

Computer Algorithms (COS 101)

Week 2



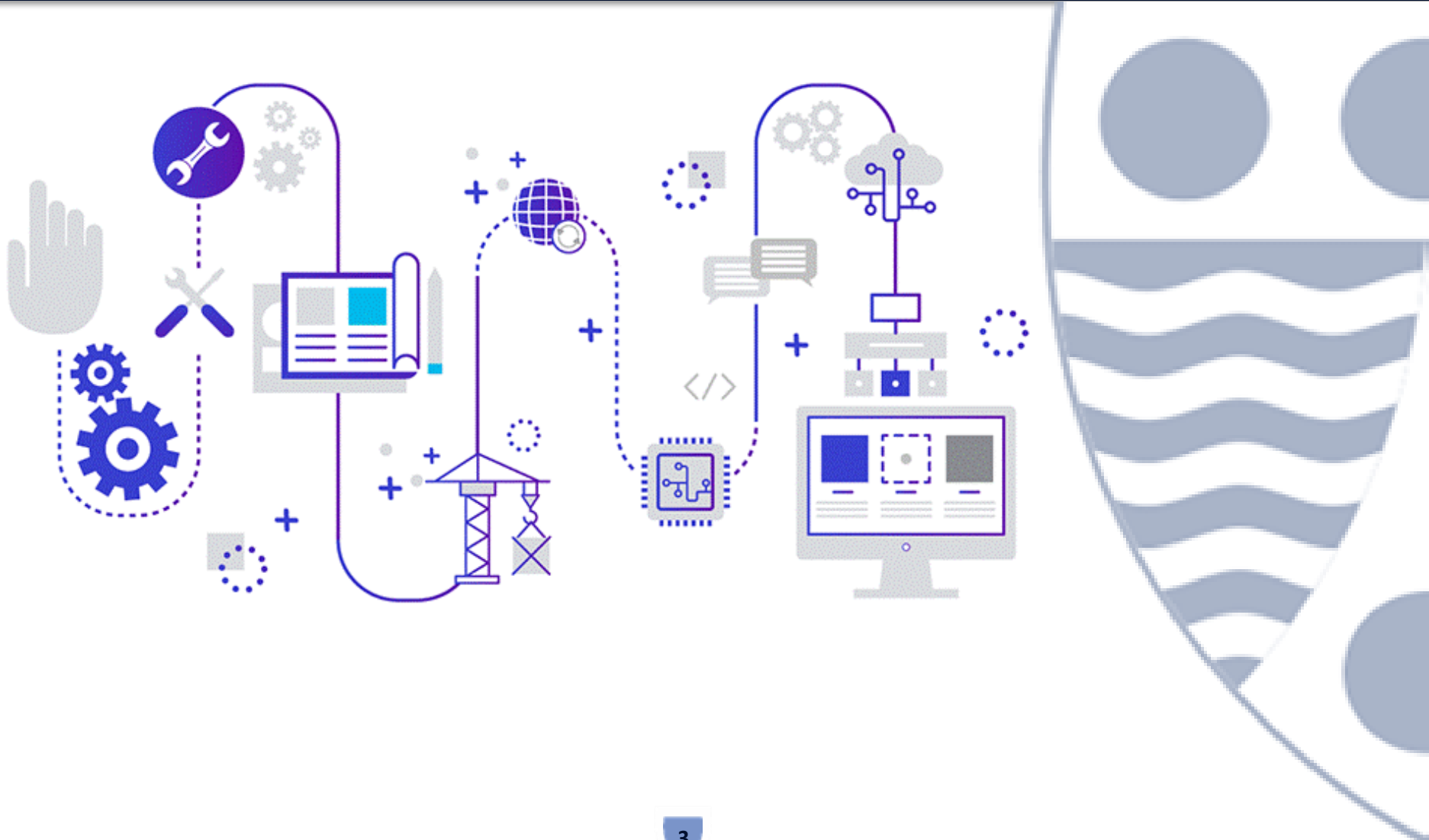
**SCHOOL OF
SCIENCE AND
TECHNOLOGY**

PAN-ATLANTIC UNIVERSITY

Learning Objectives

- What is an Algorithm?
- Why do we need Algorithms?
- Characteristics of a Good Algorithm
- Advantages of Algorithm
- Components of an Algorithm
- Disadvantages of Algorithm
- Benefits of an Algorithm
- Representations of Algorithm

What is an Algorithm?



