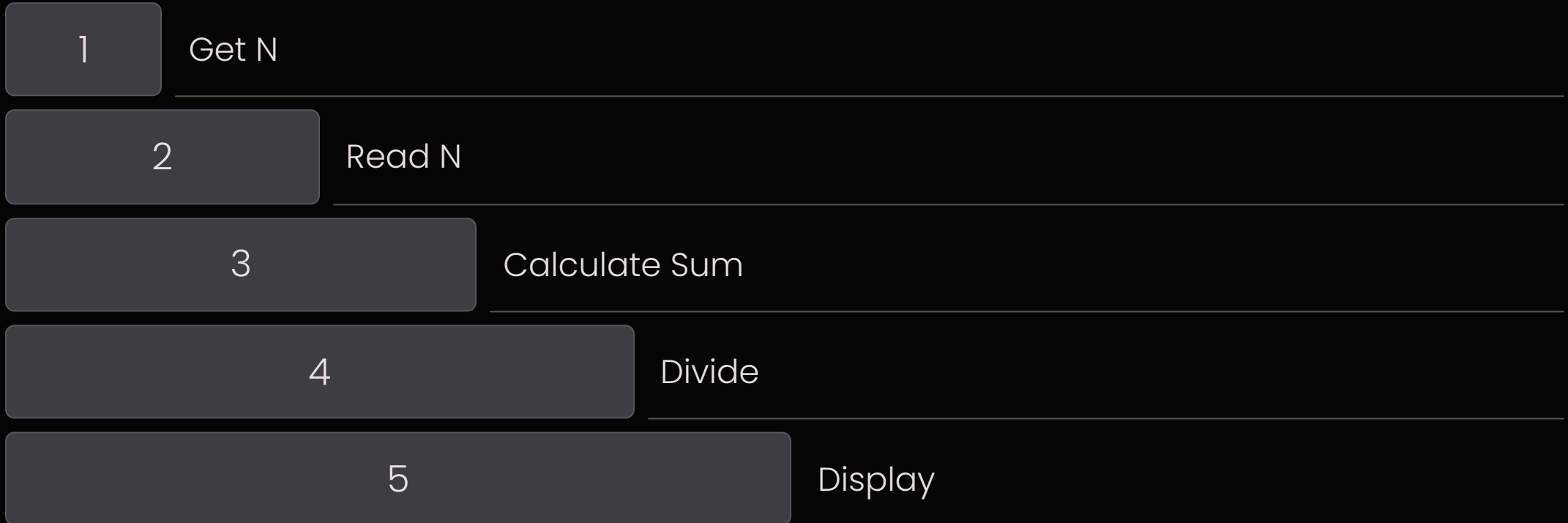
Analyze: Breaking Down Complex Problems

Decompose complex problems into smaller, manageable sub-problems, such as building a calculator app with components like input, storage and output. Simplifies development, enhances maintainability, and facilitates teamwork.



Decompose: Calculating the Average

Write a program to calculate the average of N numbers.



Algorithms: The Heart of Programming Logic

A finite sequence of well-defined instructions to solve a specific problem. They must be unambiguous, feasible, and produce the correct output.

