A Quick Talk About Computer Programming

Dr. Zubair Ahmad

Computer Programming?

The process of planning a sequence of steps(called instructions) for a computer to follow.

Algorithm = Computer Programming?

"How can the problem be solved?"

"How can the solution be implemented and executed by a machine?"

Algorithm

Step 1: Start
Step 2: Set largest = first element in list
Step 3: For each element in the list, compare it with largest
        If larger, update largest
Step 4: Return largest
Step 5: End

Computer Program

```
def find_largest(numbers):
    largest = numbers[0]
    for num in numbers:
        if num > largest:
            largest = num
    return largest
```

Programming Life Cycle Phases

- **Problem-Solving**
  - Analysis and Specification
  - Algorithm/General Solution
  - Design

- **Implementation**
  - Program/Concrete Solution
  - Test Plan

- **Maintenance**
  - Use
  - Maintain

Programming Life Cycle Phases

- **Problem-Solving**
  - Analysis and Specification
  - Algorithm/General Solution
  - Design

- Implementation
  - Program/Concrete Solution
  - Test Plan

- Maintenance
  - Use
  - Maintain

Problem-Solving

Analyze the problem and specify what the solution must do

Develop a general solution(algorithm) to solve the problem

Design your solution

Problem-Solving – Real Life Scenario

Online Shopping - Calculating Total Cost

Analysis and Specification:

Input: A list of item prices, a discount percentage, and a tax percentage.
Example: [50, 30, 20], discount = 10%, tax = 5%.

Output: The total cost after applying the discount and tax.

Algorithm/General Solution:

```
1.Start
2. Sum up all item
prices.
3. Apply the discount by
subtracting the discount
percentage.
4.Apply the tax by
adding the tax
percentage to the
discounted price.
5.Return the total cost.
6.End
```

Design

Programming Life Cycle Phases

- **Problem-Solving**
  - Analysis and Specification
  - Algorithm/General Solution
  - Verification

