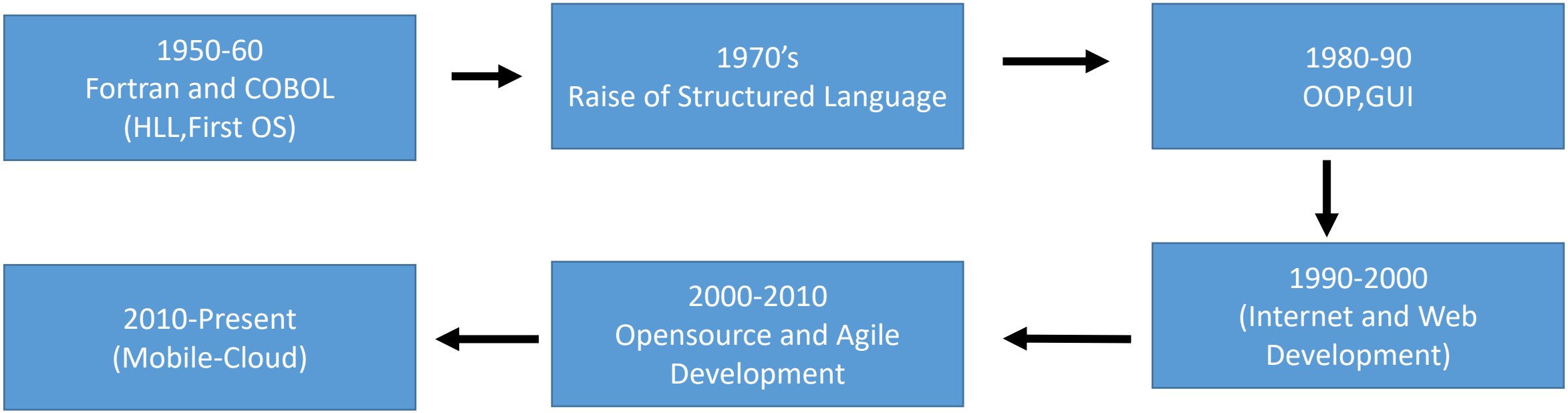
Programming Fundamentals

Topics

- SDLC
- ALGORITHM AND FLOWCHARTS

Evolution of software



Introduction

- SDLC or the Software Development Life Cycle is a process that produces software with the **highest quality** and **lowest cost** in the **shortest time** possible.
- SDLC provides a well-structured flow of phases that help an organization to quickly produce high-quality software which is well-tested and ready for production use.

Building house

- Planning
- Analysis
- Design
- Build
- Check
- Customers experiance
- Feedback

- What , where to build?
- no of floors, cost, man power
- blue print
- Start Building
- Check
- Customer experiance
- Maintanance