Other Definitions

1

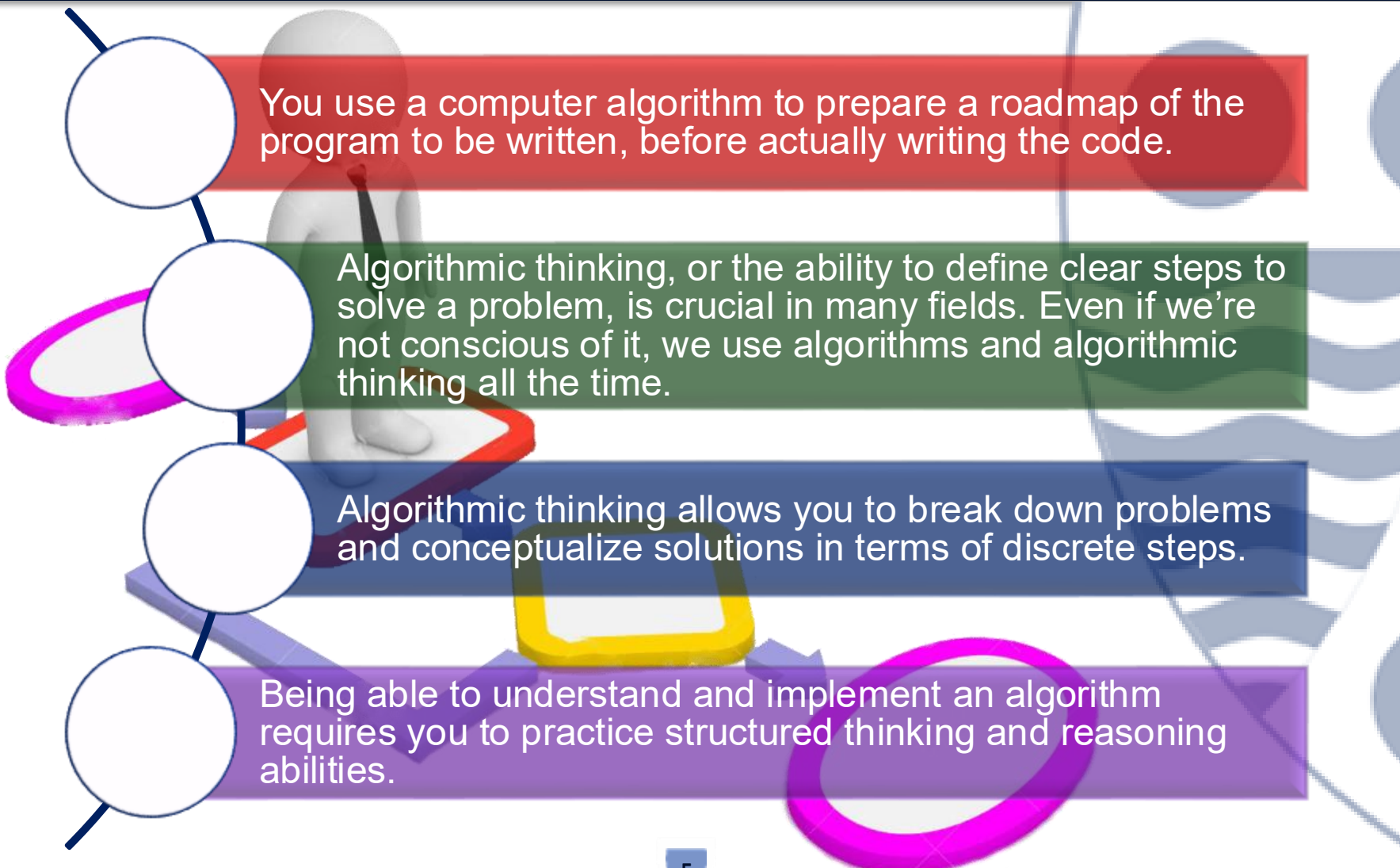
A finite set of unambiguous instructions performed in a prescribed sequence to achieve a goal, especially a mathematical rule or procedure used to compute a desired result.

