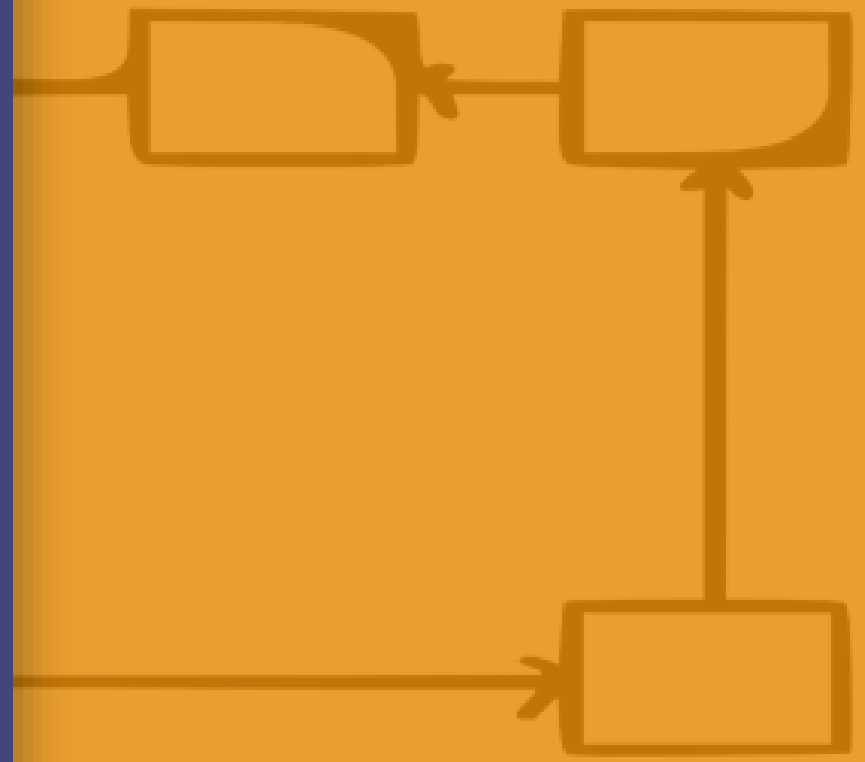
CSC121: INTRODUCTION TO ALGORITHM DESIGN AND DEVELOPMENT

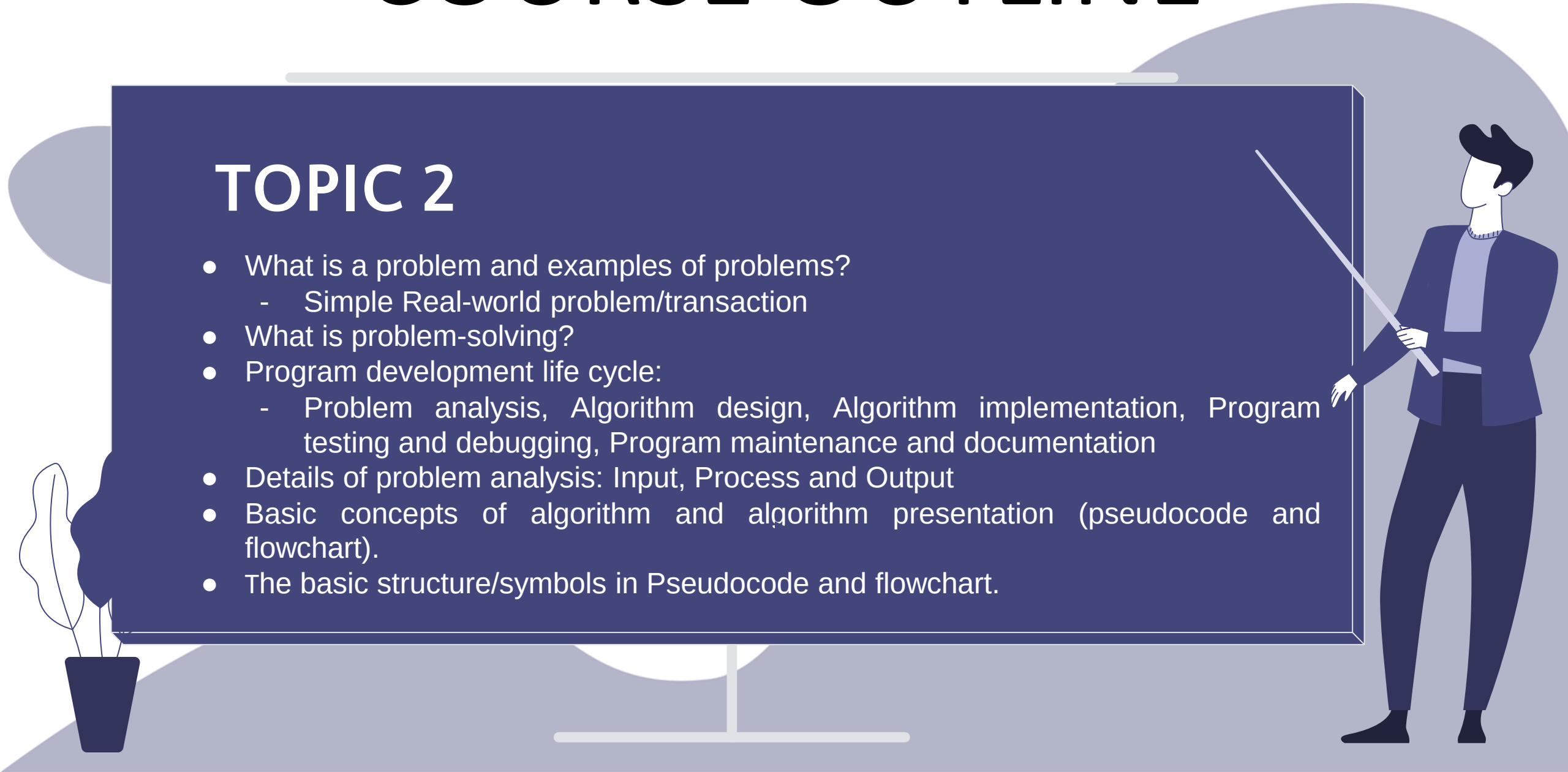
TOPIC 2: INTRODUCTION TO PROBLEM-SOLVING AND ALGORITHM DESIGN



COURSE OUTLINE

TOPIC 2

- What is a problem and examples of problems?
  - Simple Real-world problem/transaction
- What is problem-solving?
- Program development life cycle:
  - Problem analysis, Algorithm design, Algorithm implementation, Program testing and debugging, Program maintenance and documentation
- Details of problem analysis: Input, Process and Output
- Basic concepts of algorithm and algorithm presentation (pseudocode and flowchart).
- The basic structure/symbols in Pseudocode and flowchart.



WHAT IS A PROBLEM AND EXAMPLES OF PROBLEMS?

WHAT IS A PROBLEM?

DEFINITION

A state of difficulty that needs to be resolved or.
A question raised for consideration or solution.
A difficulty: a matter about something difficult to decide on what to do. A question to be answered or solve.

A problem, can be caused for different reasons, and usually can be solved in a number of different ways.

WHAT IS A PROBLEM?