SDLC Phases

Phase 1: Requirement collection and analysis

Phase 2: Feasibility study

Phase 3: Design

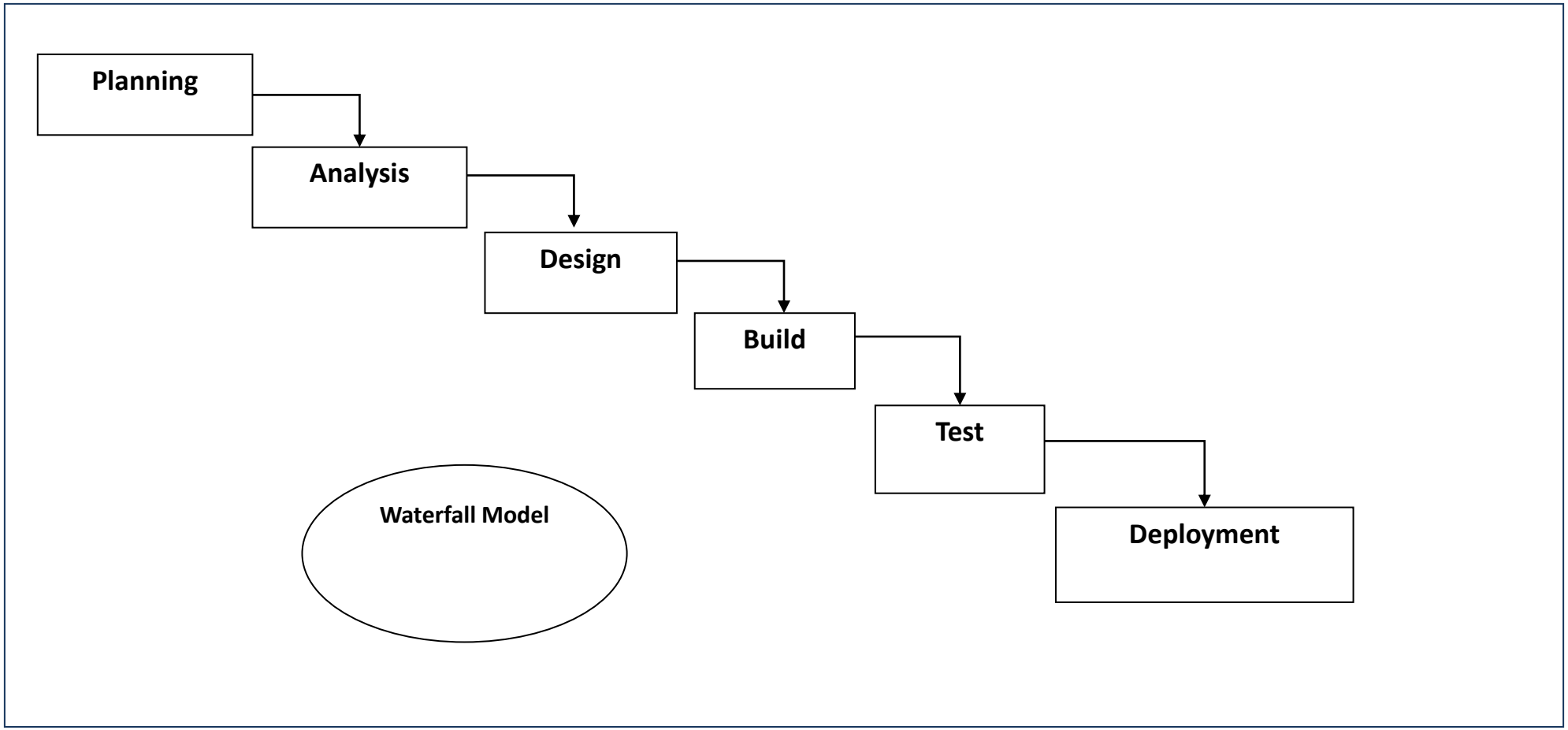
Phase 4: Coding

Phase 5: Testing

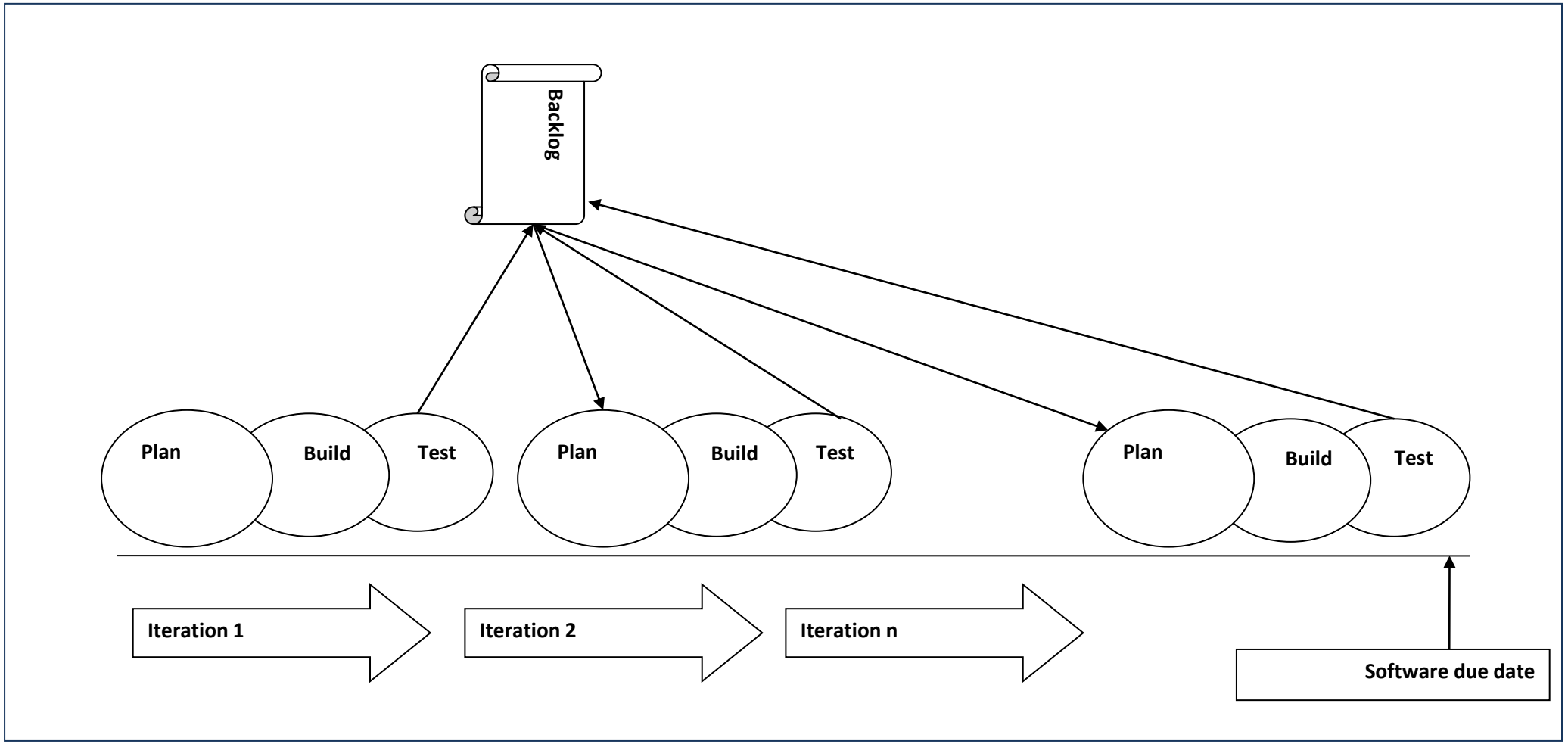
Phase 6: Installation/Deployment

Phase 7: Maintenance

Waterfall Model



Agile Model


Backlog
Plan
Build
Test
Plan
Build
Test
Plan
Build
Test
Iteration 1
Iteration 2
Iteration n
Software due date

Requirement collection and analysis

- The requirement is the first stage in the SDLC process. It is conducted by the senior team members with inputs from all the stakeholders and domain experts in the industry. Planning for the quality assurance requirements and recognition of the risks involved is also done at this stage.

- This stage gives a clearer picture of the scope of the entire project and the anticipated issues, opportunities, and directives which triggered the project.

Feasibility study

- Once the requirement analysis phase is completed the next SDLC step is to define and document software needs. This process conducted with the help of ‘Software Requirement Specification’ document also known as ‘SRS’ document. It includes everything which should be designed and developed during the project life cycle.
- There are mainly five types of feasibilities checks:

- Economic: Can we complete the project within the budget or not?
- Legal: Can we handle this project as cyber law and other regulatory framework/compliances.
- Operation feasibility: Can we create operations which is expected by the client?
- Technical: Need to check whether the current computer system can support the software
- Schedule: Decide that the project can be completed within the given schedule or not.

Design

- In this third phase, the system and software design documents are prepared as per the requirement specification document. This helps define overall system architecture.
- This design phase serves as input for the next phase of the model.

- Brief description and name of each module
- An outline about the functionality of every module
- Interface relationship and dependencies between modules
- Database tables identified along with their key elements
- Complete architecture diagrams along with technology details

Coding

- Once the system design phase is over, the next phase is coding. In this phase, developers start build the entire system by writing code using the chosen programming language.
- In the coding phase, tasks are divided into units or modules and assigned to the various developers.
- It is the longest phase of the Software Development Life Cycle process.

- In this phase, Developer needs to follow certain predefined coding guidelines.
- They also need to use programming tools like compiler, interpreters, debugger to generate and implement the code.

Testing

- Once the software is complete, and it is deployed in the testing environment. The testing team starts testing the functionality of the entire system.
- This is done to verify that the entire application works according to the customer requirement.

- During this phase, QA and testing team may find some bugs/defects which they communicate to developers.
- The development team fixes the bug and send back to QA for a re-test.
- This process continues until the software is bug-free, stable, and working according to the business needs of that system.

Installation/Deployment

- Once the software testing phase is over and no bugs or errors left in the system then the final deployment process starts. Based on the feedback given by the project manager, the final software is released and checked for deployment issues if any.
- Maintenance
- Once the system is deployed, and customers start using the developed system, following 3 activities occur

- Bug fixing – bugs are reported because of some scenarios which are not tested at all
- Upgrade – Upgrading the application to the newer versions of the Software
- Enhancement – Adding some new features into the existing software

ALGORITHM AND FLOWCHART

What is an algorithm?

