Data Structures and Algorithms
(ES221)

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.

Step 1

Step 2

Step 3

Algorithm = Computer Programming?

“How can the problem be solved?”

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

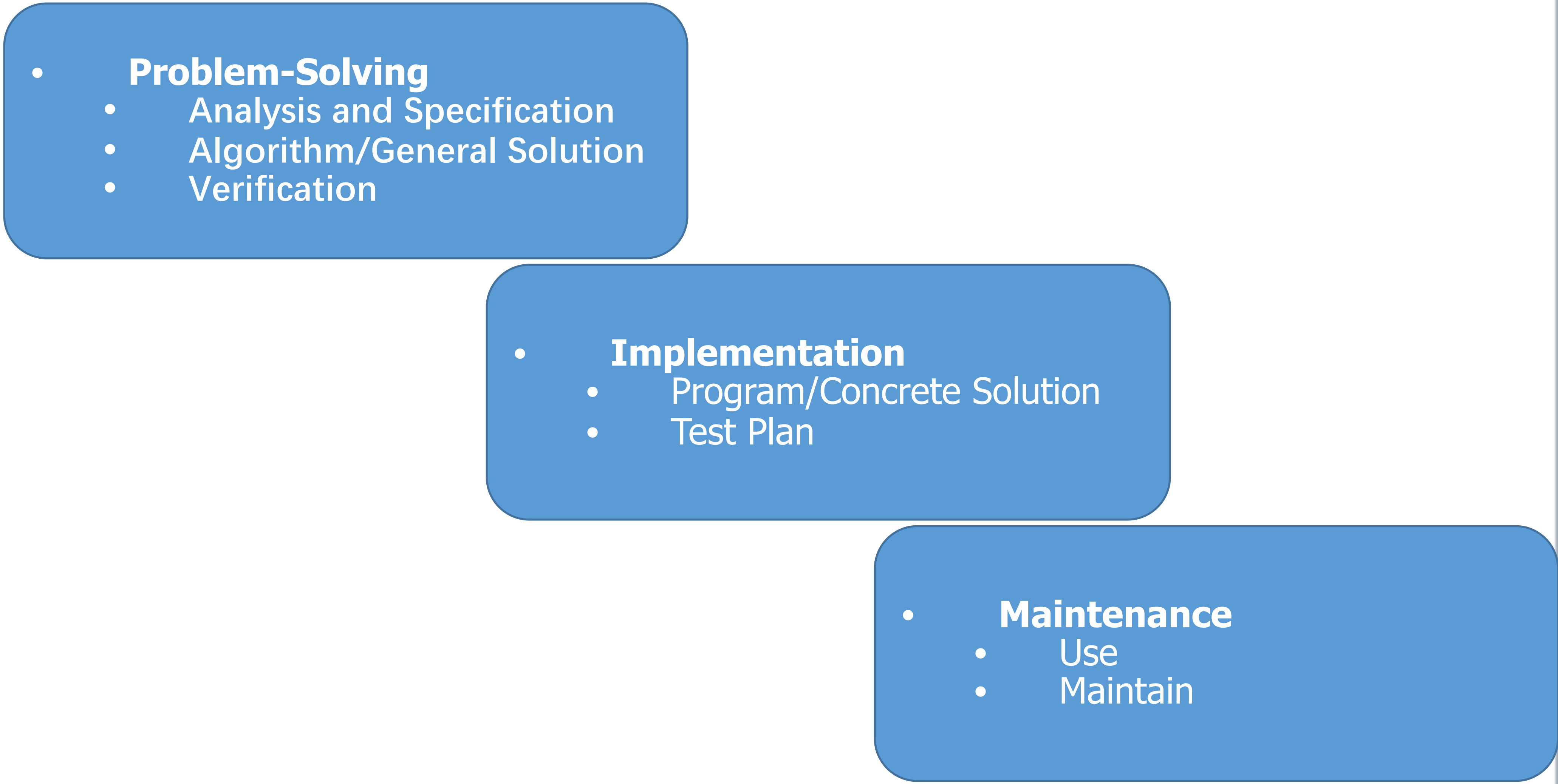