- **Implementation**
  - Program/Concrete Solution
  - Test Plan

- **Maintenance**
  - Use
  - Maintain

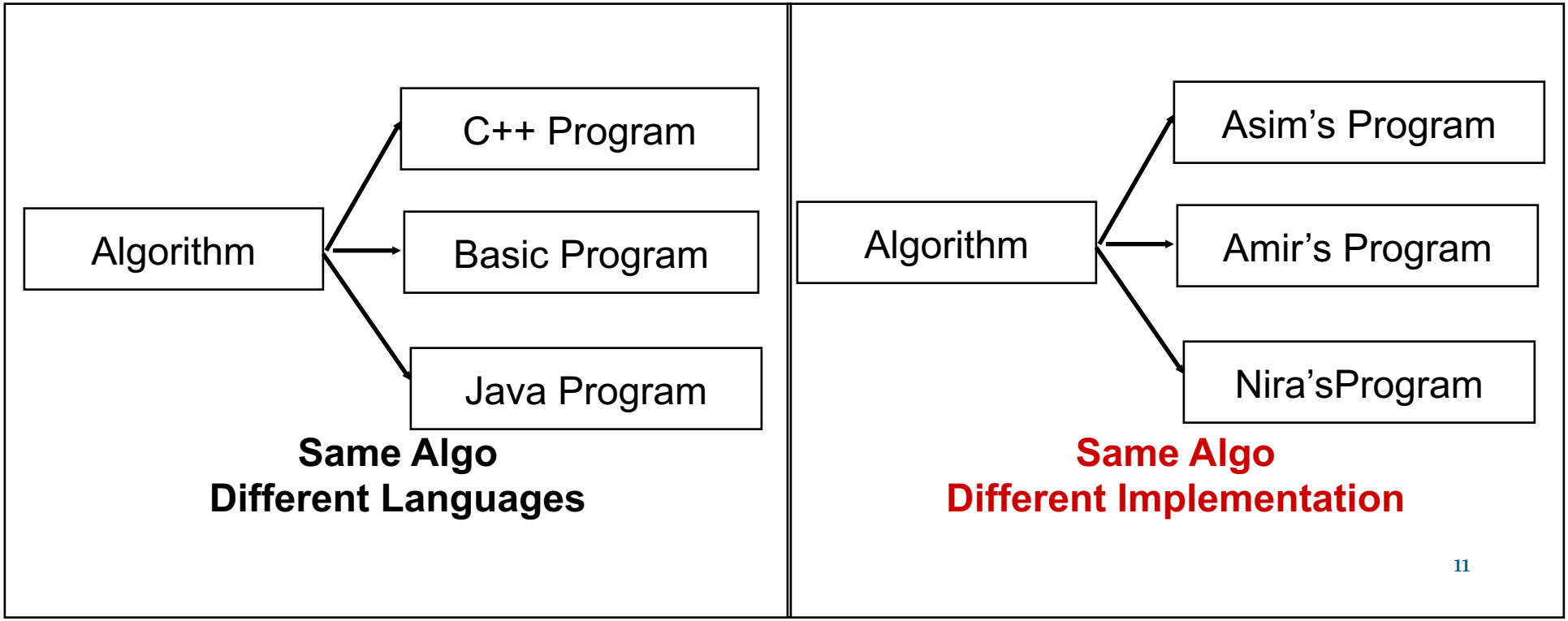
Implementation Phase

Concrete Solution = Computer Language

A programming language is a language with strict grammar rules, symbols, and special words used to construct a computer program

Translating your algorithm into a programming language is called coding

Implementation Phase



Implementation Phase: Test

- Testing your program means running(executing) your program on the computer, to see if it produces correct results

- If it does not, then you must find out what is wrong with your program or algorithm and fix it--this is called debugging

Testing Scenarios

Test Case	Prices	Discount (%)	Tax (%)	Expected Output
Case 1	[50, 30, 20]	10	5	94.50
Case 2	[100, 200, 300]	20	10	528.00
Case 3	[0, 0, 0]	10	5	0.00
Case 4	[100]	0	10	110.00
Case 5	[50, -10, 30]	10	5	Error Handling**

Programming Life Cycle Phases

- **Problem-Solving**
  - Analysis and Specification
  - Algorithm/General Solution
  - Verification
- **Implementation**
  - Program/Concrete Solution
  - Test Plan
- **Maintenance**
  - Use
  - Maintain

Maintenance Phase

- Use and modify the program to meet changing requirements or correct errors that show up in using it
- Maintenance begins when your program is put into use and accounts for the majority of effort on most programs

Testing Scenarios

Test Case	Prices	Discount (%)	Tax (%)	Expected Output
Case 1	[50, 30, 20]	10	5	94.50
Case 2	[100, 200, 300]	20	10	528.00
Case 3	[0, 0, 0]	10	5	0.00
Case 4	[100]	0	10	110.00
Case 5	[50, -10, 30]	10	5	Error Handling**

Problem Solving Techniques

- **Ask questions** -- about the data, the process, the output, error conditions
- **Look for familiar things** -- certain situations arise again and again
- **Solve by analogy** -- it may give you a place to start
- **Use means-ends analysis** -- determine the I/O and then work out the details

