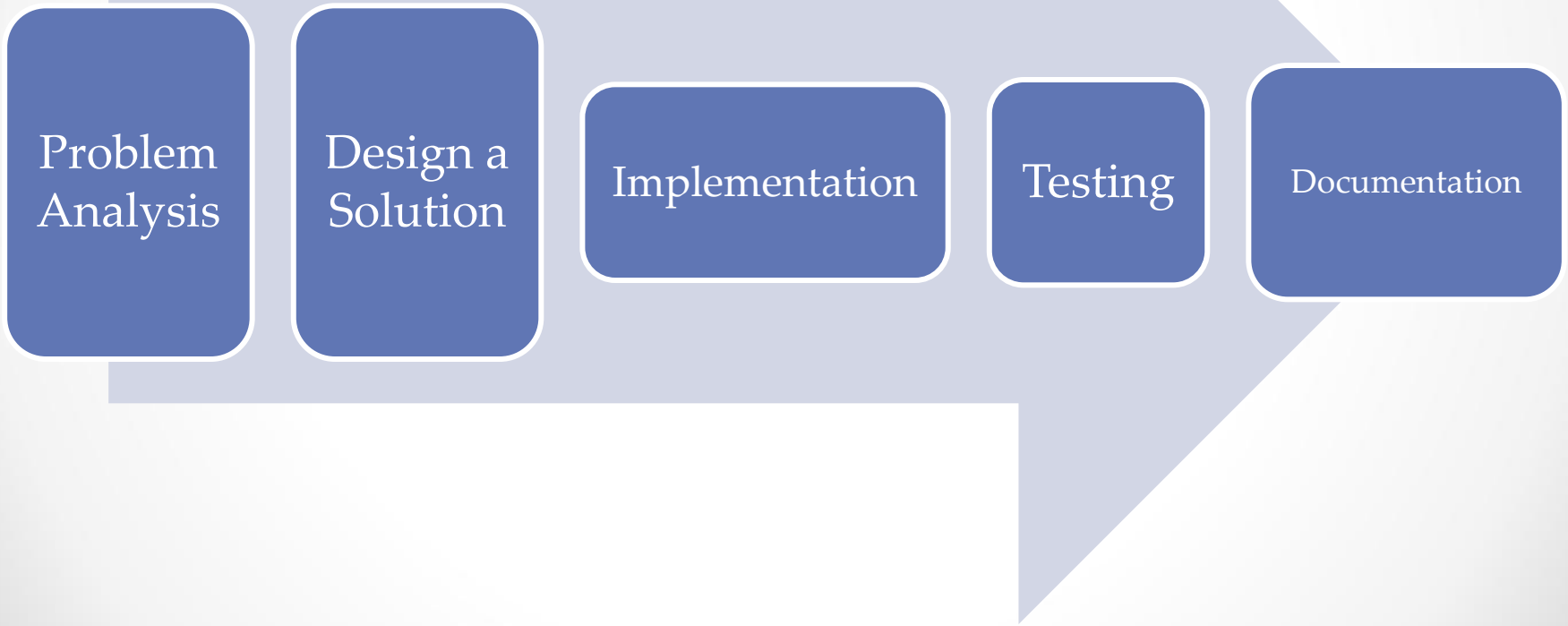
Problem Solving

Learning Outcome

- List five (5) steps in problem solving 8.2.1
- Identify input, process and output from a given problem. 8.2.2
- Define algorithm. 8.2.3.1
- Solve a given problem using algorithm. 8.2.3.1
- Explain the purpose of each control structure. 8.2.3.2
- Apply appropriate control structure in problem solving. 8.2.3.2

Steps in Problem Solving

- There are five (5) steps in problem solving.



Steps in Problem Solving

Problem Analysis

- Be sure you understand what the program should do,
- What the output should be.
- Have a clear idea of what data (or input) are given and the
- Relationship between the input and the desired output.