Algorithm

```
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
  - Verification

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

- **Maintenance**
  - Use
  - Maintain

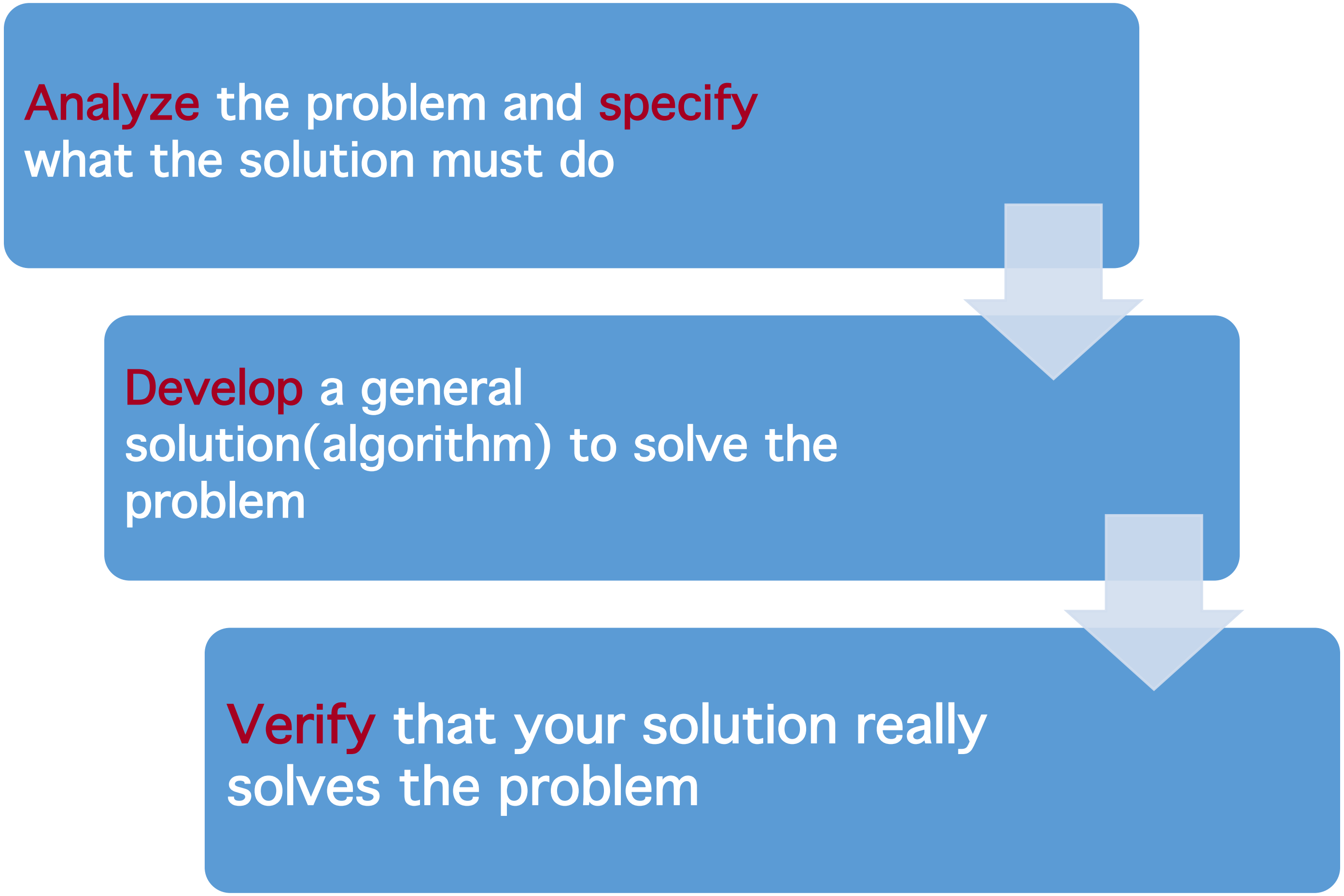
Programming Life Cycle Phases