2

A predetermined set of instructions for solving a specific problem in a limited number of steps.

A precise step-by-step plan for a computational procedure that possibly begins with an input value and yields an output value in a finite number of steps.

Why do we need an Algorithm?



You use a computer algorithm to prepare a roadmap of the program to be written, before actually writing the code.

Algorithmic thinking, or the ability to define clear steps to solve a problem, is crucial in many fields. Even if we're not conscious of it, we use algorithms and algorithmic thinking all the time.

Algorithmic thinking allows you to break down problems and conceptualize solutions in terms of discrete steps.

Being able to understand and implement an algorithm requires you to practice structured thinking and reasoning abilities.

Characteristics of a Good Algorithm

Precision:

- The steps are precisely stated or defined.

Uniqueness:

- Results of each step are uniquely defined and only depend on the input and the result of the preceding steps.

Finiteness:

- The algorithm always stops after a finite number of steps.

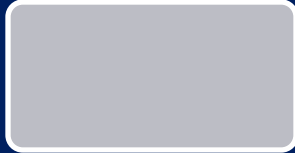
Input:

- The algorithm receives some input.

Output:

- The algorithm produces some output.

Advantages of Algorithm



It is a step wise representation of a solution to a given problem, which makes it easier to understand.



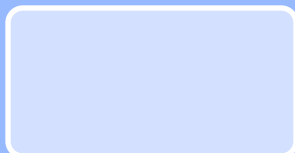
An algorithm uses a definite procedure.



It is not dependent on any programming language, so it is easy to understand for anyone without programming knowledge.

