

# Elementary Programming Principles

## Lesson#2 : Program Development Cycle

**Kalema Golooba Ayub**

agkalema@gmail.com

077 259 2826

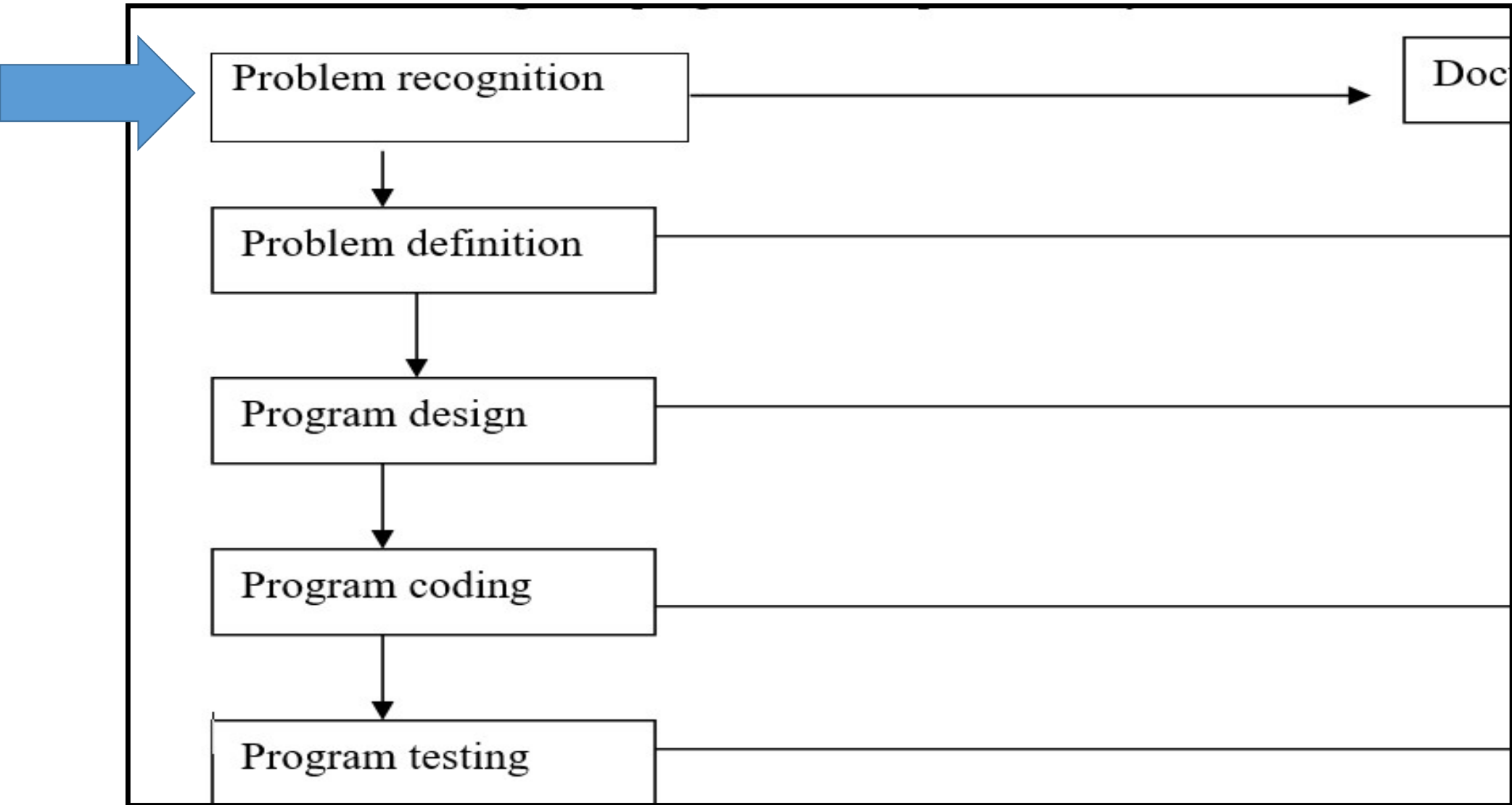
070 169 2724

www.kalsworld.com



# Program Development Cycle

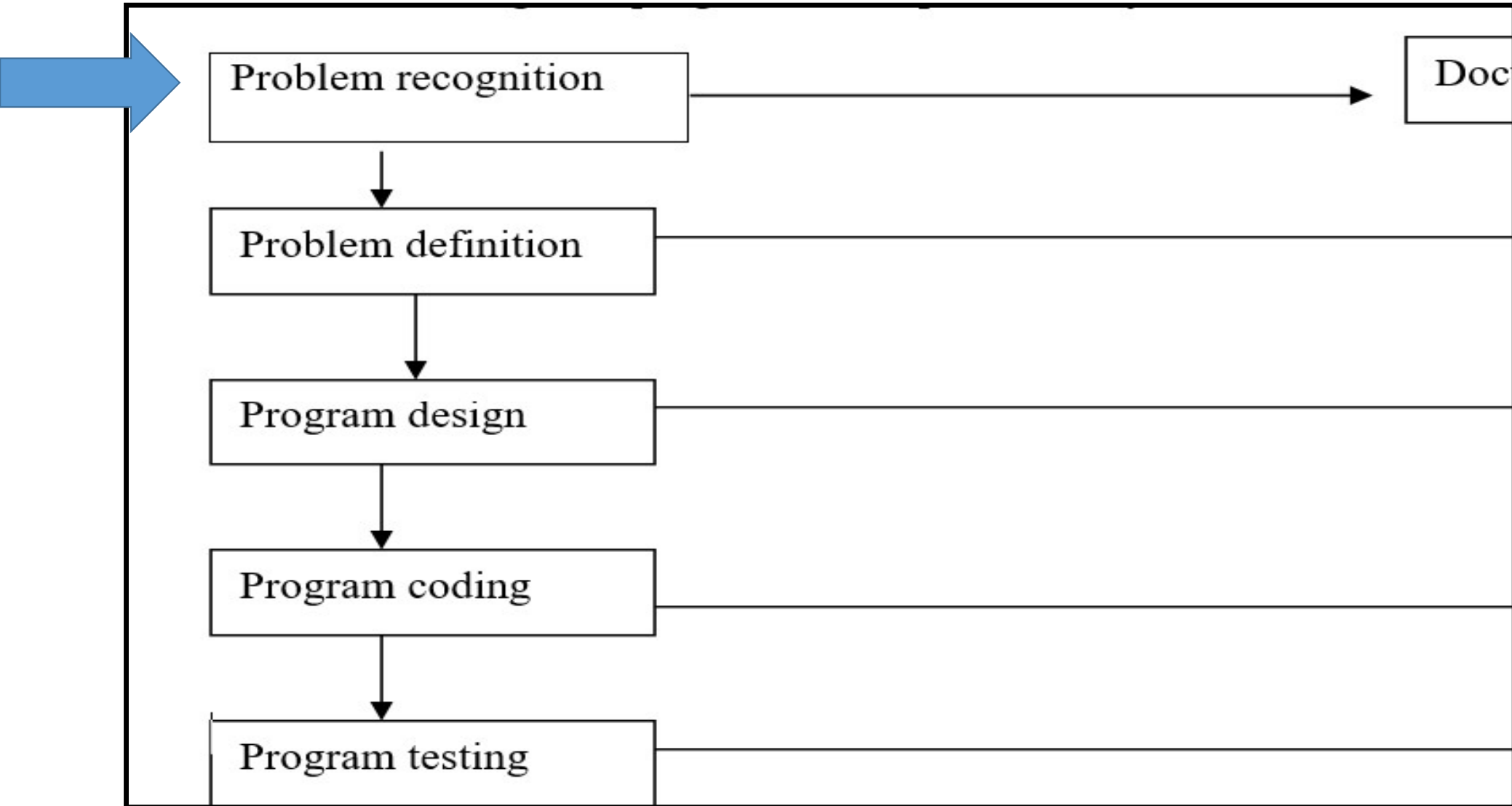
1. Problem recognition/Identification
2. Problem definition/Analysis
3. Program design
4. Program coding
5. Program testing and debugging
6. Implementation and maintenance



# 1. Problem Recognition

- Identify the problem in the environment ....
- 1. Problems** or undesirable situations that prevent an individual or organization from achieving their purpose
  - 2. Opportunity** to improve the current program.
  - 3. A new directive** given by the management/Authorities requiring a change in the status quo

**Example:** A Student has received his examination papers from 3 exams in a subject. He is having trouble arriving at his average grades for the term.