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

- 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

Verify that your solution really solves the problem

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.

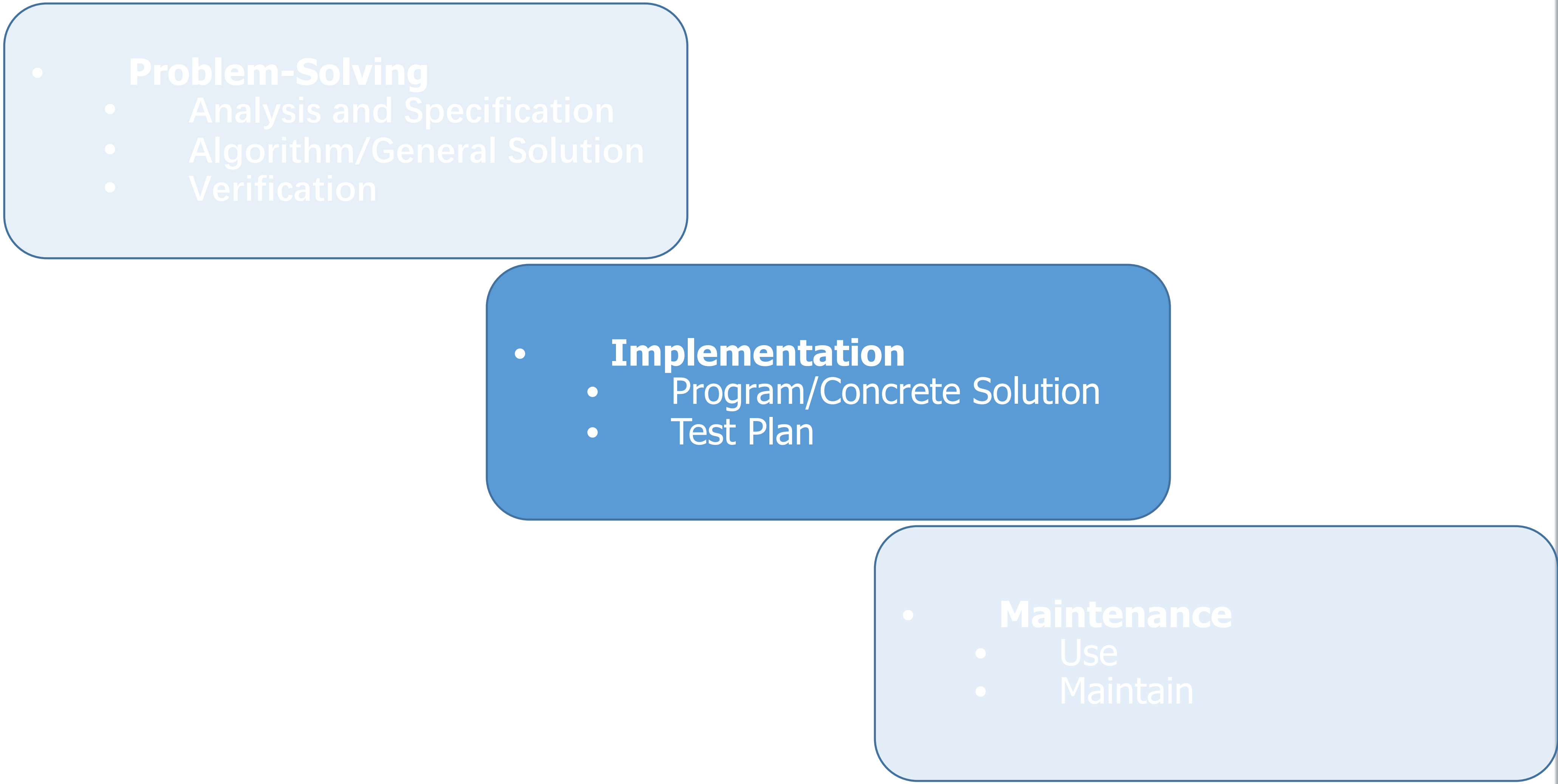
Algorithm/General Solution:

```
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
```

Verification:

Lets Verify it

Programming Life Cycle Phases



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

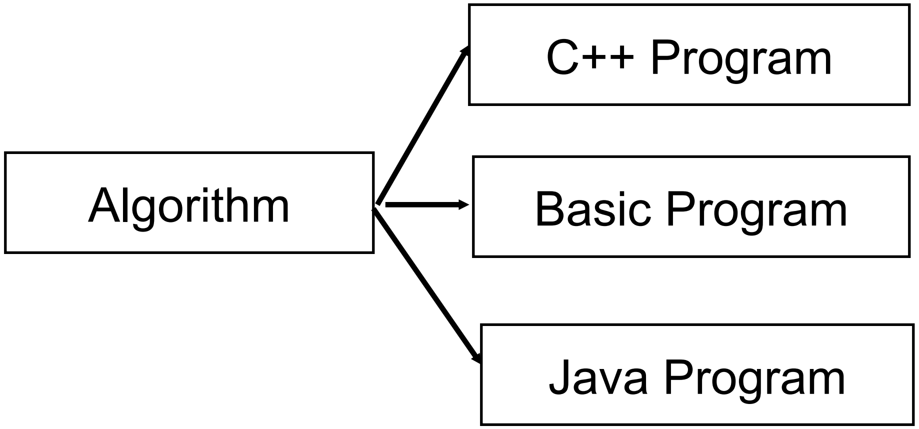
Implementation Phase

Concrete Solution = Computer Language

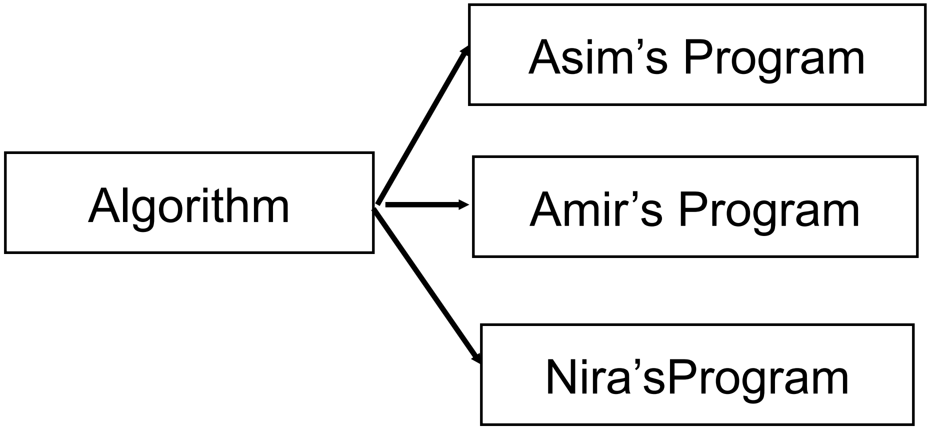
A programming language is a language with strict grammar rules, symbols, and special words used to construct a computer program

Translating your algorithm into a programming language is called coding

Implementation Phase



Same Algo
Different Languages



Same Algo
Different Implementation

Implementation Phase – Code it

```
def calculate_total_cost(prices, discount_percentage,
tax_percentage):
    # Step 1: Calculate the total price of items
    total_price = sum(prices)

    # Step 2: Apply discount
    discount = (discount_percentage / 100) * total_price
    discounted_price = total_price - discount

    # Step 3: Apply tax
    tax = (tax_percentage / 100) * discounted_price
    final_price = discounted_price + tax

    # Step 4: Return the final price
    return final_price

# Example usage:
prices = [50, 30, 20]  # Item prices
discount_percentage = 10  # Discount in %
tax_percentage = 5        # Tax in %

total_cost = calculate_total_cost(prices,
discount_percentage, tax_percentage)
print(f"The total cost is: ${total_cost:.2f}")
```

Lets Code it

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**

- **Lets check Case-5 and modify the code to remove the error**

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

Questions?

zahmaad.github.io