- The step-by-step procedure to solve any logical and mathematical problem is called an Algorithm.

SCENARIO

- algorithm for making Maggi Noodles‟
- To go for a class picnic
- To make tea/coffee
- To celebrate New Year
- Preparing for exam
- Wedding preparation
- ATM money withdrawal

Algorithm to add two numeric values

Step **1. Declare two variables n1 and n2**

Step **2. Read values in n1 and n2**

Step **3. Declare one more variable sum**

Step **4. Add n1 and n2 and assign result in sum**

Step **5. Print sum**

pseudocode to add two numeric values

```
BEGIN
   PRINT "Enter first number: "
   READ num1

   PRINT "Enter second number: "
   READ num2

   sum ← num1 + num2   // Perform addition

   PRINT "The sum is: ", sum
END
```

pseudocode

- Pseudo code is a simple way of writing programming code in English
- Pseudo code is simply an implementation of an algorithm in the form of
- annotations and informative text written in plain English.
- Pseudo code is not actual programming language
- It has no syntax like any of the programming language and thus can’t
- be compiled or interpretedby the computer.
- “While understanding pseudo code is usually not difficult, writing it can be a challenge”

Why use pseudo code at all?

- If we write an algorithm in English, the description may be at so high a level that it is difficult to analyze the algorithm and to transform it into code.

- **1. Input / Output**
  - READ – To take input from the user
  - INPUT – Alternative to READ
  - WRITE – To display output
  - PRINT – Alternative to WRITE
  - DISPLAY – Another alternative for output
- **2. Control Structures**
  - Conditional Statements (Decision Making)
  - IF condition THEN – Starts a conditional block
  - ELSE IF condition THEN – Additional condition
  - ELSE – Alternative execution path
  - ENDIF – Ends the conditional block

3. Looping Statements (Iteration)

- FOR variable = start TO end DO – For loop
- WHILE condition DO – While loop
- REPEAT UNTIL condition – Repeat loop (executes at least once)
- ENDWHILE – Ends the while loop
- ENDFOR – Ends the for loop
- **3. Operators**
- AND, OR, NOT – Logical operators
- = – Assignment operator
- +, -, *, /, % – Arithmetic operators
- >, <, >=, <=, ==, != – Comparison operators

4. Function / Procedure Definition

- FUNCTION function_name(parameters) – Defines a function
- CALL function_name(arguments) – Calls a function
- RETURN value – Returns a value from a function
- END FUNCTION – Ends the function
- **5. Miscellaneous**
- BEGIN / START – Start of the pseudocode
- END / STOP – End of the pseudocode
- COMMENT or // – To add comments for explanation

- Common Action Keywords:
- Input:
  - READ
  - OBTAIN
  - GET
- Output:
  - PRINT
  - DISPLAY
  - SHOW
- Compute:
  - COMPUTE
  - CALCULATE
  - DETERMINE
- Initialize:
  - SET
  - INIT

- Set area
- READ the r
- COMPUTE area=3.14*r*r
- PRINT area
- stop

- IF-THEN-ELSE
- The general form is:
- IF condition THEN
- sequence 1
- ELSE
- sequence 2
- ENDIF
- IF HoursWorked > NormalMax THEN
- Display overtime message
- ELSE
- Display regular time message
- ENDIF

- The general form is:
- WHILE condition
- sequence
- ENDWHILE
- The loop is entered only if the condition is true.
- WHILE Population < LimitCompute
- Population as Population + Births – Deaths
- ENDWHIL

WHILE

- The general form is:
- WHILE condition
- sequence
- ENDWHILE
- The loop is entered only if the condition is true.
- WHILE Population < LimitCompute
- Population as Population + Births – Deaths
- ENDWHILE

CASE

- The general form is:
- CASE expression OF
- condition 1 :
- sequence 1
- condition 2 :
- sequence 2
- …
- condition n :
- sequence n
- OTHERS:
- default sequence
- ENDCASE

- CASE grade OF
- A :
- points = 4
- B :
- points = 3
- C :
- points = 2
- D :
- points = 1
- OTHERS :
- points = 5
- ENDCASE

EXCEPTION HANDLING

- BEGIN
- statements
- EXCEPTION
- WHEN exception type
- statements to handle exception
- WHEN another exception type
- statements to handle exception
- END

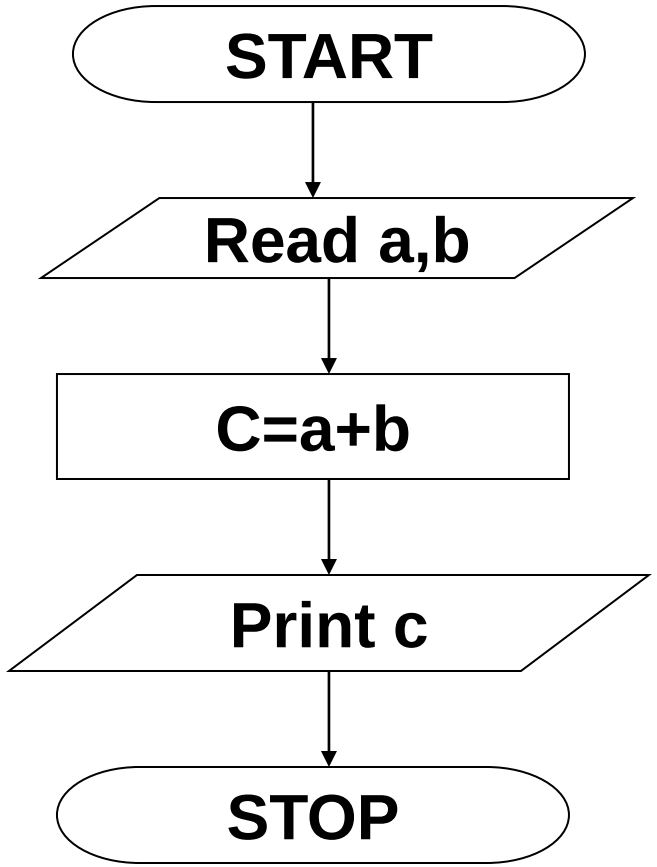
Flowcharts

- A **flowchart** is a **visual representation of a process**, showing the steps or actions in sequential order using **symbols and arrows**. It's used to help people understand, analyse, or improve a process.