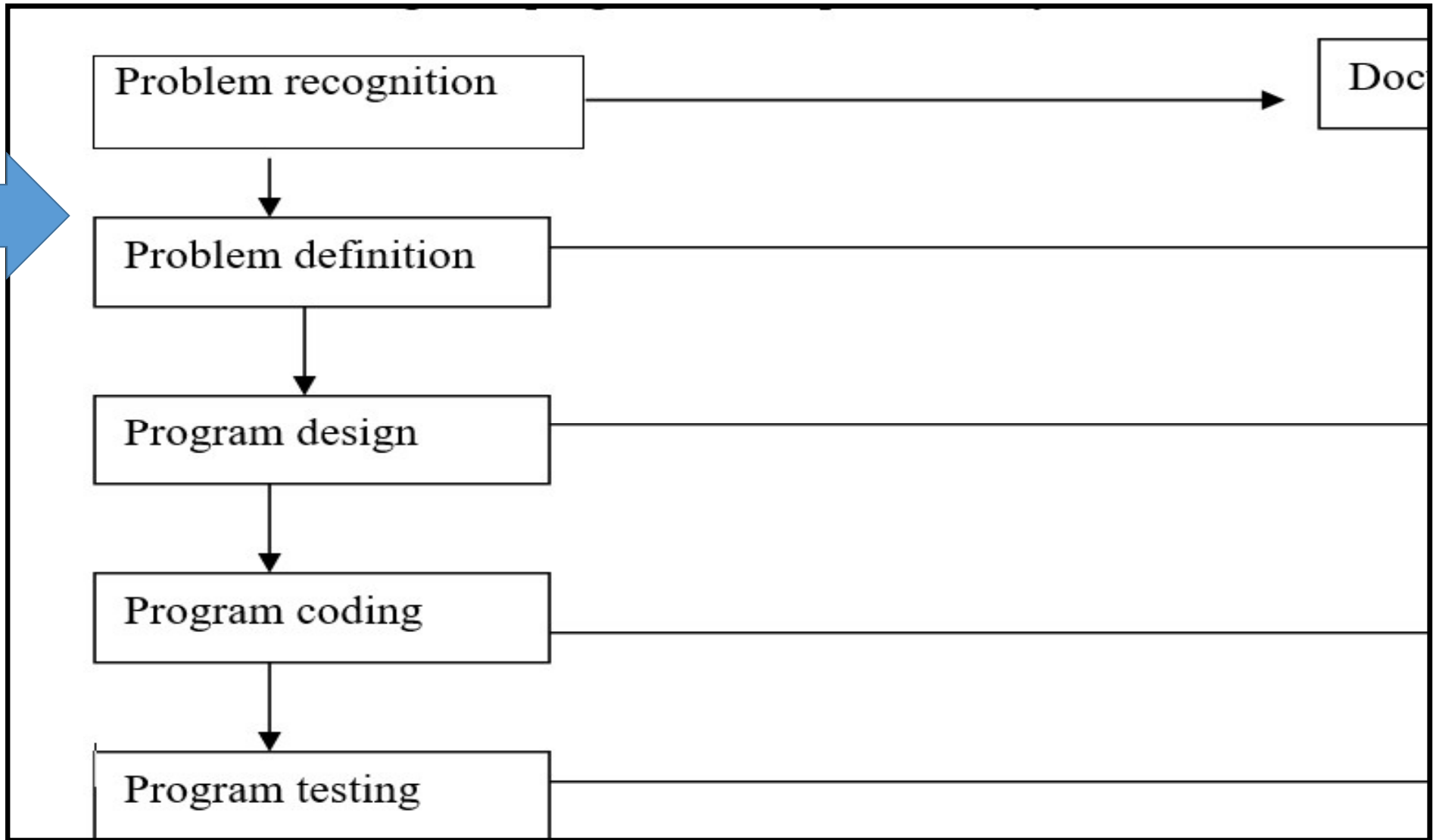


## 2. Problem Definition/Analysis

...Determine or define the likely ...