TYPES OF PROBLEM

KNOWLEDGE-LEAN PROBLEMS

- Can be solved (though not always skillfully) by use of instructions for the task and general problem solving skills
- Ex: finding a parking space in the mall, shampooing hair.

KNOWLEDGE-RICH PROBLEMS

- Requires specific knowledge or skill to solve the problem
- Ex: calculus, computer-programming problems

EXAMPLES OF PROBLEMS

SIMPLE PROBLEM

- Make a cup of tea.
- Cook a pot of rice.
- Log in the email account.
- Unlock the front door.
- Switch on a fan or lamp.

COMPLEX PROBLEM

- Traffic light control.
- Public transport schedule.
- Online transaction payment.
- Recommendation in online shopping application.
- Automatic washing machine.

WHAT IS PROBLEM-SOLVING?

DEFINITION

Solution: an action to solve a problem.
Solving a problem: making the problem go away, so that the it does not exist any longer.

Solving a problem of “making a cup of tea”:

1. Put a teabag into a cup.
2. Pour boiled water into the cup.
3. Add sugar and milk into the cup.
4. Stir.

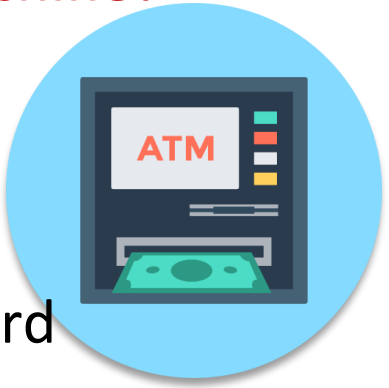
Solving a problem of “log in Google email account”:

1. Go to mail.google.com
2. Enter email ID
3. Enter password
4. Enter or click on login button.

WHAT IS PROBLEM-SOLVING?

#	PROBLEMS	EXAMPLE
1	Transaction	ATM Machine, Web Application
2	Decision Making	Forecasting
3	Control problems	Traffic Controller
4	Searching problems	Search Engines
5	Sorting problems	Transport Schedule

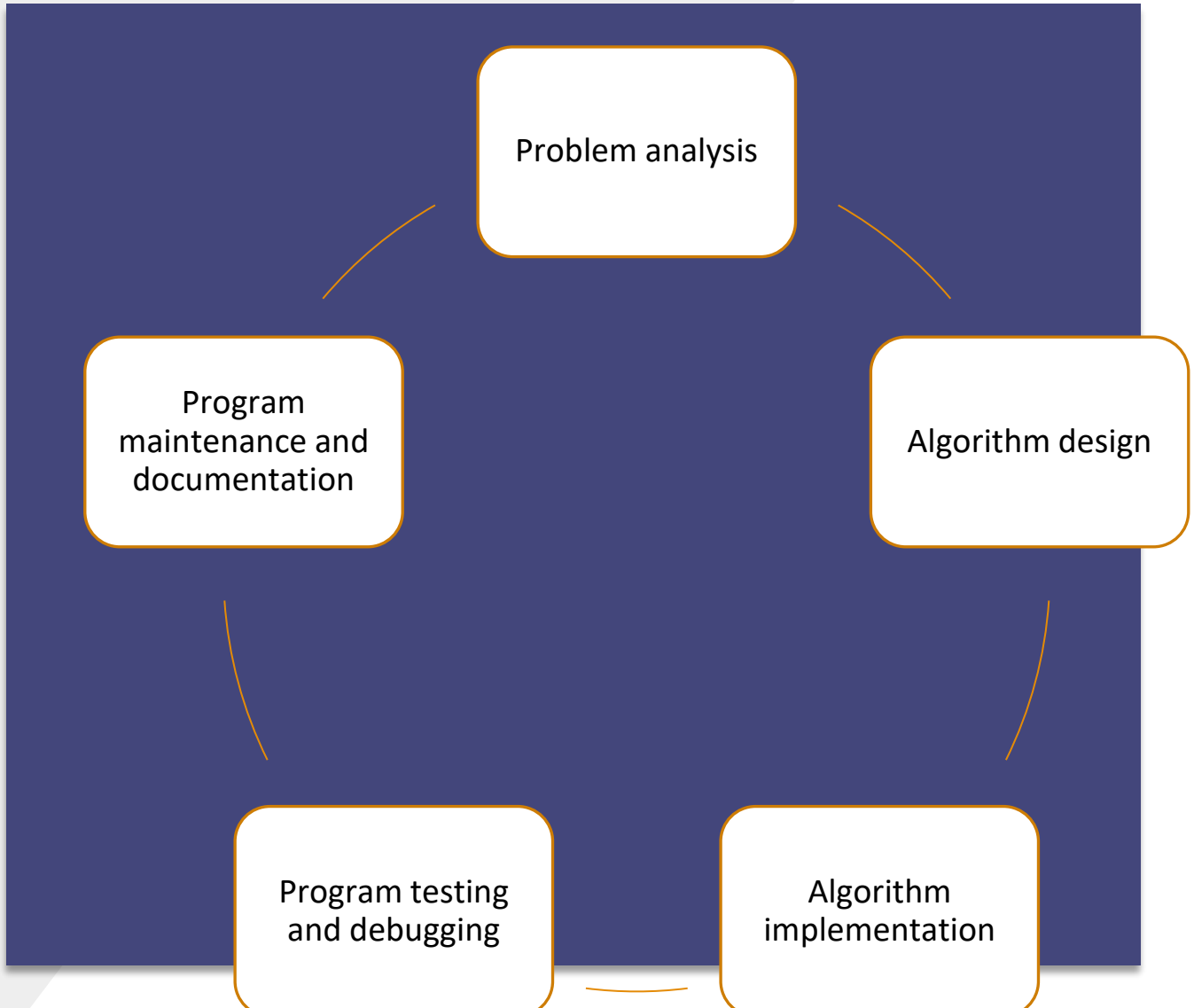
How to draw your money at ATM machine?



1. Insert ATM card in ATM machine
2. Choose a language
3. Enter pin number of your ATM card
4. Choose a withdrawal process
5. Enter the amount of money you want to withdraw
6. Take your ATM card. Wait for your money and transaction slip to come out.

PROGRAM DEVELOPMENT LIFE CYCLE

PROGRAM DEVELOPMENT LIFE CYCLE



PROGRAM DEVELOPMENT LIFE CYCLE

1. **Problem Analysis**

- Understanding the problem.
- Purpose: clearly analyse the problem.
- Try to break up the problem into smaller meaningful workable pieces of information.
- Solution: depends on the outcome from this phase.

Steps of Problem Analysis

1. Determine required information.
2. List all problem facts. Determine how the fact can be used in the solution.
3. Determine assumption to be used. Avoid irrelevant or over assumptions.
4. Determine the data to be used.
5. Determine input data from users.
6. Determined formula to be used.
7. Design the expected screen appearance.

PROGRAM DEVELOPMENT LIFE CYCLE

TERM	DEFINITION
Information	Meaningful interpretation of data
Data	Raw material used to get the information.
Formula	An expression that tells the computer what mathematical operation to perform upon a specific value.
Expected screen appearance	Communication channel between the program built and the real user. It is used to allow users and computer program to communicate to each other. Ex: response messages, the design of user interface.