→ IPO

Design a Solution

- Find a logical sequence of precise steps that solve the problem
- Such a sequence of steps is called an algorithm
- Every detail, including obvious steps, should appear in the algorithm

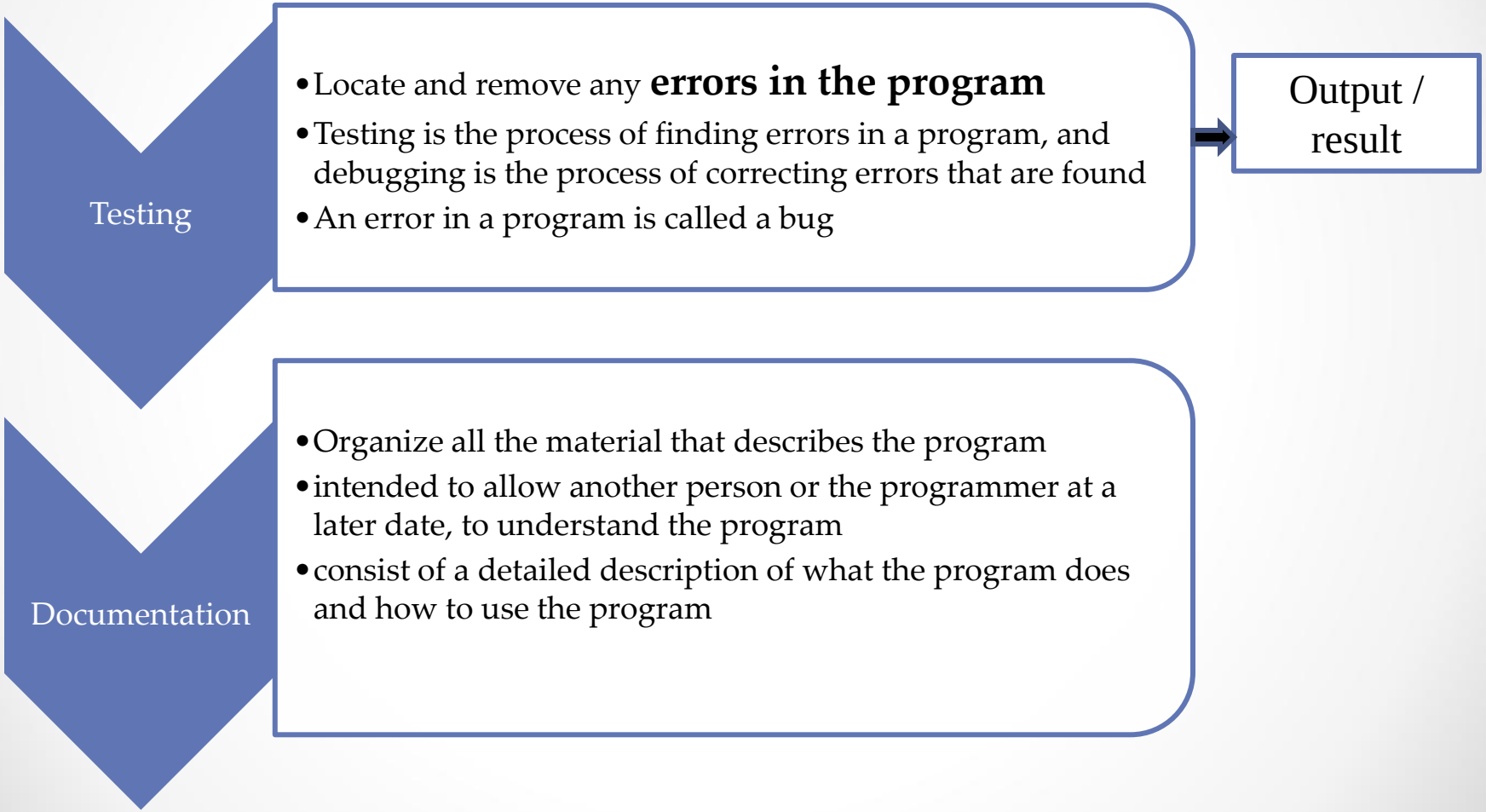
→ Algorithm
- Pseudocode
- Flow chart

Implement

- Translate the algorithm into a programming language
- Coding is the technical word for writing the program.

→ C++ Coding

Steps in Problem Solving



Error in the program

- **Syntax error :**

  Program error that occurs when the code violates the syntax or grammar of the programming language.

  e.g : misspelling a command, leaving out require command.