1. Input (Given Data)
2. Processing activities (How?)
3. The expected output (Required Data)

...Write out a requirements report/document



## 3. Program Design

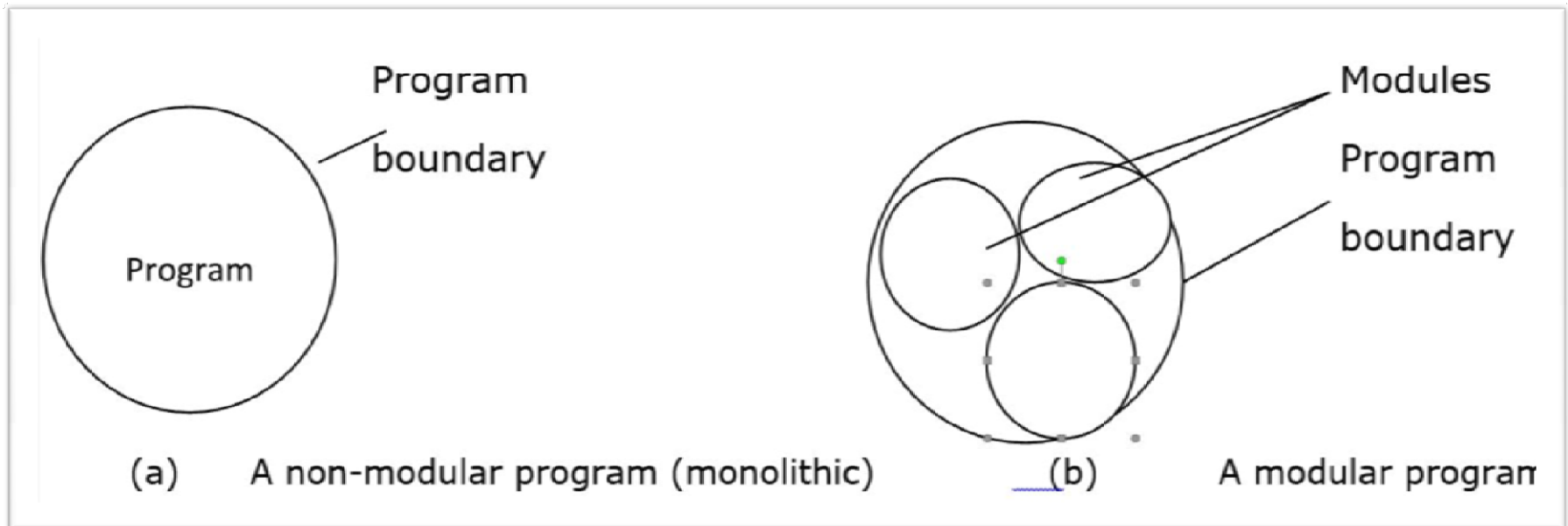
- ..is the actual development of the program's processing or problem solving logic called the **algorithm**
- Algorithm = a definite number of **logical steps** that a program follows in order to solve a problem
- ...Result is a Pseudo code / Flow chart
- Monolithic Vs Modular