Sorting

- Bubble sort
- Insertion sort

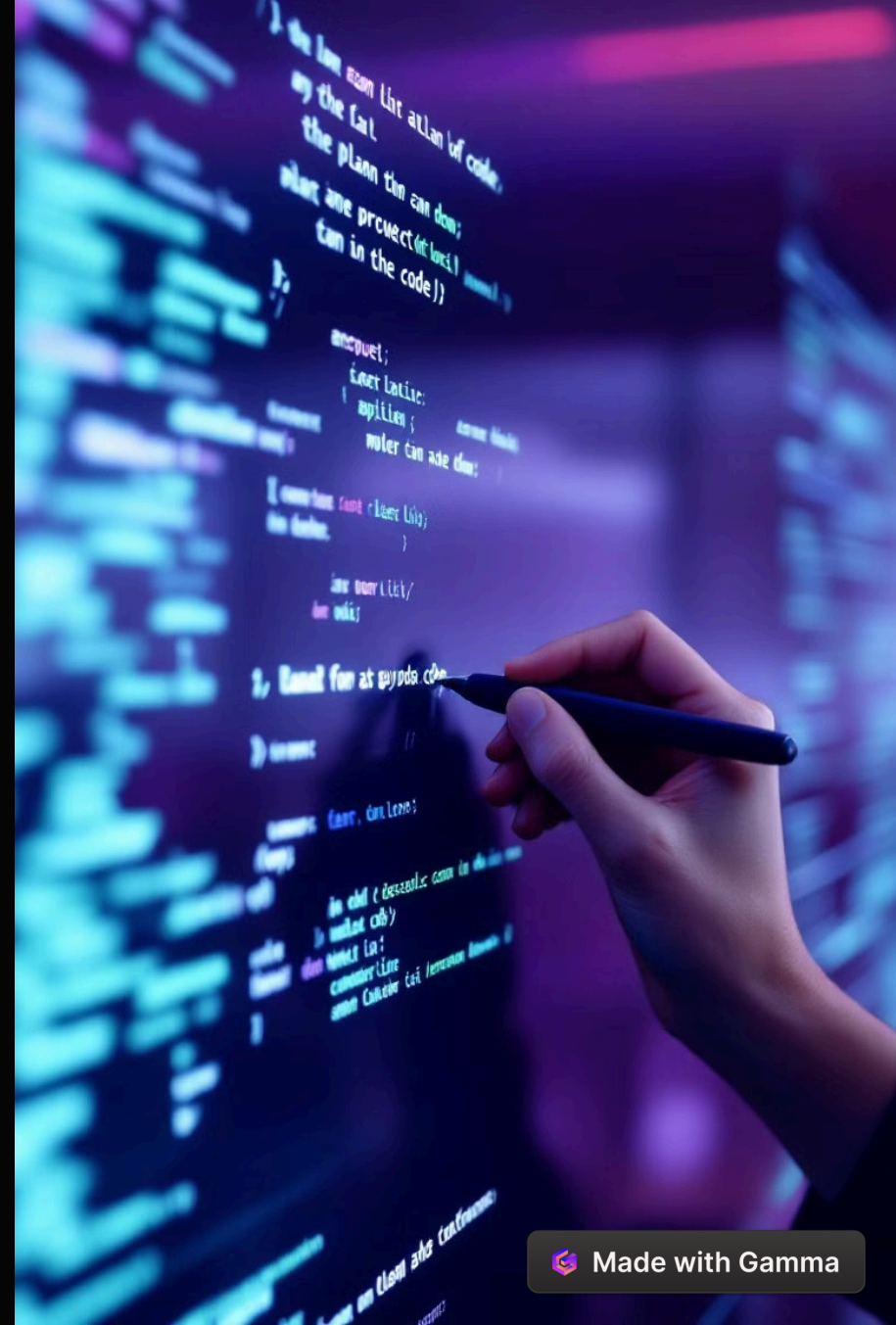
Searching

- Linear search
- Binary search

Pseudocode: Bridging the Gap to Code

An informal, high-level description of an algorithm. It outlines the logic of a program before writing actual code, making it easier to understand, facilitate communication, and helps in debugging.

```
Input: array A of numbers
max = A[0]
for each element in A:
    if element > max:
        max = element
Output: max
```







Flow Charts: Visualizing the Process

A diagram that uses symbols to represent the steps of an algorithm. It's easy to visualize the flow of control and helps in identifying logical errors.

 Start/End

 Process

 Decision

Week 1 Recap & Looking Ahead

Key takeaways: Problem-solving methodology, algorithms, pseudocode, and flow charts. Next up are the C++ basics: Variables, data types, and operators. Implement simple algorithms in C++ to reinforce concepts.

C++ Basics

Variables, data types

Practice

Implement algorithms in C++

Next Week

Diving into C++ syntax