- **Logic error :**

  Flow in program design that causes inaccurate results.

  e.g : sum = no1- no2

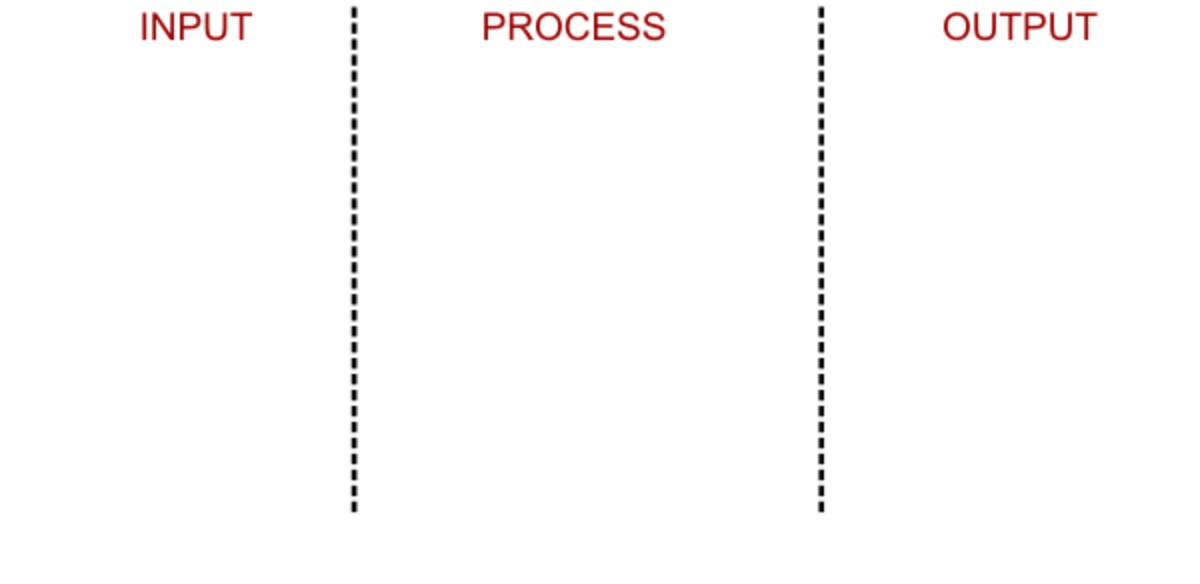
- **Runtime error :**

  Program error or event that causes the program to stop running.

  e.g : error when using the internet

8.2.2 Problem Analysis

- Input-Process-Output (IPO) Analysis
  - to analyze problems and develop algorithms
  - to organize and summarize the results of a problem analysis
  - to shows where in the solution the processing takes place
  - It can also be represent using IPO chart/table.



8.2.2 Problem Analysis with

Input-Process-Output (IPO) Chart

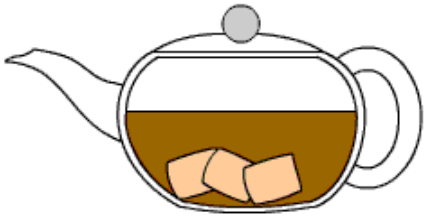
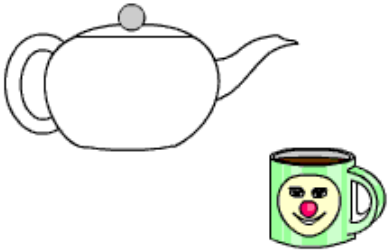
Input	Processing	Output
Read Data An item – an input/data that is needed by computer.	**Perform Computation** an intermediate value that the algorithm uses when processing the input into the output	**Display Results** Print result/s on screen, on paper etc.