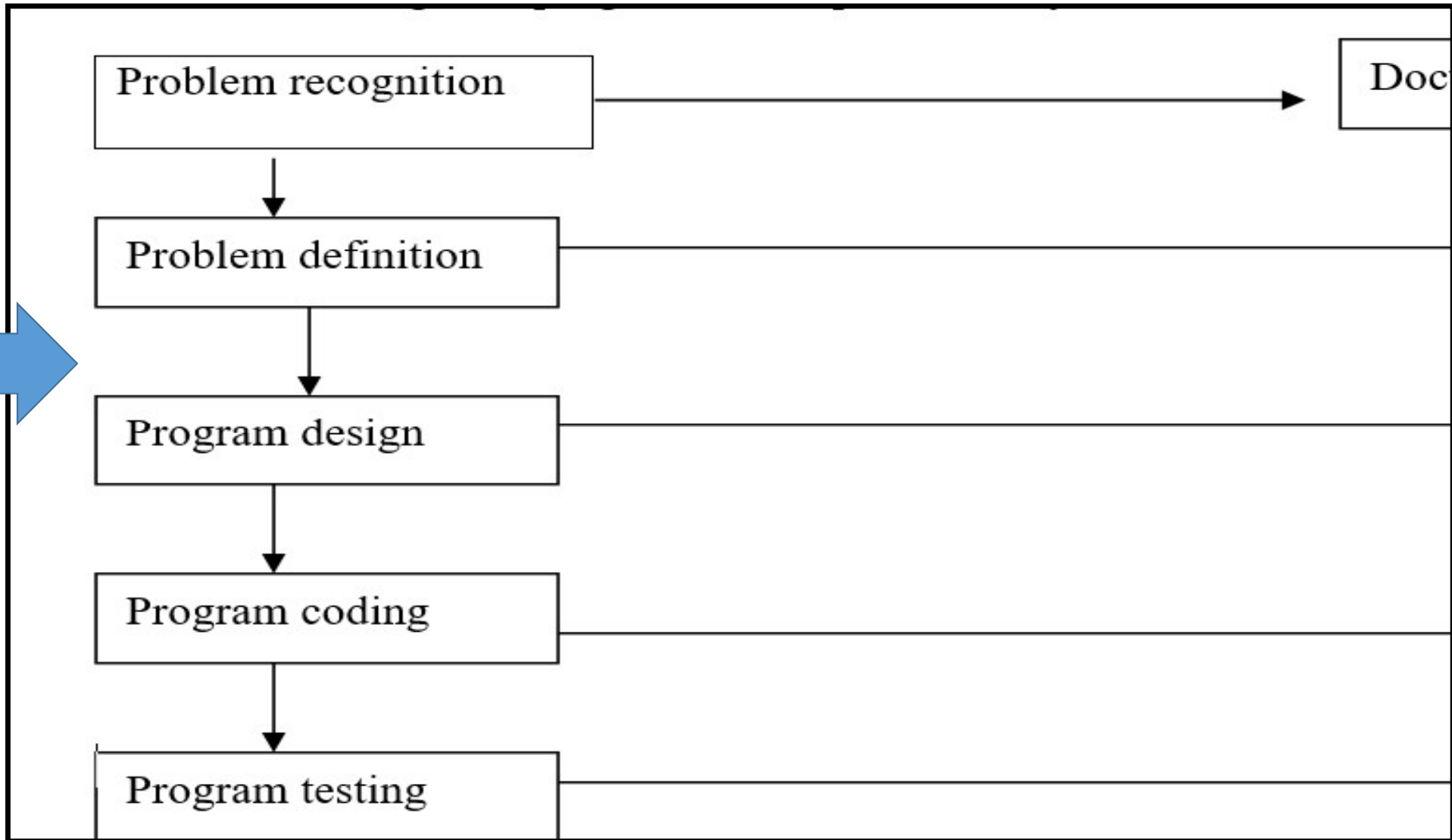
Monolithic = One large block of code



# Monolithic Vs Modular Program Design

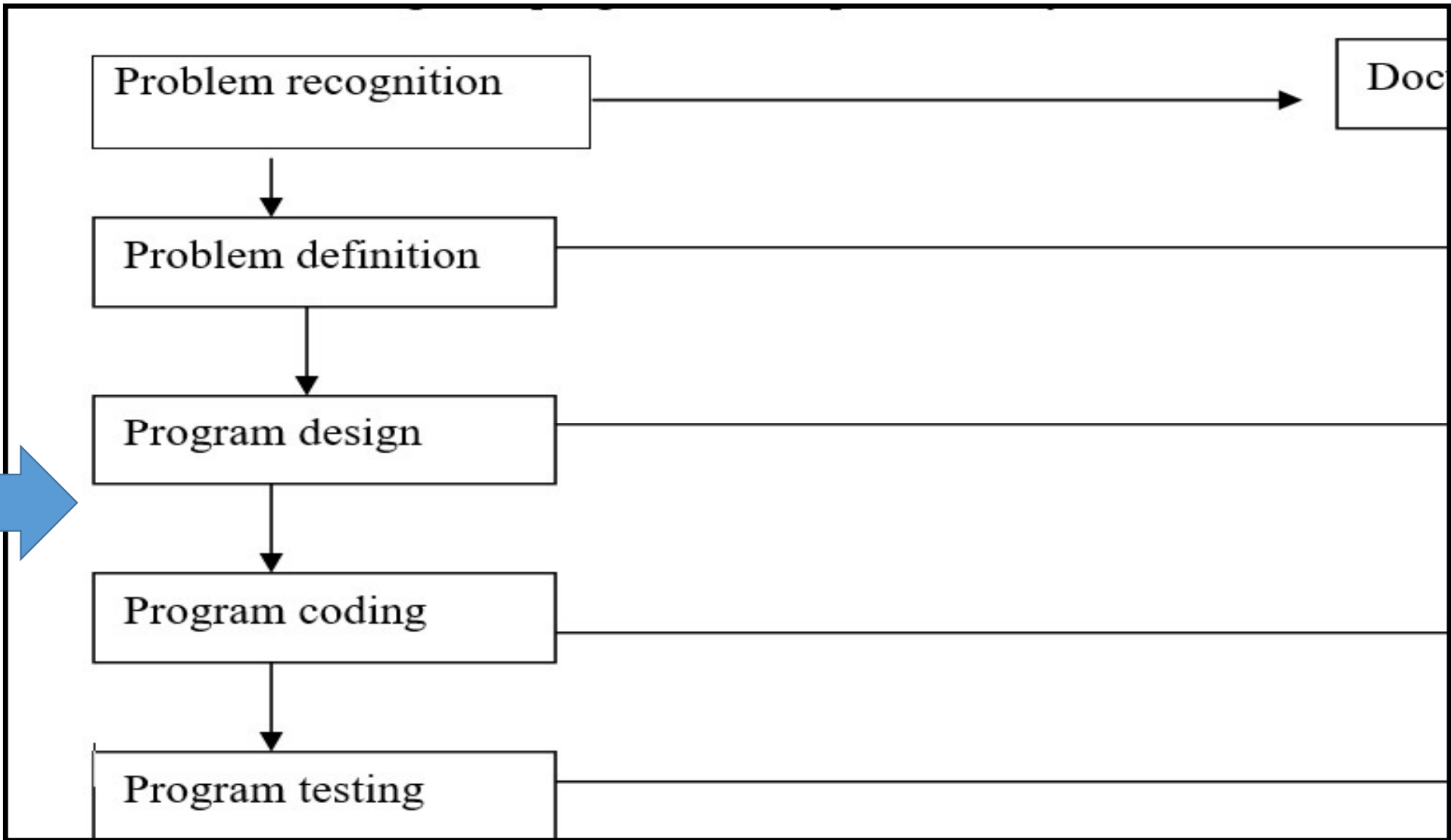
In modular programming, each module performs a specific task. This approach makes a program flexible, easier to read and carry out error correction.





## 4. Program Coding

- ..is the actual process of converting a design model into its equivalent program
- ..is done by typing using particular programming language application software.
- .. end result is a source program (source code) that can be translated into machine readable form for the computer to execute and solve the target problem



# 5. Program Testing/Debugging

- the program is tested in order to detect & correct errors (debug)