PROGRAM DEVELOPMENT LIFE CYCLE

2. **Algorithm Design**

- Develop and carry out the problem solving plan.
- Focus on logical solution of the problem.
- Developing an algorithm. An algorithm is the sequence of steps or rules you follow to solve a problem.
- Shouldn’t worry about the syntax of any particular language, but focus on figuring out what sequence of events will lead from the available input to the desired output.
- Think carefully about all the possible data values a program might encounter and how the program will handle each scenario.

PROGRAM DEVELOPMENT LIFE CYCLE

Steps of Algorithm Design

1. Get information from phase 1.
2. Sketch the plan of problem solving.
3. Integrate the logical solution algorithm in the form of pseudocode or flowchart.
4. If the problem is too complex, break the problem into sub-problems.
5. Find solutions for each of the sub-problem.
6. Combine the solution for the entire problem.

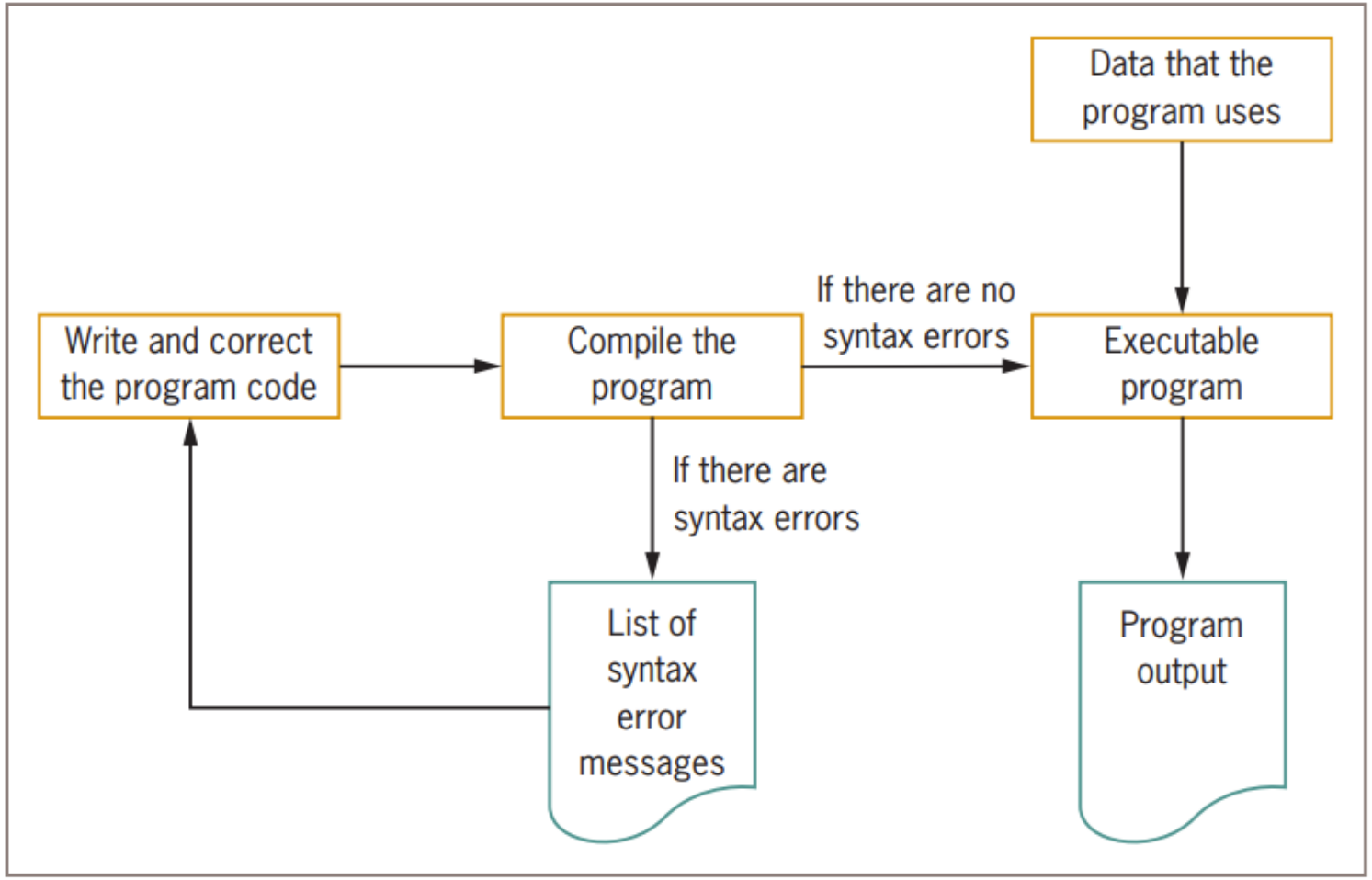
PROGRAM DEVELOPMENT LIFE CYCLE

3. **Algorithm implementation**

- Choose particular languages. Some have built-in capabilities that make them more efficient than others at handling certain types of operations.
- Programming language can handle input operations, arithmetic processing, output operations, and other standard functions.
- After choosing a language, the programmer prepares with proper punctuation and the correct syntax (spelling of commands).

PROGRAM DEVELOPMENT LIFE CYCLE

Steps of Algorithm Implementation



PROGRAM DEVELOPMENT LIFE CYCLE

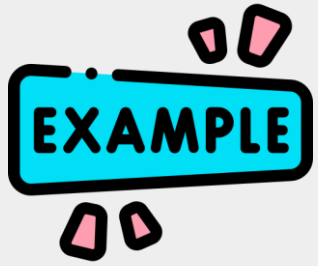
4. **Program testing and debugging**

- Program testing is the process of executing a program with the intent of finding errors.
- A good test is one that has a high probability of finding an error.
- A program that is free of syntax errors is not necessarily free of logical errors.
- Once a program is free of syntax errors, the programmer can test it—that is, execute it with some sample data to see whether the results are logically correct.

PROGRAM DEVELOPMENT LIFE CYCLE

Steps of Testing and Debugging

1. Check for syntax error.
2. Execute the program with some sample data, to validate the logic of the program.

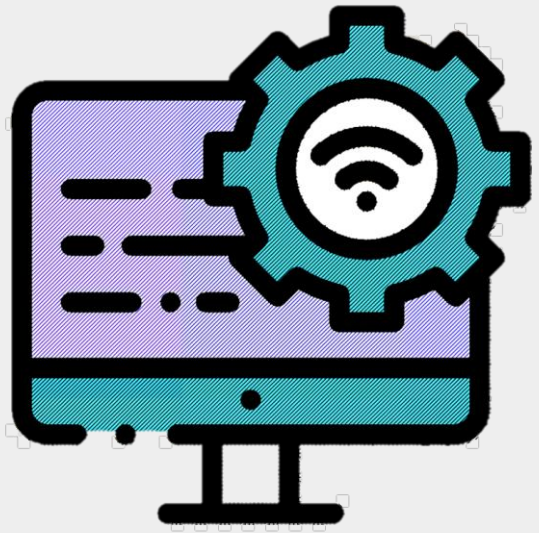



```
input myNumber
set myAnswer = myNumber * 2
output myAnswer
```

If you execute the program, provide the value 2 as input to the program, and the answer 4 is displayed, you have executed one successful test run of the program. However, if the answer 40 is displayed, maybe the program contains a logical error.