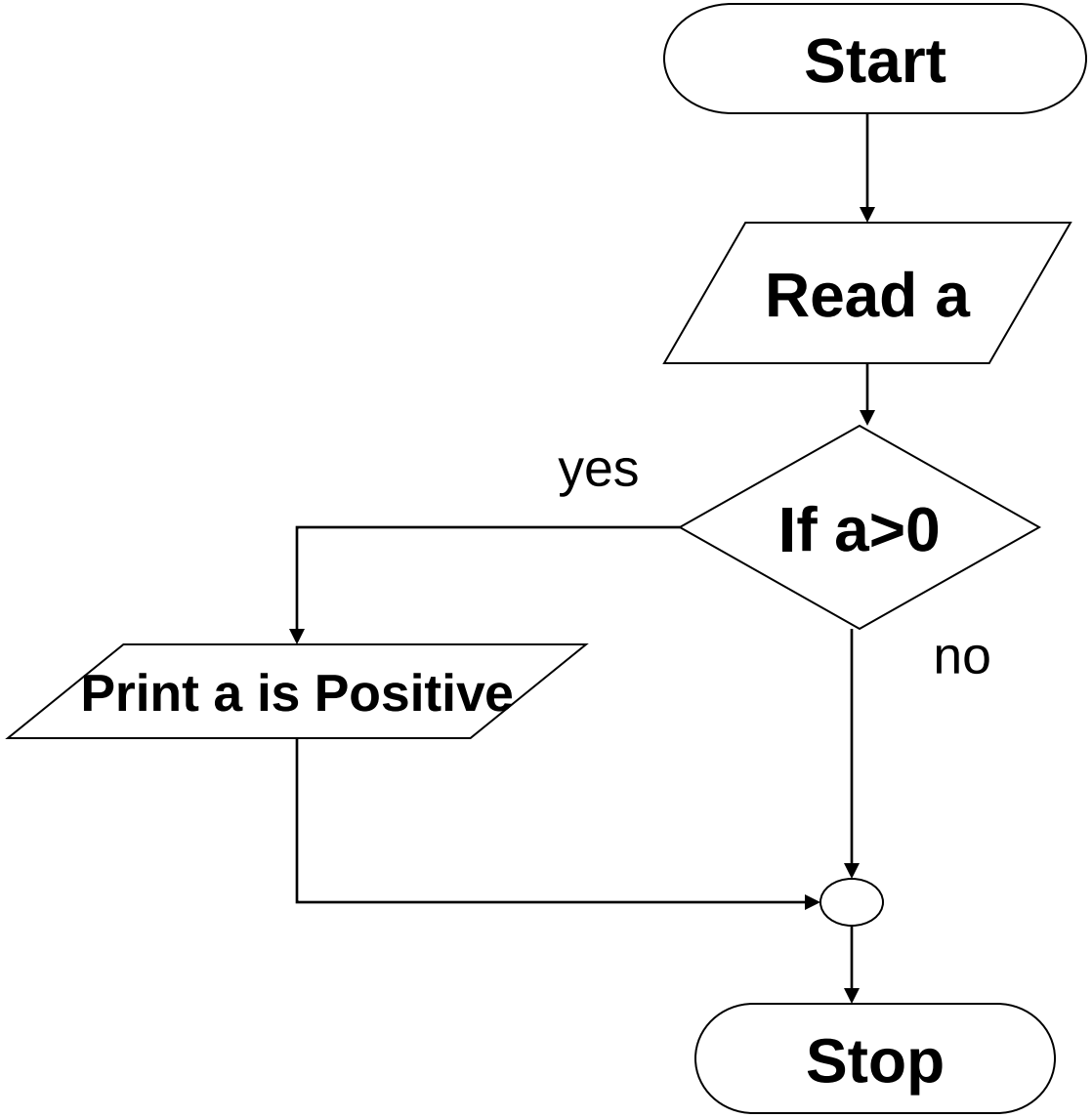
Symbols

Symbol	Symbol Name	Purpose
	Start/Stop	Used at the beginning and end of the algorithm to show start and end of the program.
	Process	Indicates processes like mathematical operations.
	Input/ Output	Used for denoting program inputs and outputs.
	Decision	Stands for decision statements in a program, where answer is usually Yes or No.
	Arrow	Shows relationships between different shapes.
	On-page Connector	Connects two or more parts of a flowchart, which are on the same page.
	Off-page Connector	Connects two parts of a flowchart which are spread over different pages.