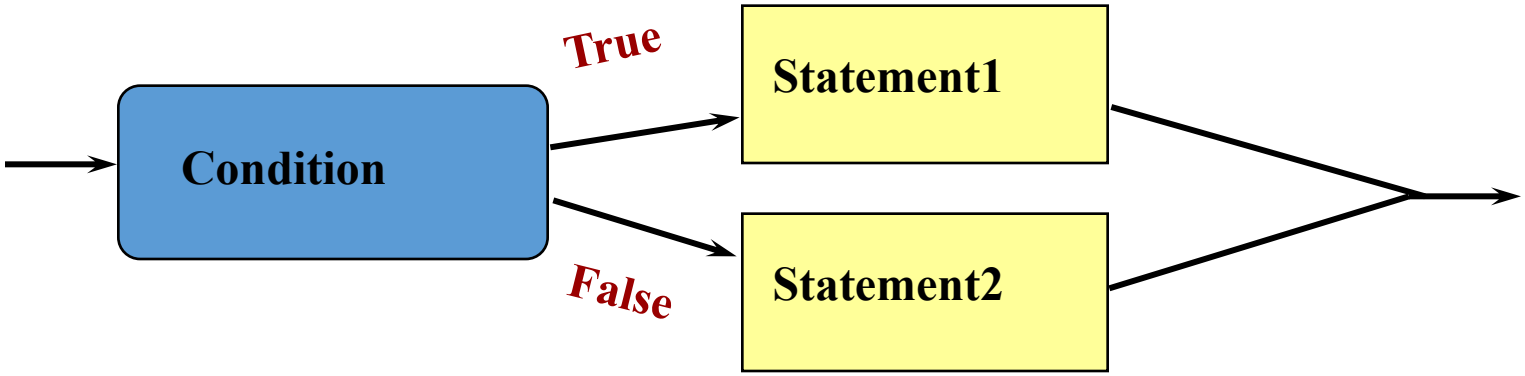
Problem Solving Techniques

- **Divide and conquer** -- break up large problems into manageable units
- **Building-block approach** -- can you solve small pieces of the problem?
- **Merge solutions** -- instead of joining them end to end to avoid duplicate steps
- **Overcome mental block** -- by rewriting the problem in your own words

Basic Control Structures

- A sequence is a series of statements that execute one after another
- A selection(branch) statement is used to determine which of two different statements to execute depending on certain conditions
- A looping(repetition) statement is used to repeat statements while certain conditions are met
- A subprogram is a smaller part of another program; a collection of subprograms solves the original problem

SELECTION (Branch)



SELECTION (Branch)

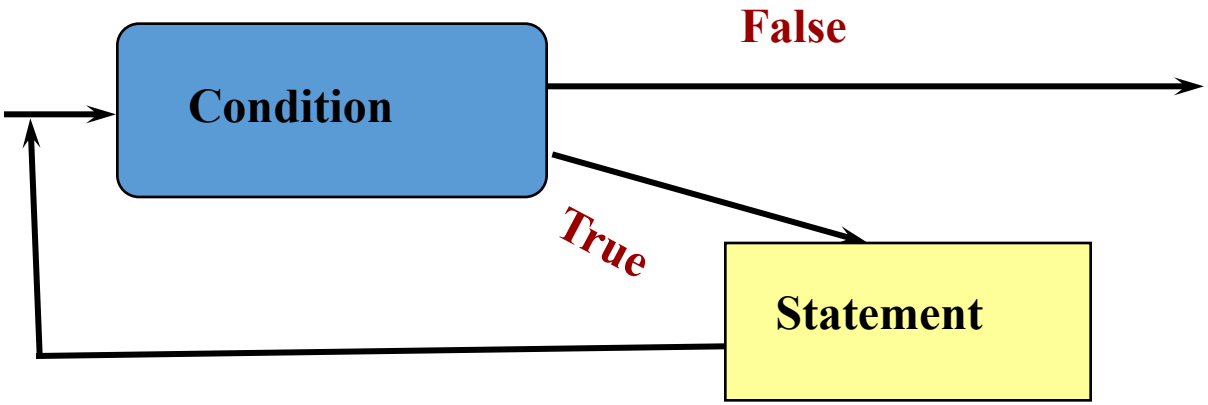
Scenario: Online Shopping Discount
Imagine you're running an online store that offers discounts based on the customer's total purchase amount:

1.If the total purchase is **greater than $100**, the customer gets a **20% discount**.

2.If the total purchase is between **$50 and $100**, the customer gets a **10% discount**.

3.If the total purchase is less than **$50**, there is **no discount**.

LOOP



Scenario: Daily Step Tracker

Imagine you are developing a step tracker app that encourages users to achieve their daily goal of **10,000 steps**. The app checks the steps entered by the user at regular intervals and provides feedback until the goal is reached

- Imagine you have a **house** and an **address** written on a piece of paper. The address tells you where the house is located.

The House:

- The house represents the **actual data** or value stored in memory.
- Example: The house contains a value, like "10 chairs."

The Address:

- The address is not the house itself, but it tells you where the house is.
- In programming, this is like a pointer storing the memory address of a variable.

The Paper with the Address (Pointer Variable):

- The paper itself is the pointer variable. It stores the location (address) of the house, so you know where to find it.

- If someone gives you the address (pointer), you can go to the house and see whats inside (dereferencing the pointer).

- If you lose the paper (pointer), you can no longer find the house, even though the house still exists in its location.

Real-Life Analogy for Pointers

Accessing the Value:

To see the value of the variable (e.g., "10 chairs"), you need to visit the house using the address written on the paper.

Changing the Value:

If you write “15 chairs” inside the house, it means you have updated the variable value using the pointer.

Pointer Arithmetic:

If the houses are arranged in a row (like an array), moving to the next house is like incrementing the pointer to point to the next variable.

Questions?

zahmaad.github.io