PROGRAM DEVELOPMENT LIFE CYCLE

Steps of Testing and Debugging



4. After the program is thoroughly tested and debugged, put the program into production.
   i. Data-entry training.
   ii. End-user training.
   iii. Data migration and conversion.

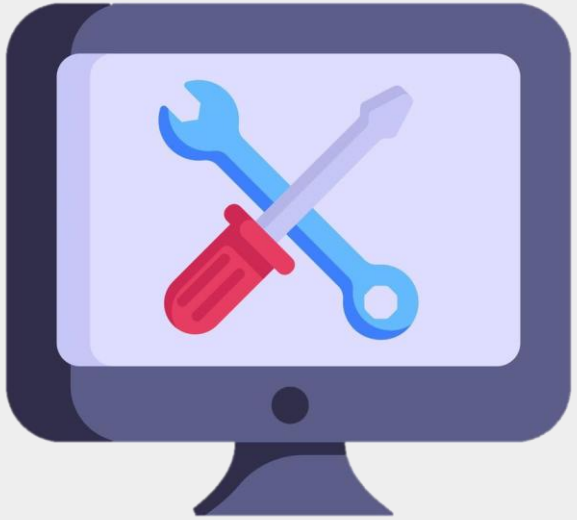
PROGRAM DEVELOPMENT LIFE CYCLE

5. **Program maintenance & documentation**

- Maintenance: make necessary changes on the completed program.
- Why? To sustain the capability of a program to provide a service.
- For example, new tax rates are legislated, the format of an input file is altered, or the end user requires additional information not included in the original output specifications.

PROGRAM DEVELOPMENT LIFE CYCLE

Steps of Testing and Debugging



1. When existing programs are changed, the development cycle is repeated.
2. These elements must be understood:
   - the changes
   - then plan
   - code
3. Then translate and test the program before putting them into production.
4. If a substantial number of program changes are required, the original program might be retired, and the program development cycle might be started for a new program.

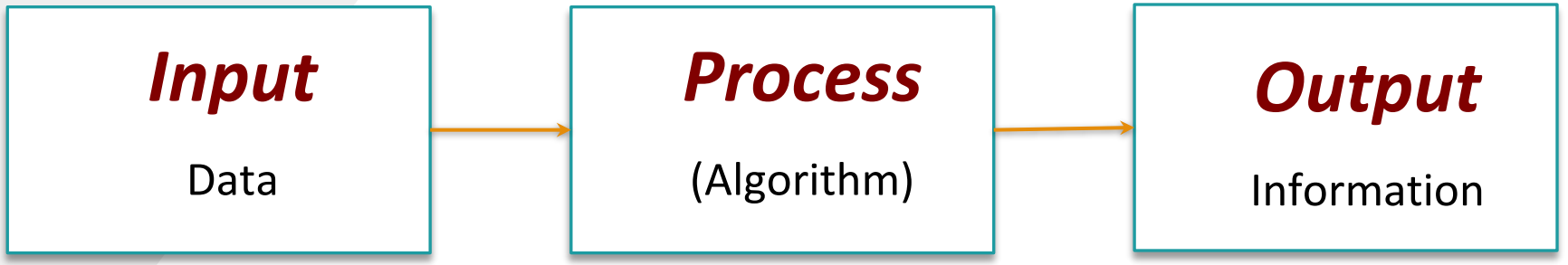
DETAILS OF PROBLEM ANALYSIS: INPUT, PROCESS AND OUTPUT

PROBLEM ANALYSIS

DEFINITION

During problem analysis, you should specify requirement as below:

- Input
- Processing
- Output



INPUT

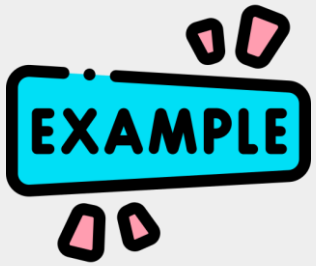
DEFINITION

- Data that is entered into or received by a computer.
- It is sent to a program for processing.

PROCESS

DEFINITION

- A process or running process refers to a set of instructions currently being processed by the computer processor.
- A process runs in a computer. This can be anything from a small background task, such as a spell-checker or system events handler to a full-blown application like internet browser or word processor applications.




```
input myNumber
set myAnswer = myNumber * 2
output myAnswer
```

Receive input is a process

Execute multiplication is a process

Display the answer is a process

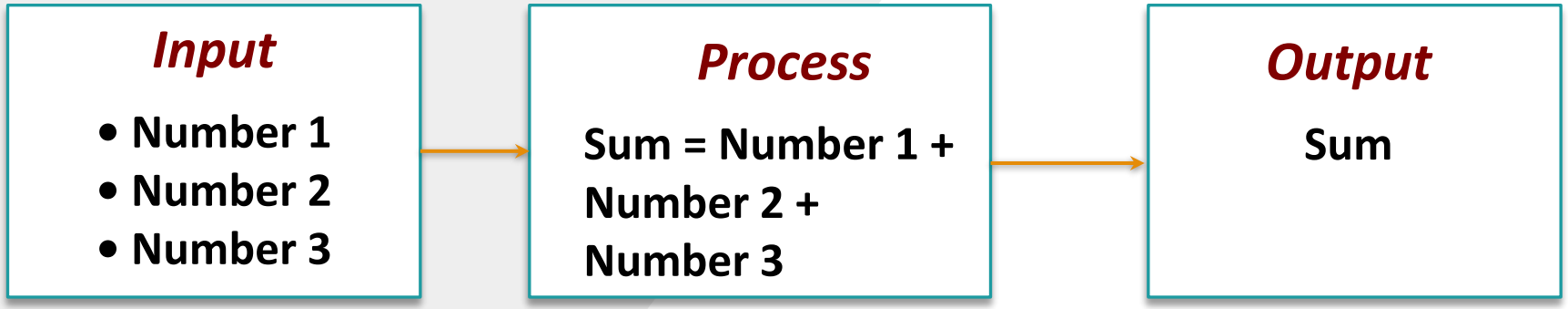
OUTPUT

DEFINITION

- The output of a computer or word processor is the information that it displays on a screen or prints on paper as a result of a particular program.
- There are four basic types of output: audio output, graphics output, text output, and video output.

PROBLEM ANALYSIS: EXAMPLE

EXAMPLE # 1: Compute the sum of 3 numbers

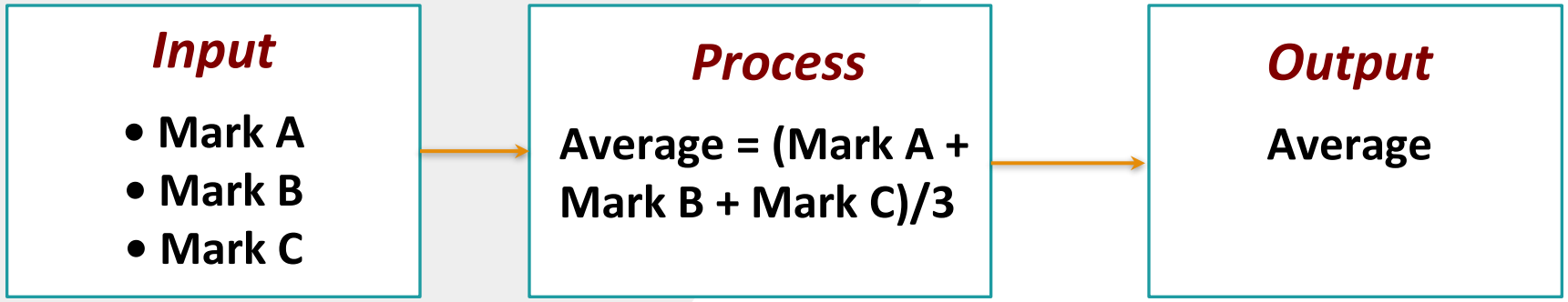


Algorithm:

Input Number1, Number2, Number3
Calculate Sum by
    Sum = Number 1 + Number 2 + Number 3
Display Sum

PROBLEM ANALYSIS: EXAMPLE

EXAMPLE # 2: Calculate and display the average mark of three students



Algorithm:

Input Mark A, Mark B, Mark C
Calculate Average by
    Adding the numbers and
    Sum = Mark A + Mark B + Mark C
    Dividing the sum by 3
    Average = Sum / 3
Display Average

BASIC CONCEPTS OF ALGORITHM AND ALGORITHM PRESENTATION (PSEUDOCODE)

PSEUDOCODE

DEFINITION

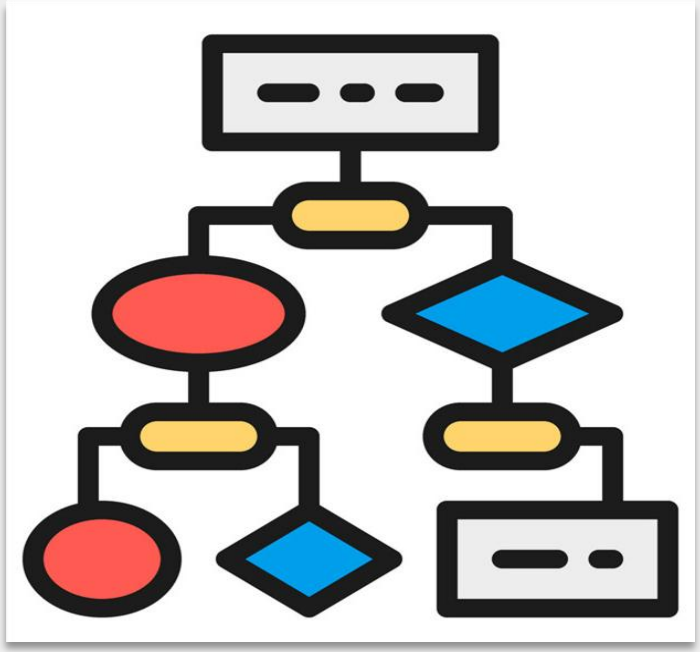
```
IF spaceship sprite touches asteroid sprite THEN
   show explosion sprite
   play explosion sound
   subtract a life
END IF

IF lives = 0 THEN
   stop game
   show game over screen
ELSE
   restart game
ENDIF
```

- A step by step problem solving procedure.
- Pseudocode is a tool programmers use to help them plan an algorithm. It is used English like phrases to described the processing process. It is not standardized since every programmer has his own way of planning the algorithm. Criteria of a good pseudocode are:
  i. Easy to understand, precise and clear.
  ii. Gives the correct solution in all cases.
  iii. Eventually ends.

PSEUDOCOCE - ALGORITHM

Characteristic of Algorithm



1. INPUT AND OUTPUT
   - Algorithm receives input and produces the output
2. UNAMBIGUOUS
   - Steps in algorithm must be clear as to what it is supposed to do and how many times it is expected to be executed.
3. CORRECT AND EFFICIENT
   - An algorithm should produce the correct and efficient output values for different set of input values.
4. FINITE
   - It must execute its instruction and terminate in a finite time.

PSEUDOCOCE - ALGORITHM

Algorithm Solution vs.
Heuristic Solution

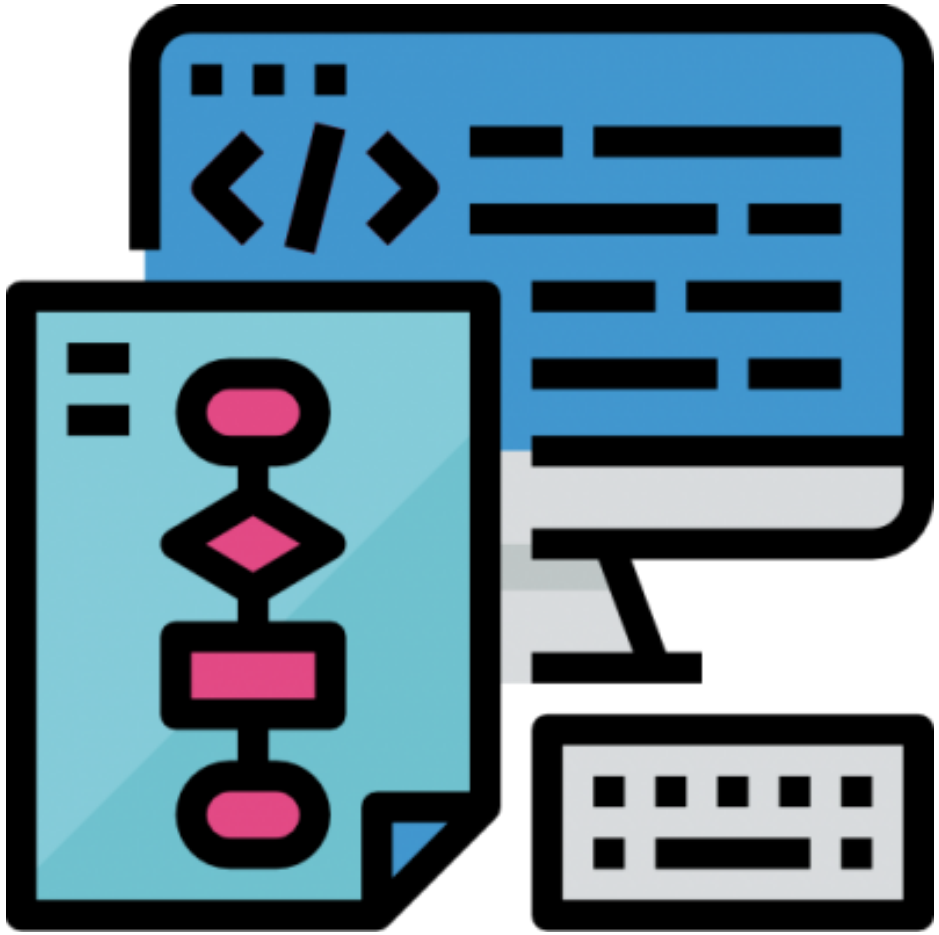
ALGORITHM SOLUTION

A series of actions / steps (algorithm) to solve a problem.

HEURISTIC SOLUTION

A solution that cannot be reached through a direct of steps and require reasonable built on knowledge and experience. (AI)

THE BASIC STRUCTURE / SYMBOLS IN PSEUDOCODE & FLOWCHART



FLOWCHART

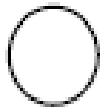
DEFINITION

- Flowchart graphically shows the logic in solution algorithm that produce in early 1960s.
- It shows the step by step solution using symbols which represent a task.
- The symbols used consist of geometrical shapes that are connected by flow lines.