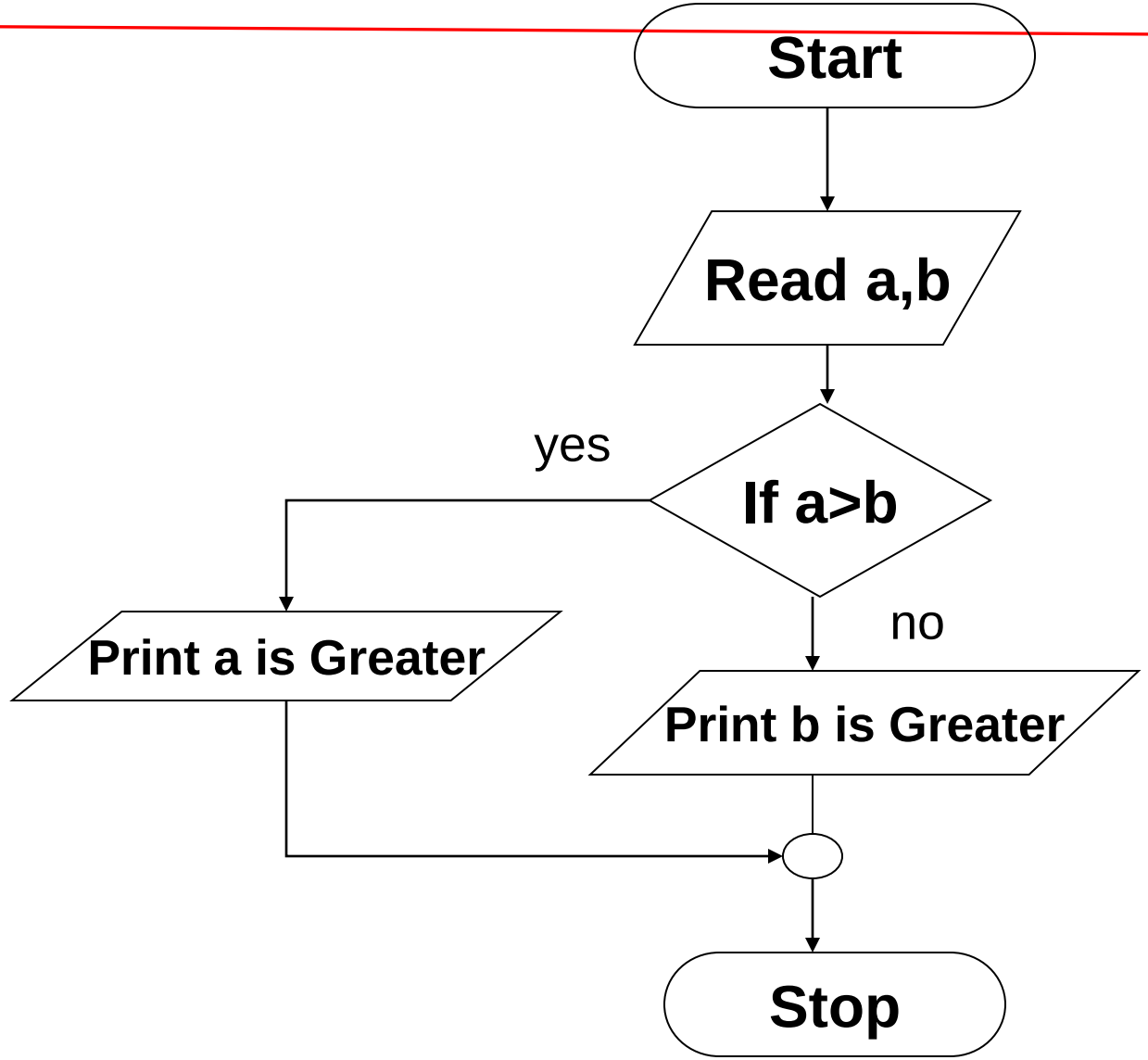
Example



Example



Example

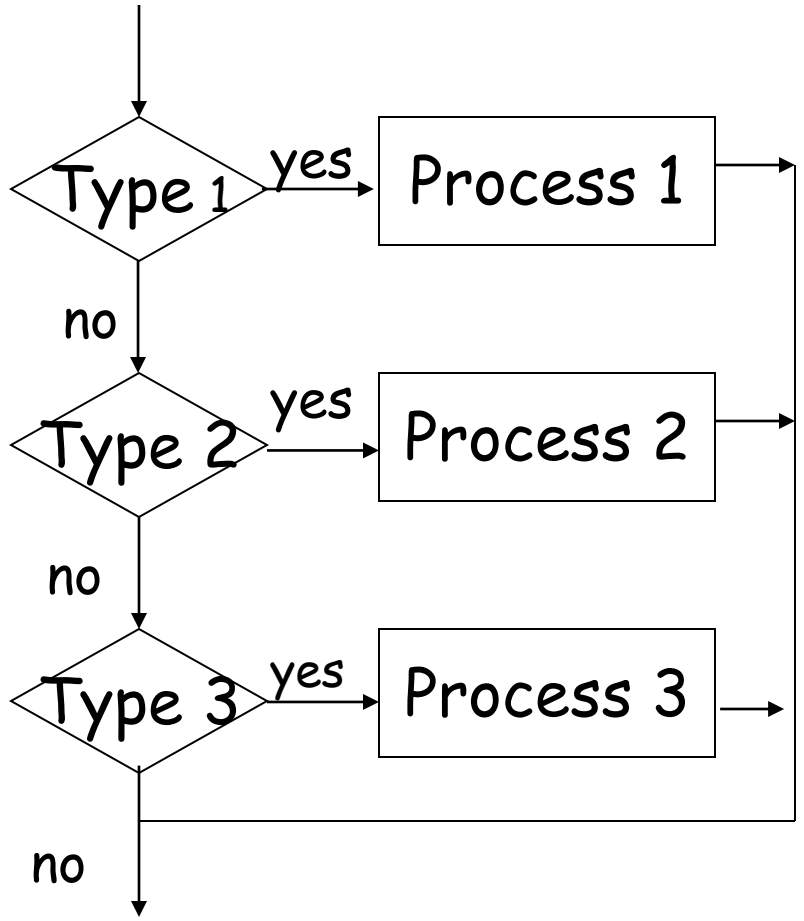

Start
Read a,b
yes
If a>b
no
Print a is Greater
Print b is Greater
Stop

CASE structure

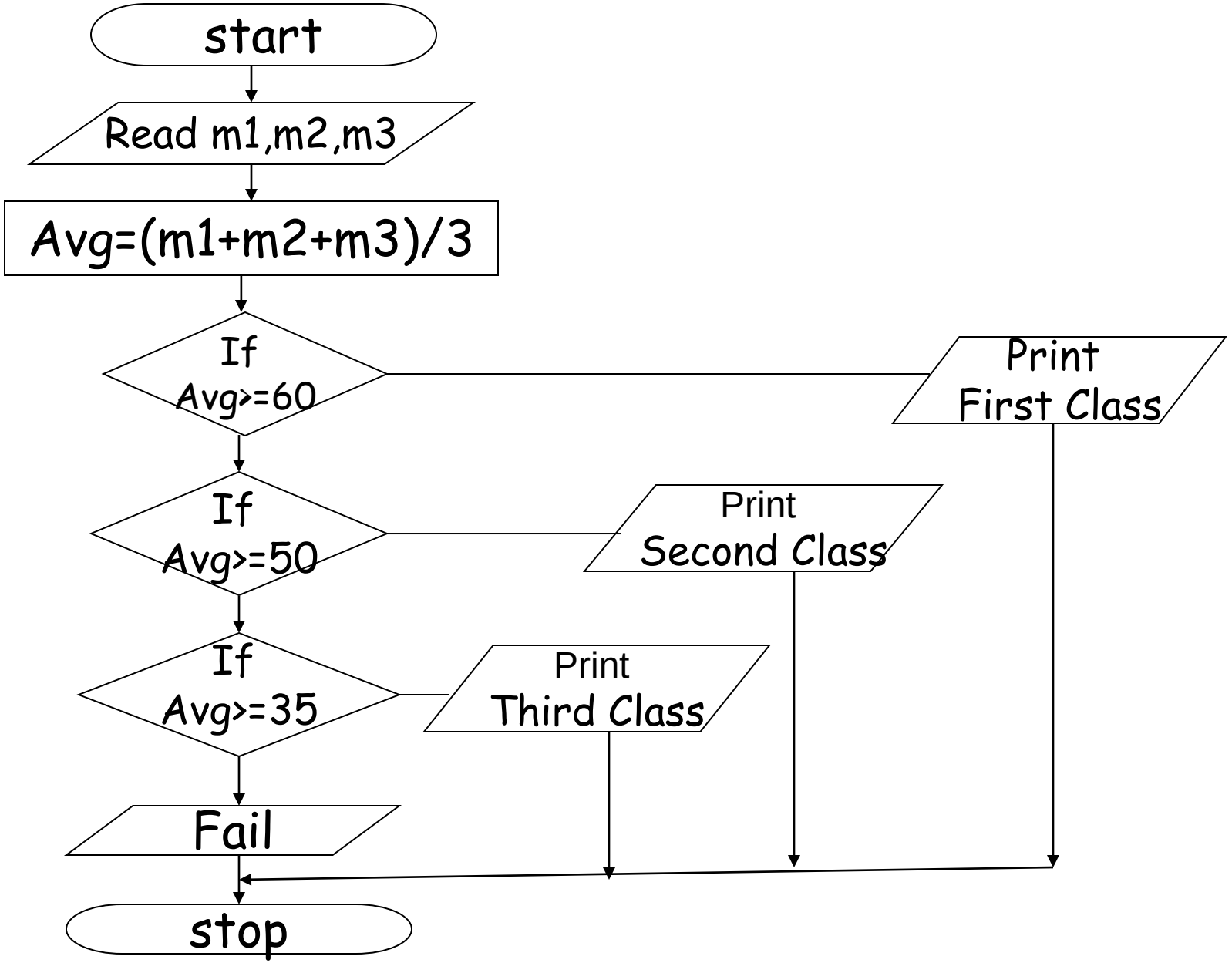
Pseudocode

```
        .
        .
CASE Type
Case Type-1:
        Process 1
Case Type-2:
        Process 2
        .
        .
Case Type-n:
        Process n
        .
        .
END CASE
```