- **Syntax Error**

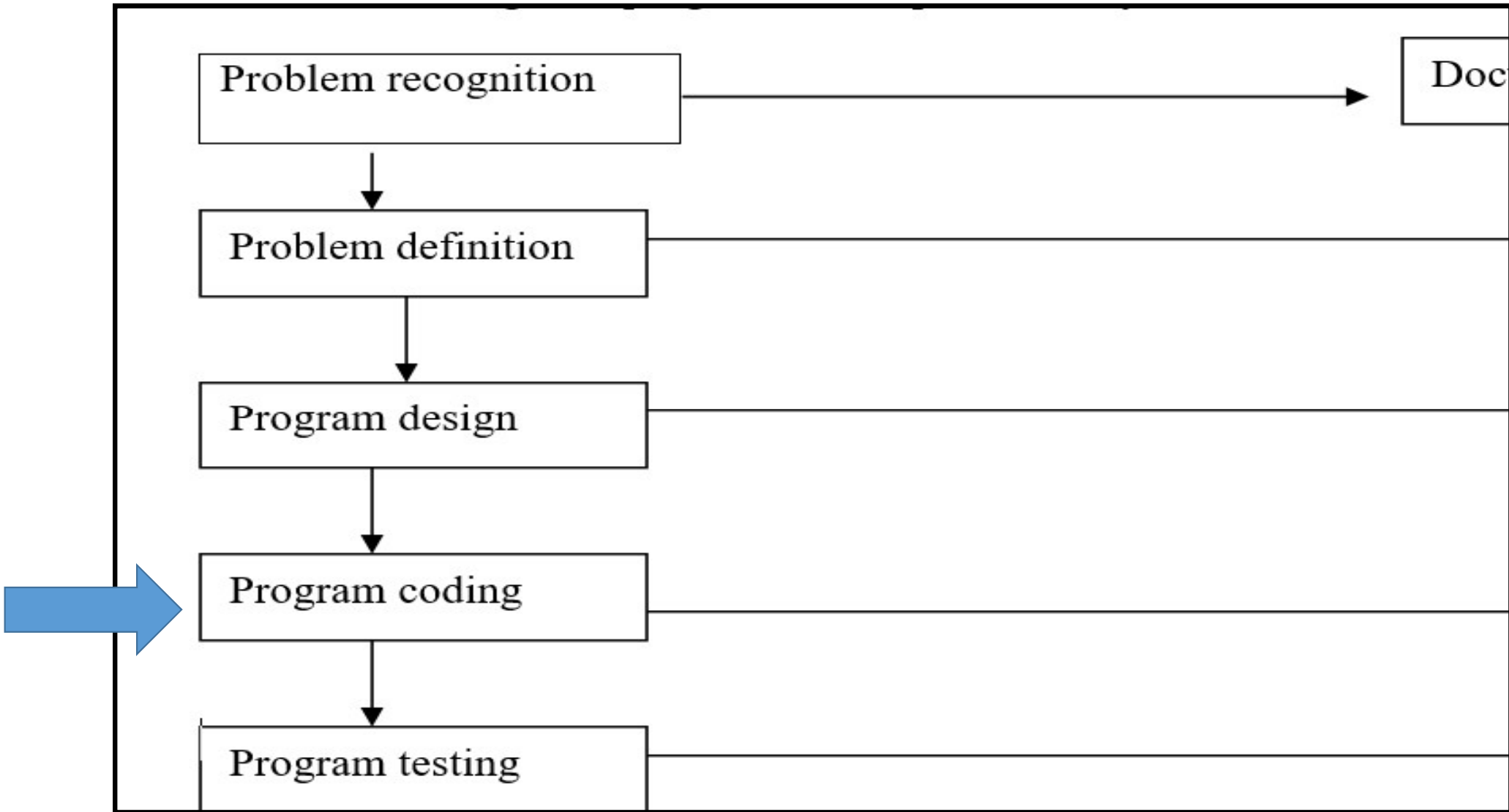
... errors that emanate from improper use of language rules e.g. grammatical mistakes, punctuation, improper naming of variables and misspelling of user defined and **reserved** words

- **Logical Error**

... errors that are not detectable by the translator. The program runs but gives wrong output or halts during execution.

# Methods or error detection

- Testing is the process of running code of a program with the aim of **finding errors**. A good test must discover errors in the code.
1. ***Desk checking (Dry-run/Code walk through)***: It involves going through the program while still on **paper before entering** it in the program editor. This helps the programmer to detect the most obvious syntax and logical errors.
  2. ***Using debugging utilities***: After entering the program in the program editor, you can run the **debugging utilities during translation** to detect **syntax errors** in order to correct them before execution.
  3. ***Using test data***: The programmer carries out **trial runs** of the new program. At each run, the programmer enters various data variations and extremes including data with errors to test whether the system will crash