FLOWCHART

Basic Symbol	
	**TERMINAL (BEGIN / END)** Indicates the beginning and end points of an algorithm.
	**PROCESS** Instructions that transform input into output
	**INPUT / OUTPUT** Input or output operation

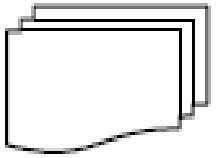
FLOWCHART

	**SELECTION / DECISION** Selection process or condition that determined a specified path to follow
	**CONNECTOR** Entry from or exit to another part of the flowchart on the same page.
	**FLOW OF ACTIVITIES** Indicate the logical sequence of execution steps in the algorithm

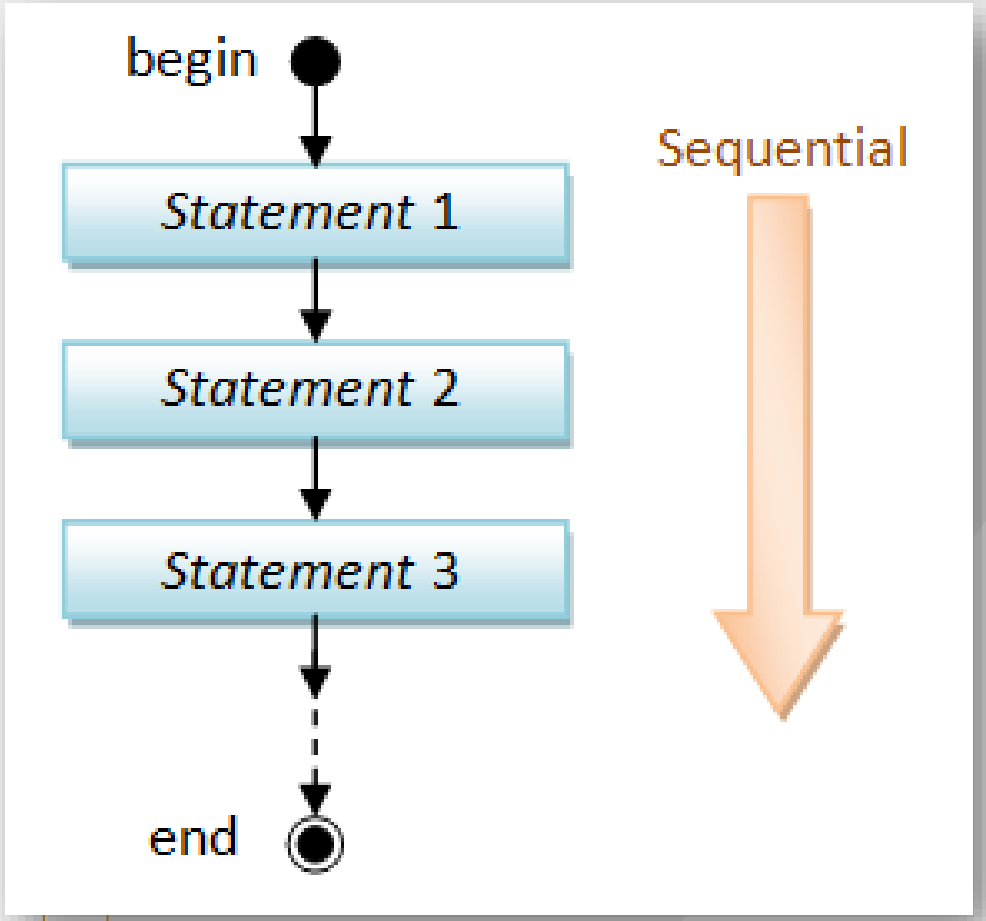
FLOWCHART

Other Symbol	
	**FUNCTION CALL** Process containing a series of program steps specified elsewhere
	**STORAGE I/O** Input from or output to disk storage.
	**CONNECTOR** Entry from or exit to another part of the flowchart on a different page
	**STORED DATA**

FLOWCHART

	DOCUMENT/FILE
	MULTIPLE DOCUMENT/FILE

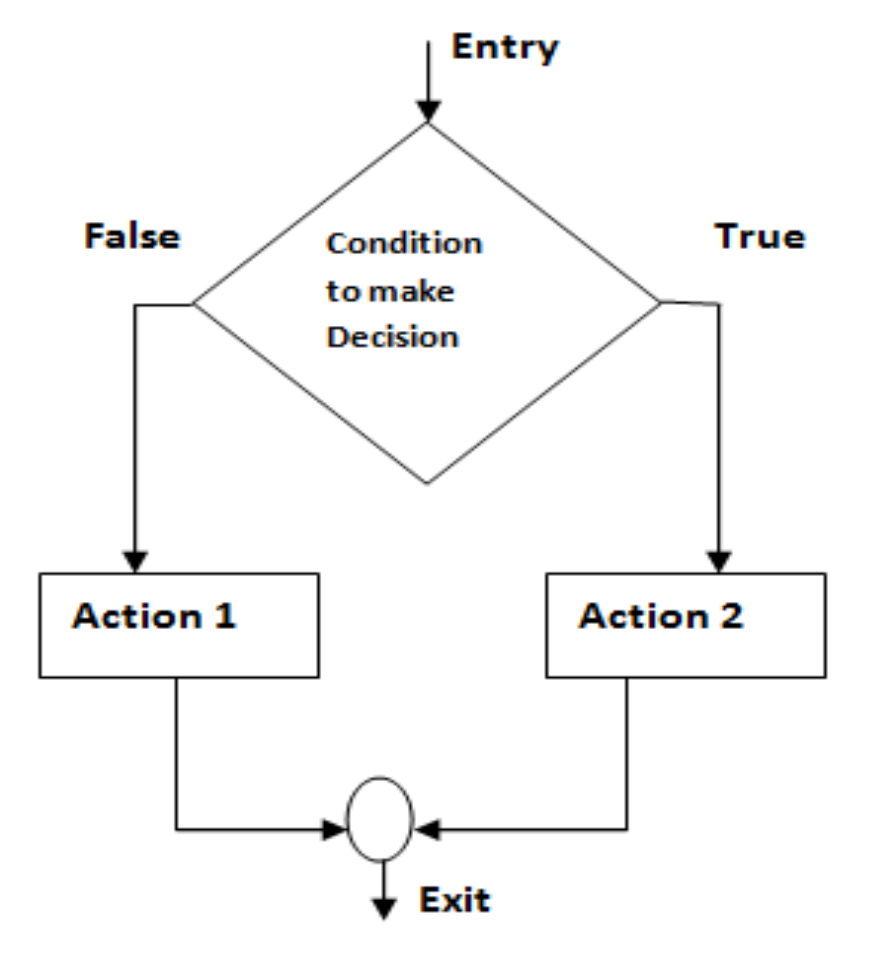
FLOWCHART



SEQUENTIAL STRUCTURE

- The sequential structure has one entry point and one exit point.
- No choices are made and no repetition.
- Statements are executed in sequence, one after another without leaving out any single statement.