Flow chart



Example: Finding the Grade



Looping control structure

- It is used to execute some instructions several time based on some condition.
- WHILE loop
- Do…WHILE loop etc.,

WHILE Loop

Pseudocode

.

.

WHILE condition

.

.

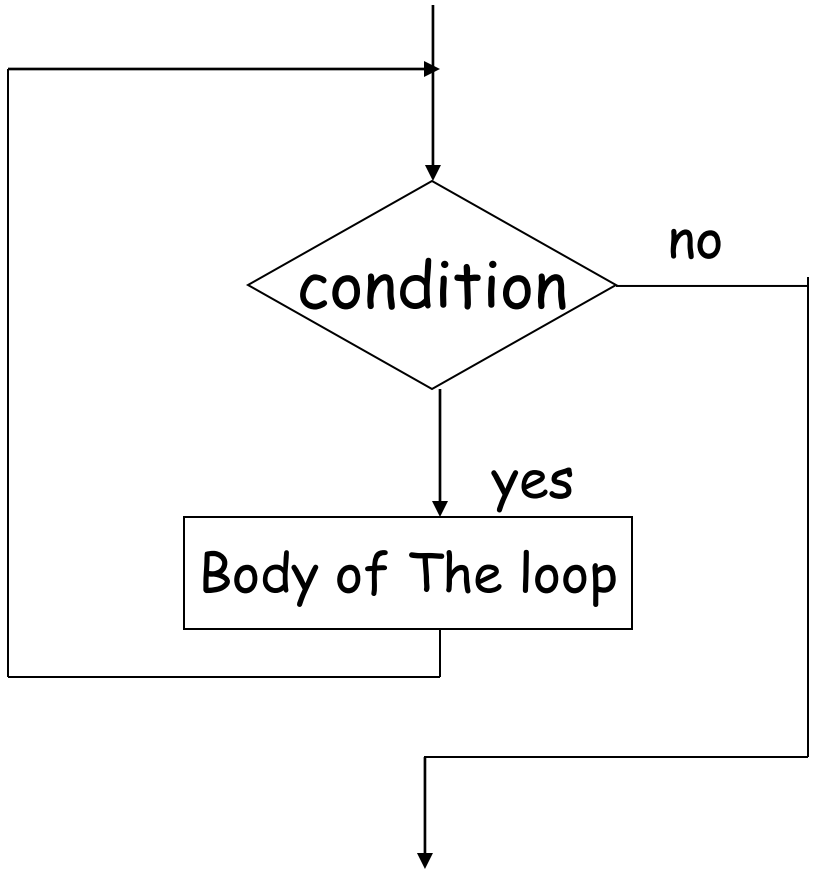
Body of the loop

.

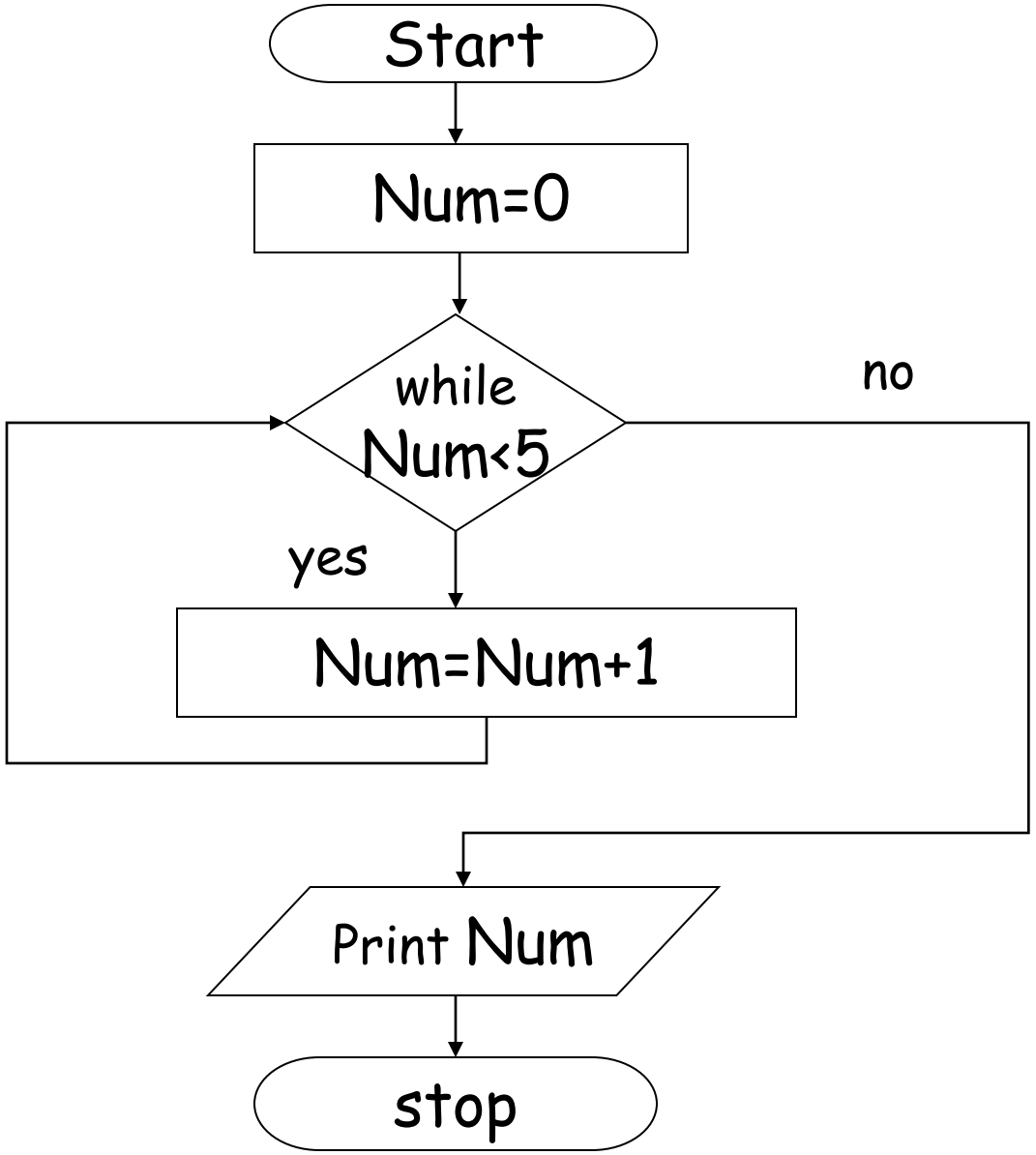
.