8.2.2 Problem Analysis with

Input-Process-Output (IPO) Chart

Examples 1 How to make a tea drink

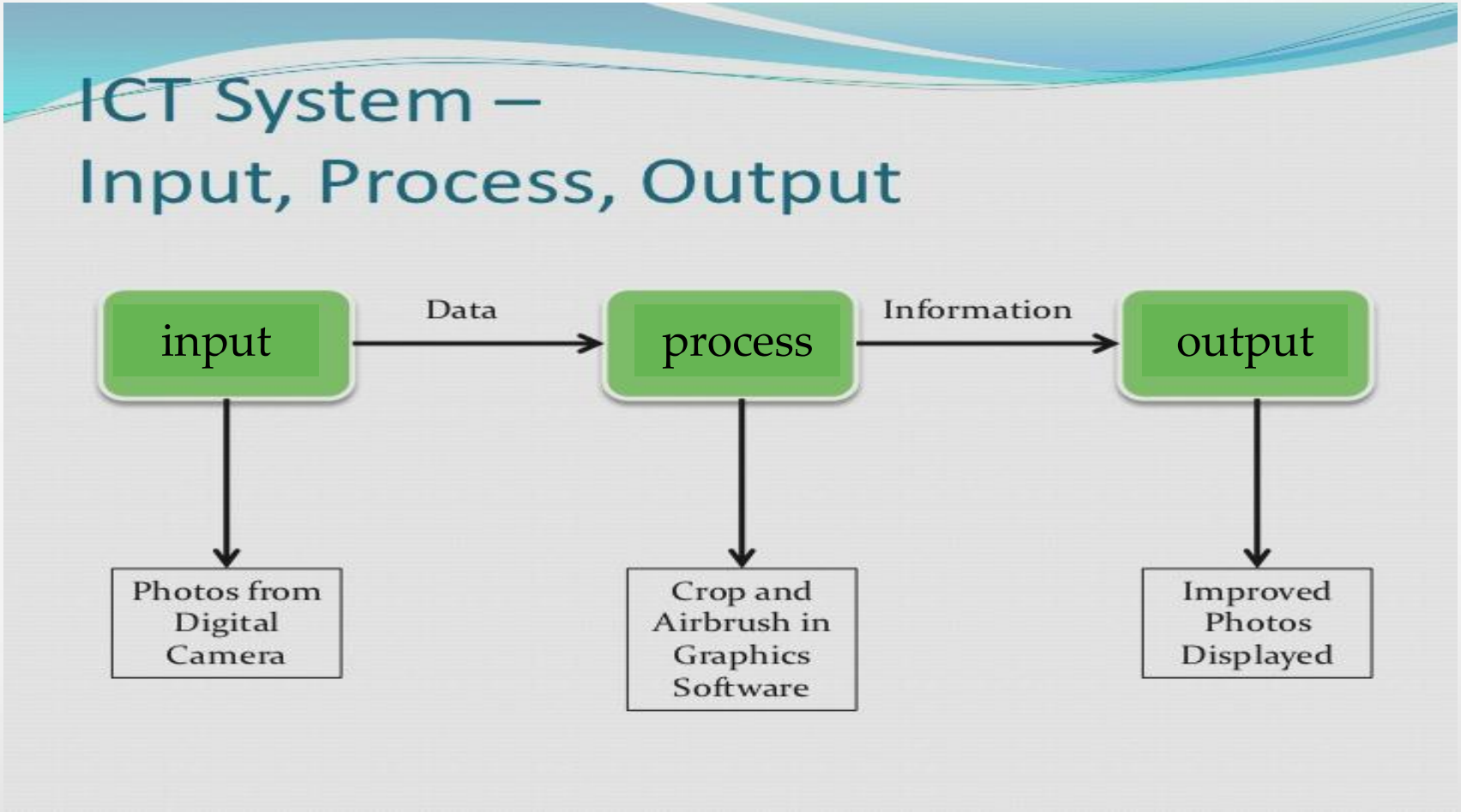
SAMPLE SYSTEMS DIAGRAM

INPUT	PROCESS	OUTPUT
		
Sugar Water Tea bag	1. Put water into a cup 2. Add tea bag and sugar into a cup 3. Stir	Tea drink

8.2.2 Problem Analysis with

Input-Process-Output (IPO) Chart

Examples 6



8.2.2 Problem Analysis with Input-Process-Output (IPO) Chart

→ Math's Problem

Example 1

Problem statement: calculate total of two numbers

Problem Analysis:

INPUT	PROCESS	OUTPUT
no1 no2	Calculate total total = no1+ no2	total

8.2.2 Problem Analysis with

Input-Process-Output (IPO) Chart

→ Math's Problem

Example 2

Problem statement: Calculate the area of a rectangle

Problem Analysis:

INPUT	PROCESS	OUTPUT
width height	Calculate area of rectangle area = width x height	area of rectangle

8.2.3.1 Algorithm

- step-by-step instructions that will transform the input into the output
- can be represent using 2 methods;
    - pseudocode OR
    - flow chart

8.2.3.1 Algorithm

Pseudocode	Flow Chart
• an artificial, informal language (similar to English) used to develop algorithms • a ***notation*** resembling a simplified programming language, used in program design	• a ***graphical representation*** of a algorithm in relation to its sequence of functions • special-purpose **symbols** connected by arrows (flow lines)

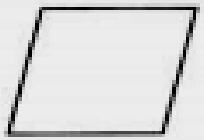
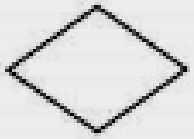
8.2.3.1

Algorithm

Solve a problem using an Algorithm

Conclusion

Flow Chart symbols and Meaning

Symbol,	Name,	Meaning
	Flow line	Used to connect symbols and indicate the flow of logic.
	Terminal	Used to represent the beginning (Start) or the end (End) of a task.
	Input/output	Used for input and output operations, such as reading and printing. The data to be read or printed are described inside.
	Process	Used for arithmetic and data-manipulation operations. The instructions are listed inside the symbol.
	Decision	Used for any logic or comparison operations. Unlike the input/output and processing symbols, which have one entry and one exit flowline, the decision symbol has one entry and two exit paths. The path chosen depends on whether the answer to a question is "yes" or "no."

8.2.3.1

Algorithm

Solve a problem using an Algorithm

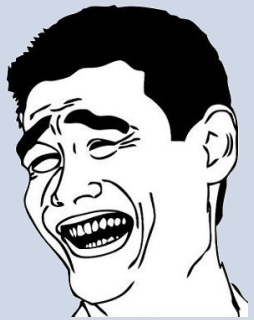
- General's Problem
  - Feeling thirsty
  - Withdrawing money from an ATM machine.
- Mathematical Problem
  - Finding an area of shape (rectangle, triangle)
  - Mean for marks

8.2.3.1

Algorithm

Solve a problem using an Algorithm

- General's Problem → **Feeling Thirsty**
  - Do Problem Analysis (prepare an IPO)

Input	Processing	Output
Water •Buy any drinking water from shop or a vending machine, OR •Get it from friend, OR •…	Drinking •Open the cap and start drinking. •Repeat the process until satisfy.	Happy Face ! 

8.2.3.1

Algorithm

Solve a problem using an Algorithm

- General's Problem
  → **Feeling thirsty**
  - Examples of **Pseudocode**
    - a ***notation*** resembling a simplified programming language, used in program design

```
START
    WALK TO NEAREST SHOP
    BUY A DRINK
    DRINK
    HAPPY FACE
END
```

8.2.3.1

Algorithm

Solve a problem using an Algorithm

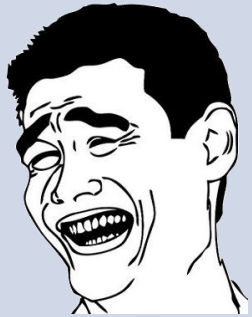
- General Problem → **Feeling thirsty**
  - Examples of **Flow Chart**
    - a ***graphical representation*** of algorithm in relation to its sequence of functions

8.2.3.1

Algorithm

Solve a problem using an Algorithm

- General's Problem → **Withdraw Money from ATM**
  - Do Problem Analysis (prepare an IPO)

Input	Processing	Output
• Auto-teller machine (ATM) card. • PIN Number.	• Insert ATM card on the intake slot. • Enter correct PIN number upon request. • Choose Withdrawal from menu list. • Enter an amount to withdraw. • Collect money from the money output deck. • Collect slip from the slip output deck.	• Money • Transaction Slip  and a Happy Face !