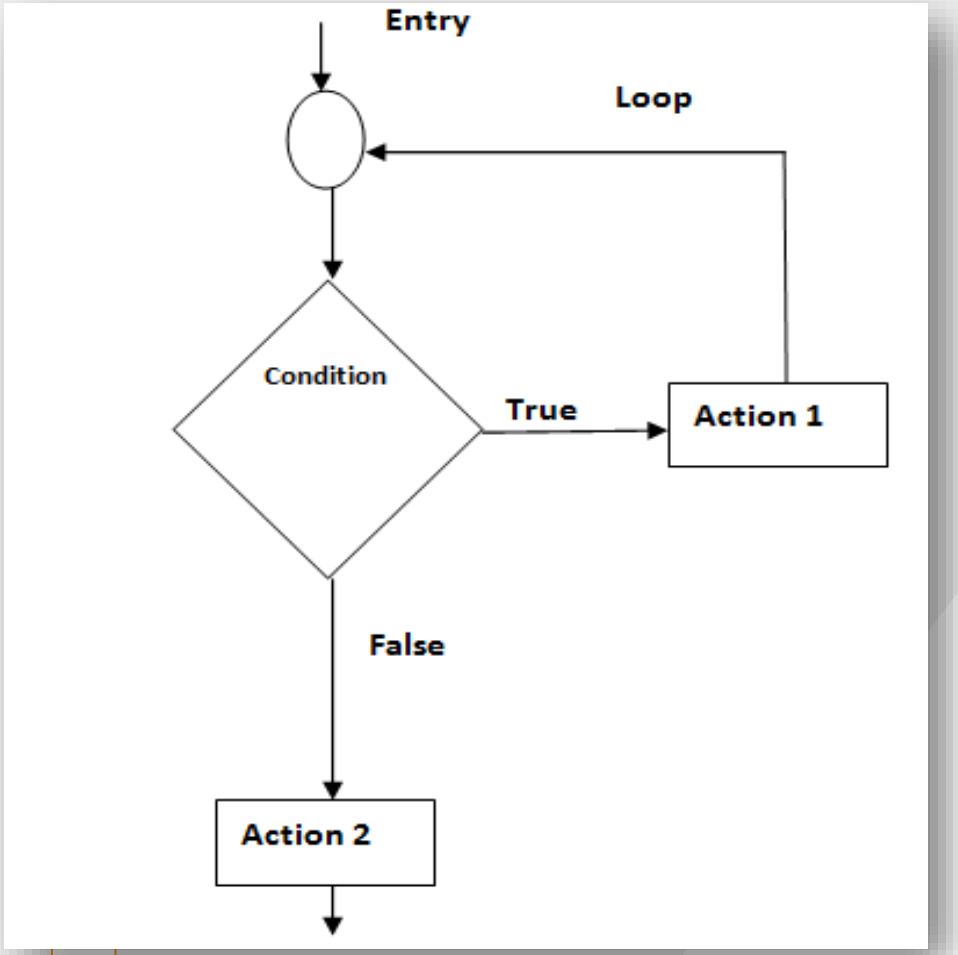
FLOWCHART



SELECTION STRUCTURE

- The selection structure is used to allow choices to be made.
- The program executes particular statements depending on some condition(s).
- C++ uses the if-else and switch statement for making decision.

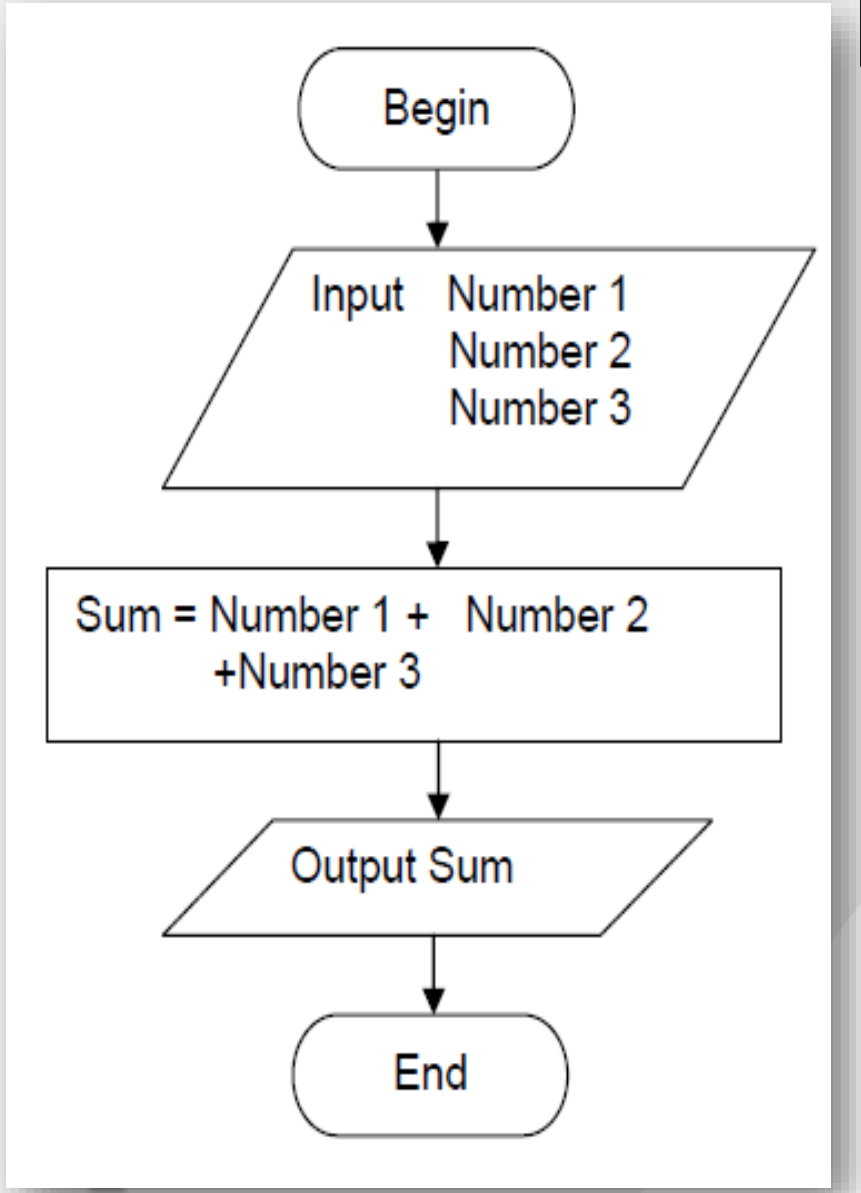
FLOWCHART



REPETITION / LOOP STRUCTURE

- Statements are executed repeatedly while certain condition remains true.
- In C++, while, do-while and for are the statements commonly used within the repetition structure.