END WHILE

Flow chart



Example



DO…WHILE Loop

Pseudocode

DO

.

.

Body of the loop

.

.

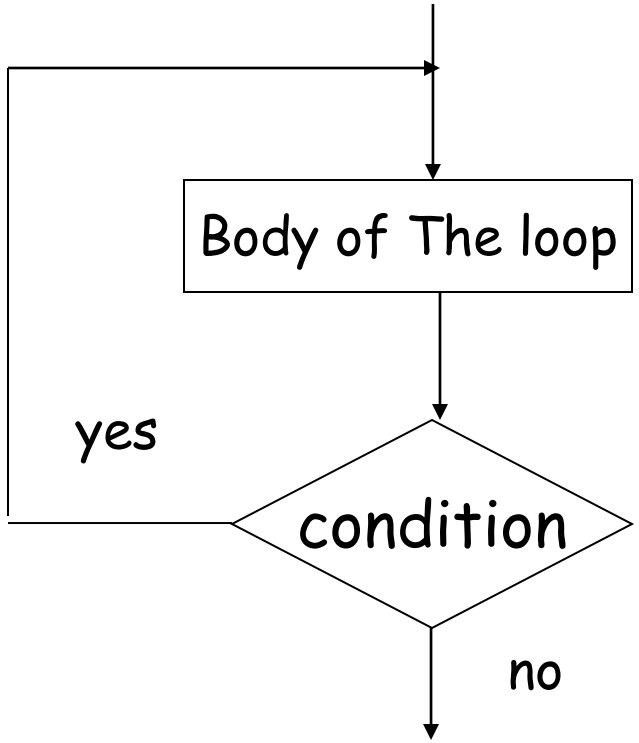
WHILE condition

.

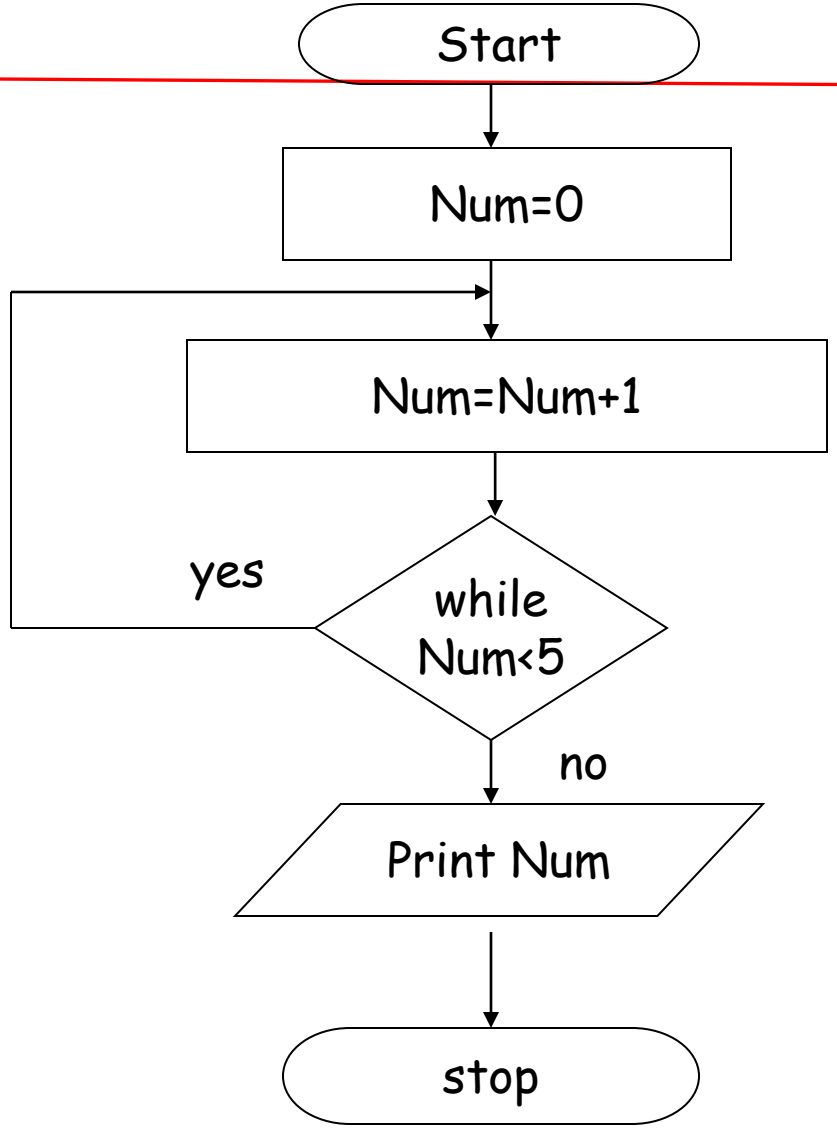
.