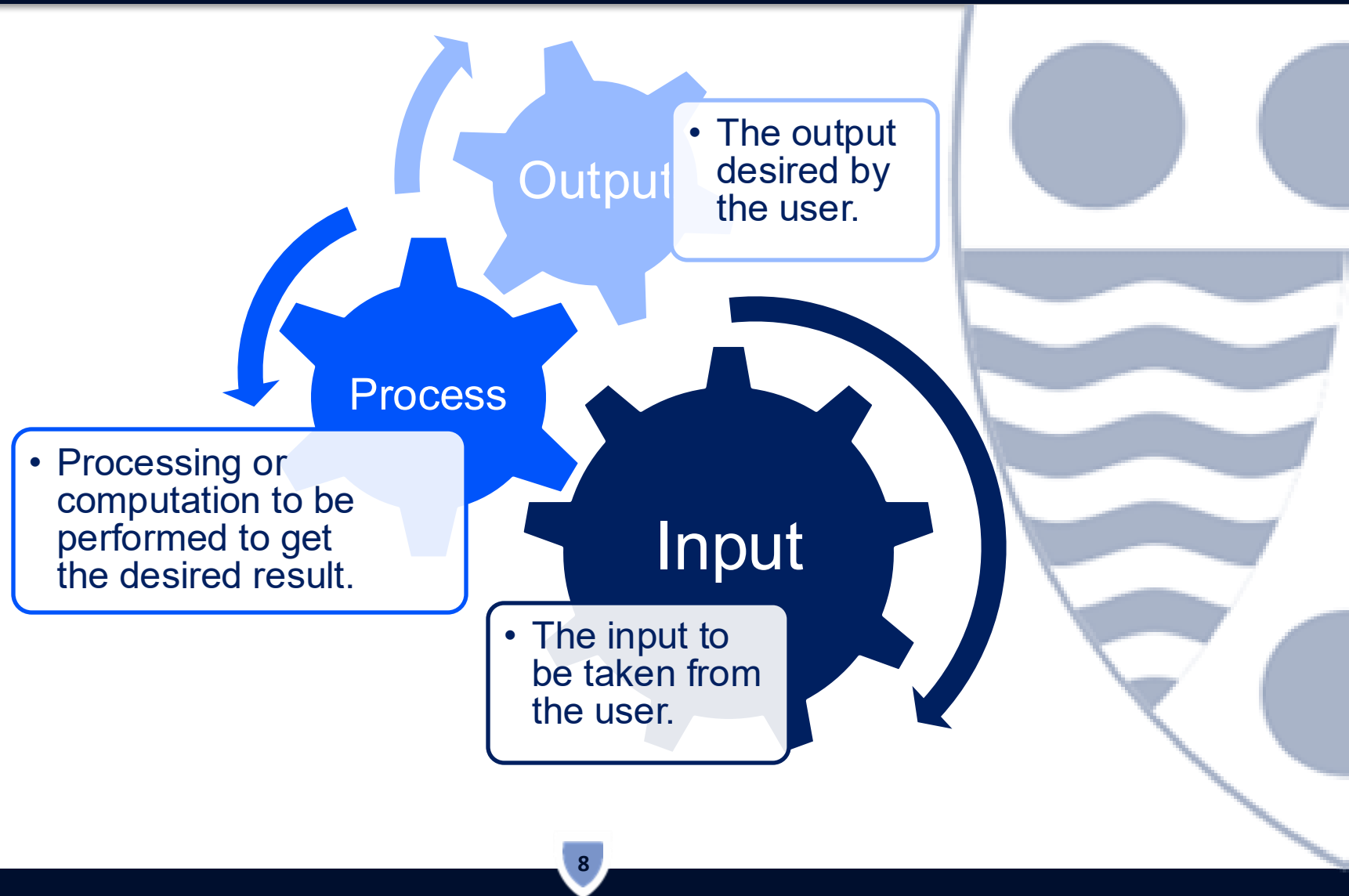


Every step has its own logical sequence, so it is easy to debug



By using algorithm, the problem is broken down into smaller pieces or steps hence, it is easier for programmer to convert it into an actual program.

Components of an Algorithm



Disadvantages of Algorithm

- Algorithms is time consuming.
- Complex problems cannot be solved.
- Small solutions are not clearly shown.
- Requires lots of data.
- Difficult to show branching and looping in algorithms.
- Big tasks are difficult to put in algorithms.
- Low performance.

Benefits of Algorithms



In computers, an algorithm is very important when we want a specific set of instructions for performing a specific task that is definite.

It works well in automated and high-frequency trending systems.

Algorithms make people's lives easier because they save slots of time for the things that are time taking if done manually.

In an algorithm the problem is divided into parts then it becomes easy to understand every level of the process with logic.

Representation of Algorithms

There are two common methods of representing an algorithm:



Flowcharts

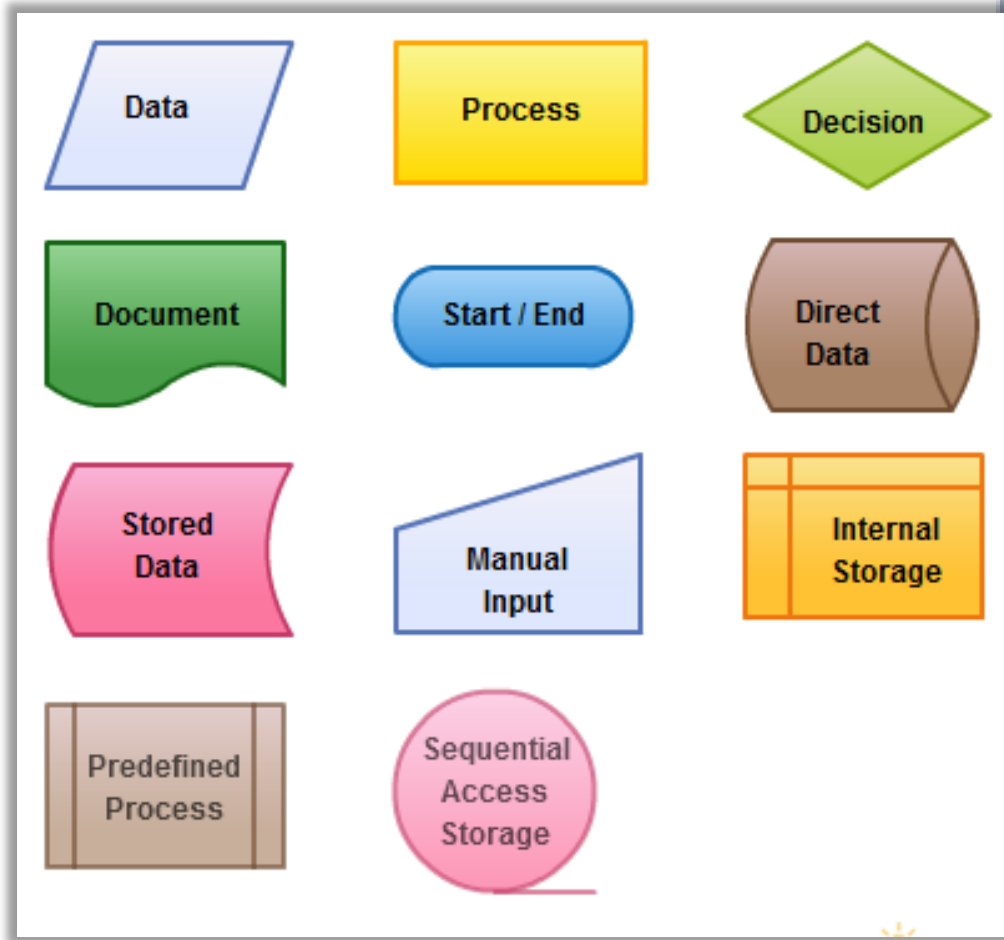


Pseudocodes

Flowchart — Visual Representation of Algorithms

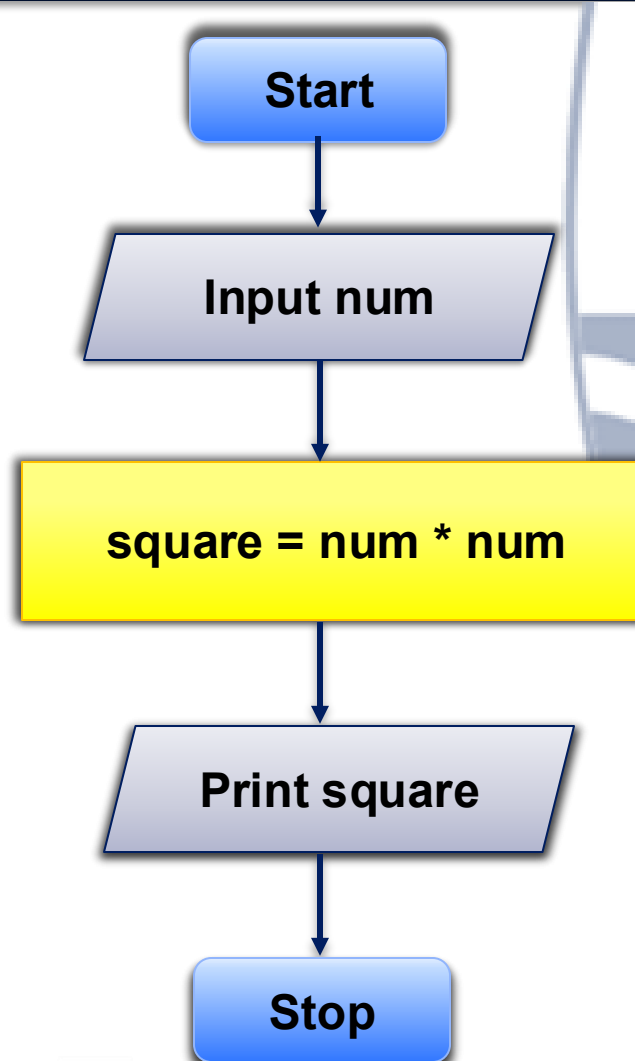
- 🏰 A flowchart is a diagram made up of boxes, diamonds and other shapes, connected by arrows.
- 🏰 Each shape represents a step of the solution process and the arrow represents the order or link among the steps.