## 6. Implementation and Maintenance

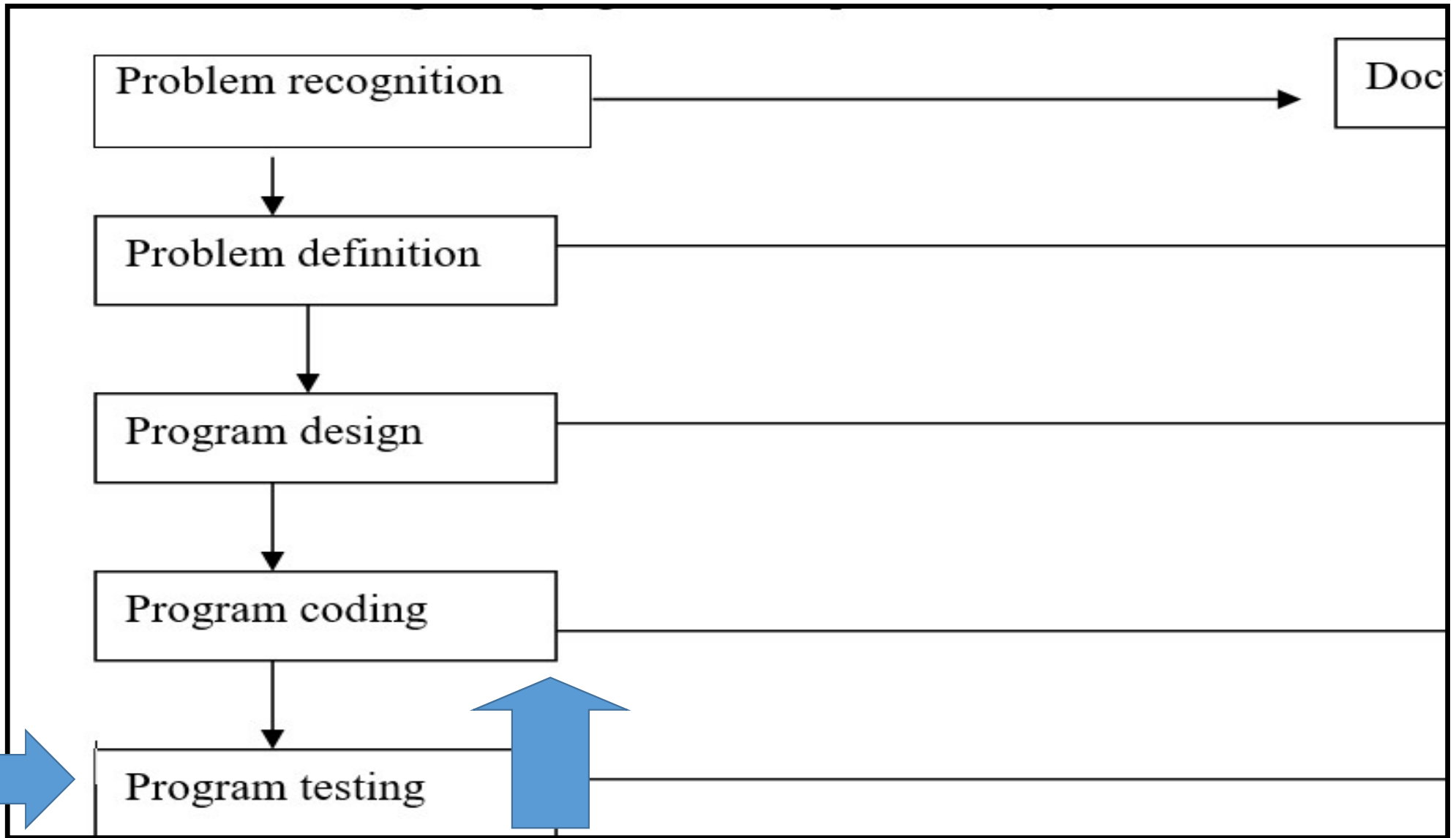
- **Implementation** = actual delivery and installation of the new program ready for use
- **Review and maintenance** = is important because of the errors that may be encountered after the program has been implemented or exposed to extensive use

*++ A program may also fail not because of poor development but also due to poor use*

- Proper training
- Post implementation support of users







# 7. Program Documentation

- Writing down formal support materials
  - Internal Vs external documentation
1. **User oriented** = how to use the program
  2. **Operator Oriented** = How to install and maintain program
  3. **Programmer orientated** = for skilled programmers for future modification

# Review Questions for the topic

1. Give one advantage of compiling a program rather than interpreting it.
2. Outline at least six stages of program development in their respective order.
3. Highlight two disadvantages of monolithic programs.
4. State two advantages of modular programming.
5. In what stage of the development does program documentation fall? Justify your answer.
6. Differentiate between a flowchart and pseudocode.
7. What is a program bug?
8. Explain why it is important to test a program before implementing it.