END WHILE

Flow chart



Example



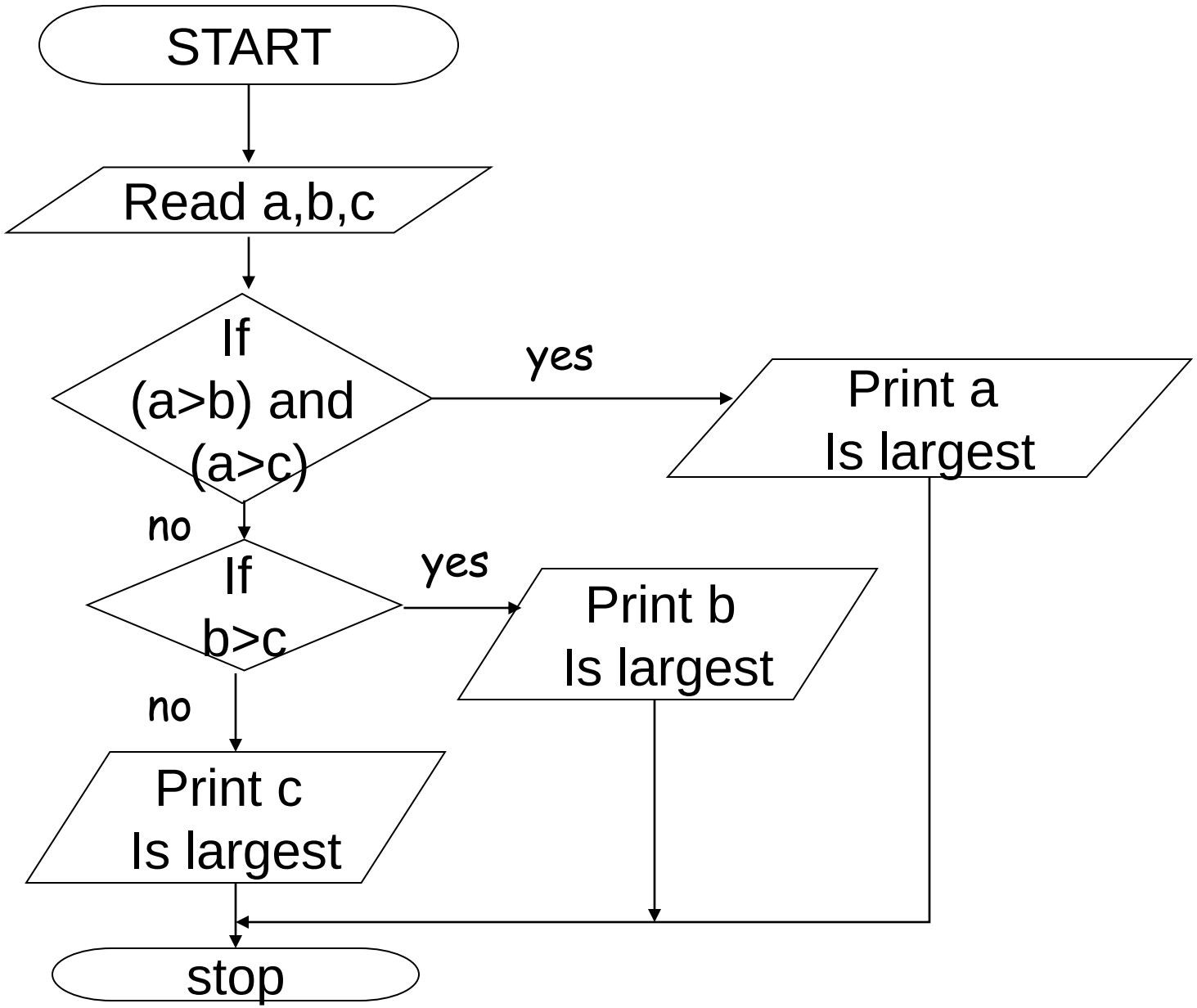
Find the largest among three Numbers

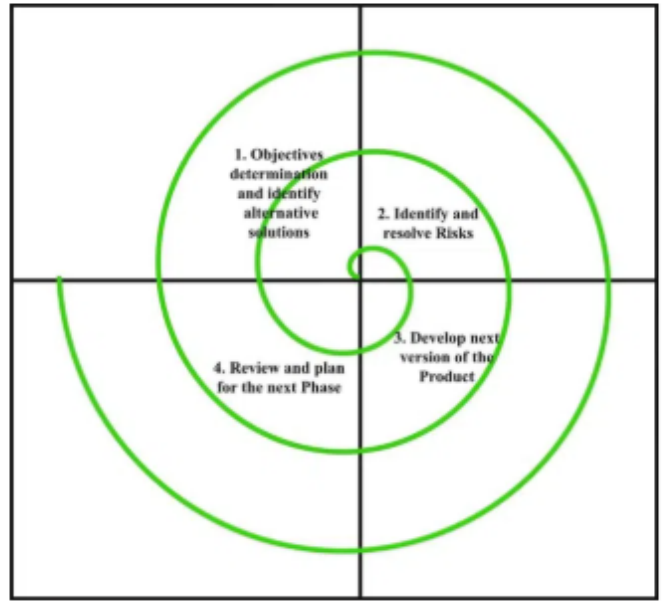
Algorithm
Step1: Start
Step2: Read the value of a, b, c
Step3: IF (a>b) and (a>c) THEN
print a is largest
ELSE IF (b>c) THEN
print b is largest
ELSE
print c is largest
Step4: Stop

Pseudocode

```
READ a, b, c
IF (a>b) and (a>c) THEN
  WRITE a is largest
ELSE IF (b>c) THEN
  WRITE b is largest
ELSE
  WRITE c is largest
ENDIF
stop
```

Flowchart


START
Read a,b,c
If
(a>b) and
(a>c)
yes
Print a
Is largest
no
If
b>c
yes
Print b
Is largest
no
Print c
Is largest
stop



Spiral Model

Each phase of the Spiral Model is divided into four quadrants as shown in the above figure. The functions of these four quadrants are discussed below:

1. **Objectives determination and identify alternative solutions:** Requirements are gathered from the customers and the objectives are identified, elaborated, and analyzed at the start of every phase. Then alternative solutions possible for the phase are proposed in this quadrant.
2. **Identify and resolve Risks:** During the second quadrant, all the possible solutions are evaluated to select the best possible solution. Then the risks associated with that solution are identified and the risks are resolved using the best possible strategy. At the end of this quadrant, the Prototype is built for the best possible solution.
3. **Develop the next version of the Product:** During the third quadrant, the identified features are developed and verified through testing. At the end of the third quadrant, the next version of the software is available.
4. **Review and plan for the next Phase:** In the fourth quadrant, the Customers evaluate the so-far developed version of the software. In the end, planning for the next phase is started.


Portability
Reliability
Reusability
Quality
Coding
Maintainability
Scalability
Testability