Flowchart — Visual Representation of Algorithms

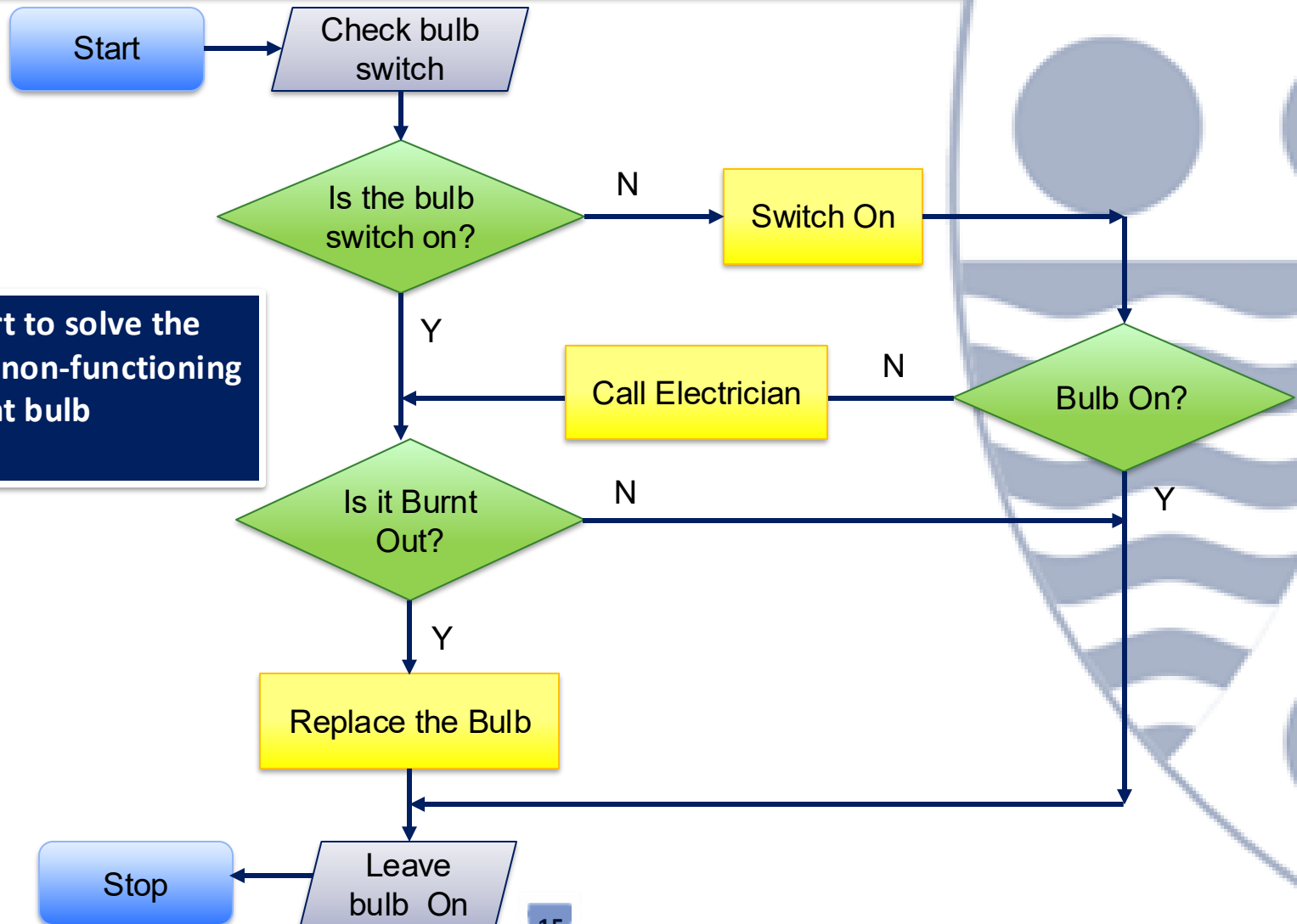


Flowchart — Visual Representation of Algorithms

Algorithm to find square of a number



Flowchart — Visual Representation of Algorithms



Pseudocode

- 🏰 A pseudocode is another way of representing an algorithm.
- 🏰 It is considered as a non-formal language that helps programmers to write algorithm.
- 🏰 It is a detailed description of instructions that a computer must follow in a particular order.
- 🏰 It is intended for human reading and cannot be executed directly by the computer.
- 🏰 No specific standard for writing a pseudocode exists.

Pseudocode Example

Write an algorithm to calculate area and perimeter of a rectangle, using both pseudocode and flowchart.

Pseudocode for calculating area and perimeter of a rectangle

```