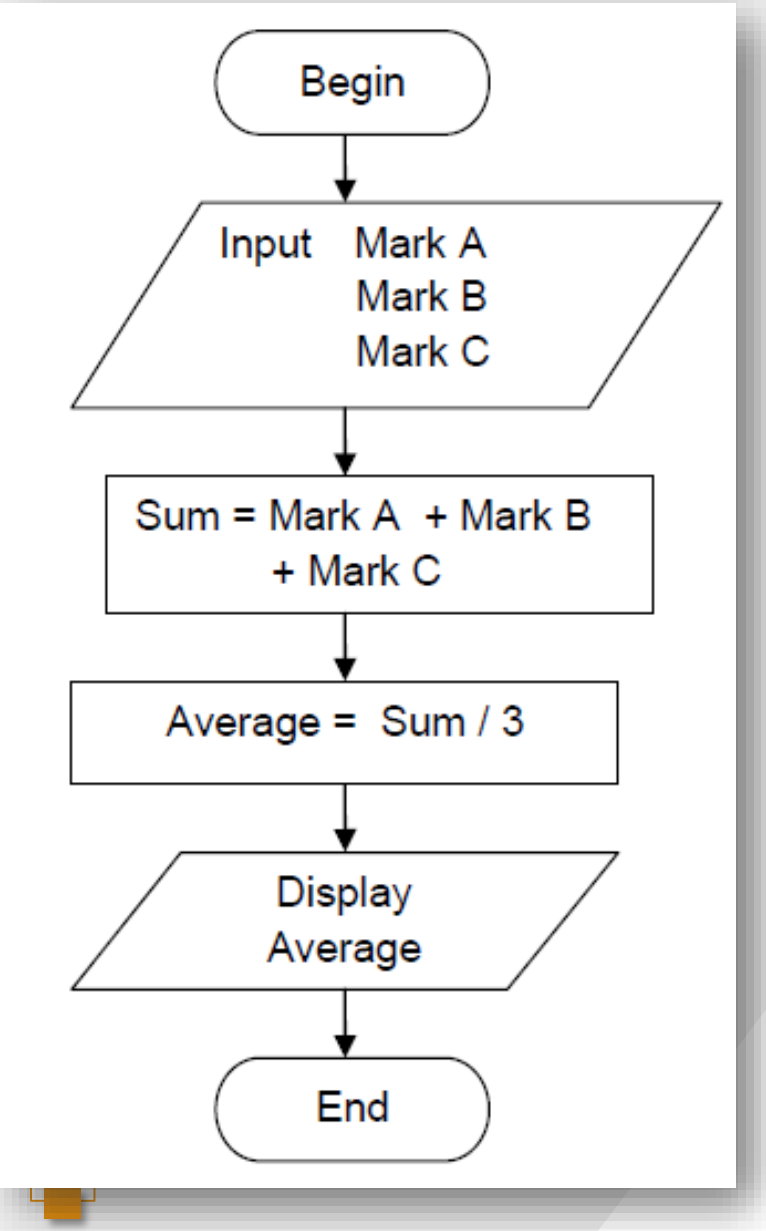
FLOWCHART



EXAMPLE # 1: Compute the sum of 3 numbers

Input	*Process*	*Output*
Number 1 **Number 2** **Number 3**	**Sum = Number 1 + Number 2 + Number 3**	**Sum**

FLOWCHART

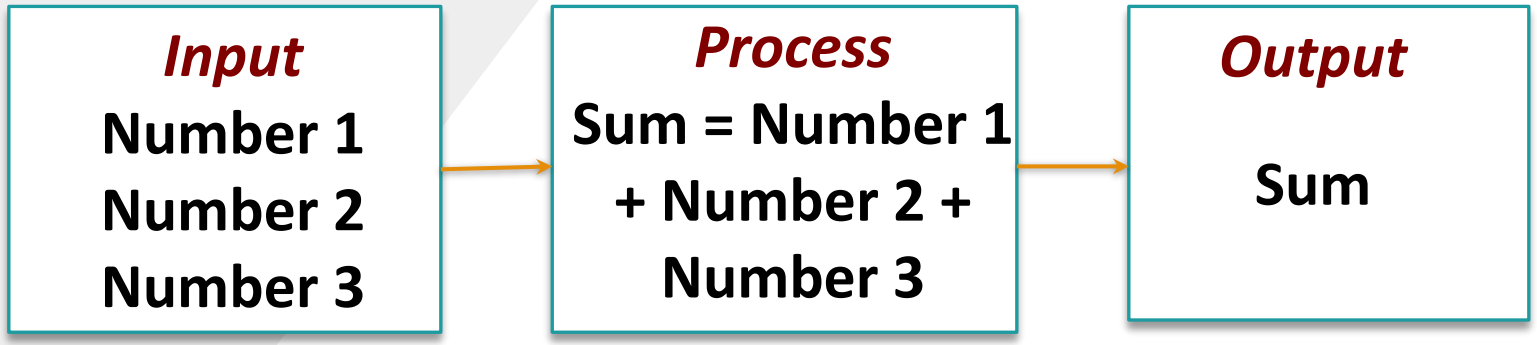


EXAMPLE # 2: Calculate and display the average mark of three students

Input	*Process*	*Output*
Mark A Mark B Mark C	Average = (Mark A + Mark B + Mark C)/3	Average

ALGORITHM

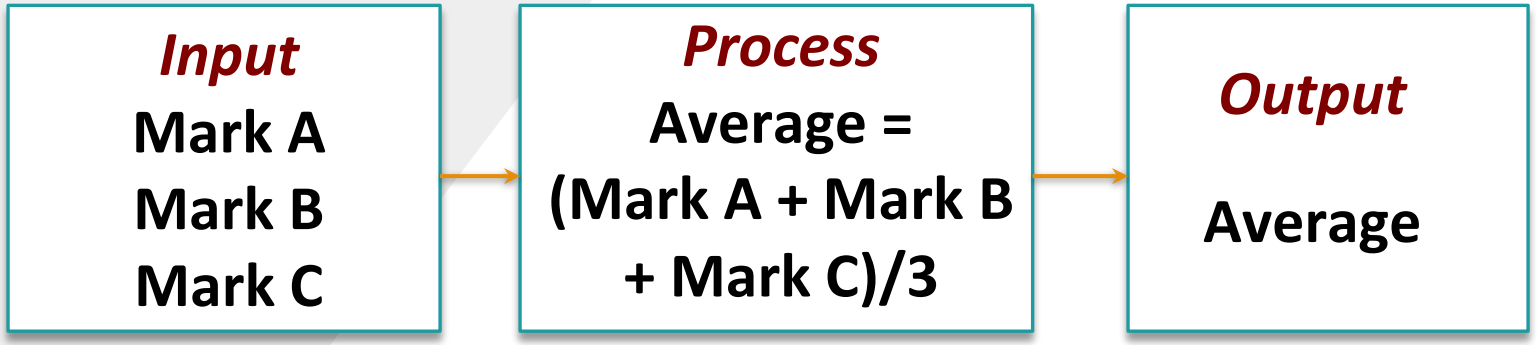
EXAMPLE # 1:
Compute the sum of 3 numbers




```
Begin
        Input Number1, Number2, Number3
        Sum = Number 1 + Number 2 +
Number 3
        Display Sum
End
```

FLOWCHART

EXAMPLE # 2: Calculate and display the average mark of three students




```
Begin
      Input Mark A, Mark B, Mark C
      Sum = Mark A + Mark B + Mark
C
      Average = Sum / 3
      Display Average
End
```

EXERCISE # 1

1. Draw a flowchart to calculate and display the price of a number of apples if the quantity n kg and price per kg are given.
2. Draw a flowchart to compute the area of a square
3. Display the status of a student based on the mark below

Mark	Status
>= 75	Excellent
< 75 and > 50	Moderate
< 50	Fail

EXERCISE # 2

1. Write a pseudocode to calculate and display the price of a number of apples if the quantity *n* kg and price per kg are given.
2. Write a pseudocode to compute the area of a square.

END OF TOPIC

THANK YOU!