INPUT length
INPUT breadth
COMPUTE Area = length * breadth
PRINT Area
COMPUTE Perim = 2 * (length + breadth)
PRINT Perim
```

Pseudocode

Some of the frequently used keywords while writing pseudocode:

INPUT

COMPUTE

PRINT

INCREMENT

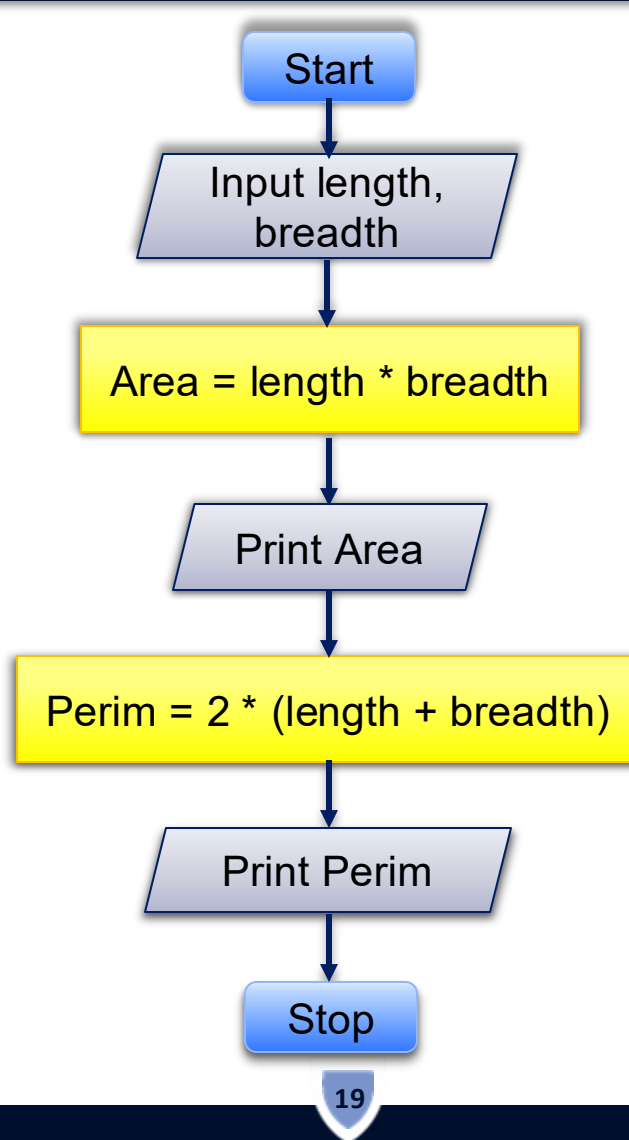
DECREMENT

IF/ELSE

WHILE

TRUE/FALSE

The Flowchart



Benefits of Pseudocode

- ❖ Pseudocode helps in representing the basic functionality of the intended program.
- ❖ By writing the code first in a human readable language, the programmer safeguards against leaving out any important step.
- ❖ For non-programmers, actual programs are difficult to read and understand, but pseudocode helps them to review the steps to confirm that the proposed implementation is going to achieve the desired output.



**SCHOOL OF
SCIENCE AND
TECHNOLOGY**

PAN-ATLANTIC UNIVERSITY