8.2.3.1

Algorithm

Solve a problem using an Algorithm

- General's Problem → **Withdraw Money from ATM**
  - Examples of **Pseudocode**
    - a ***notation*** resembling a simplified programming language, used in program design

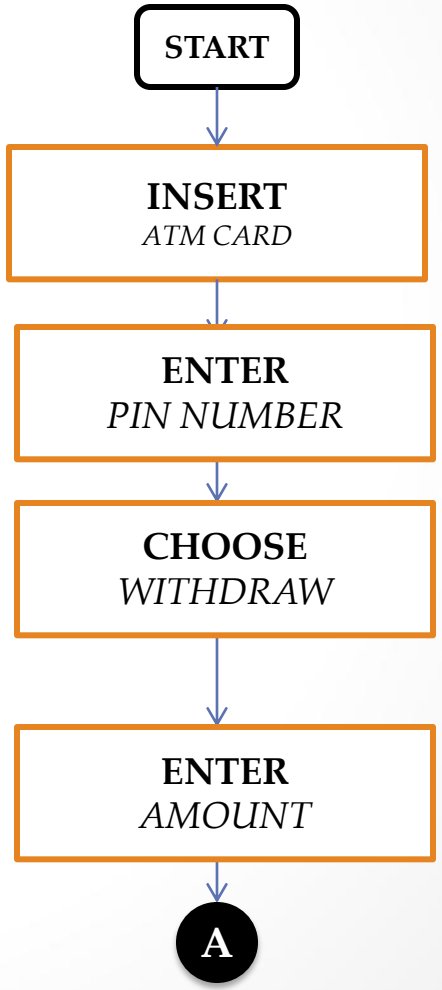
```
START
    INSERT ATM CARD
    ENTER PIN NUMBER
    CHOOSE WITHDRAW
    ENTER AN AMOUNT
    COLLECT MONEY AND SLIP
END
```

8.2.3.1

Algorithm

Solve a problem using an Algorithm

- General's Problem → **Withdraw Money from ATM**
  - Examples of **Flow Chart**
    - a ***graphical representation*** of a algorithm in relation to its sequence of functions

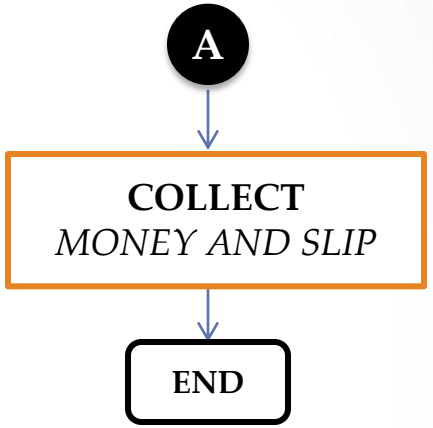


8.2.3.1

Algorithm

Solve a problem using an Algorithm

- General's Problem → **Withdraw Money from ATM**
  - Examples of **Flow Chart**
    - a ***graphical representation*** of algorithm in relation to its sequence of functions

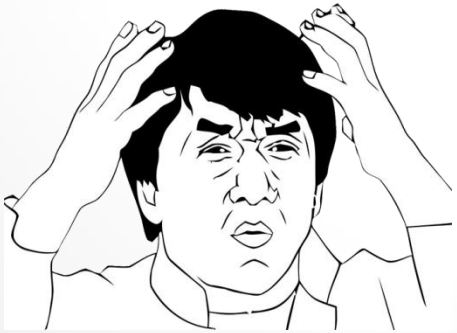


8.2.3.1

Algorithm

Solve a problem using an Algorithm

- Math's Problem → **Calculate total of two numbers.**
  - Examples of **Pseudocode**
    - a ***notation*** resembling a simplified programming language, used in program design



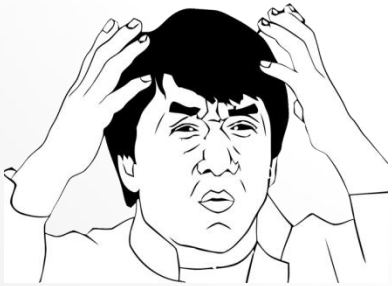
```
START
    enter no1, no2
    Calculate total
    total = no1 + no2
    Display total
END
```

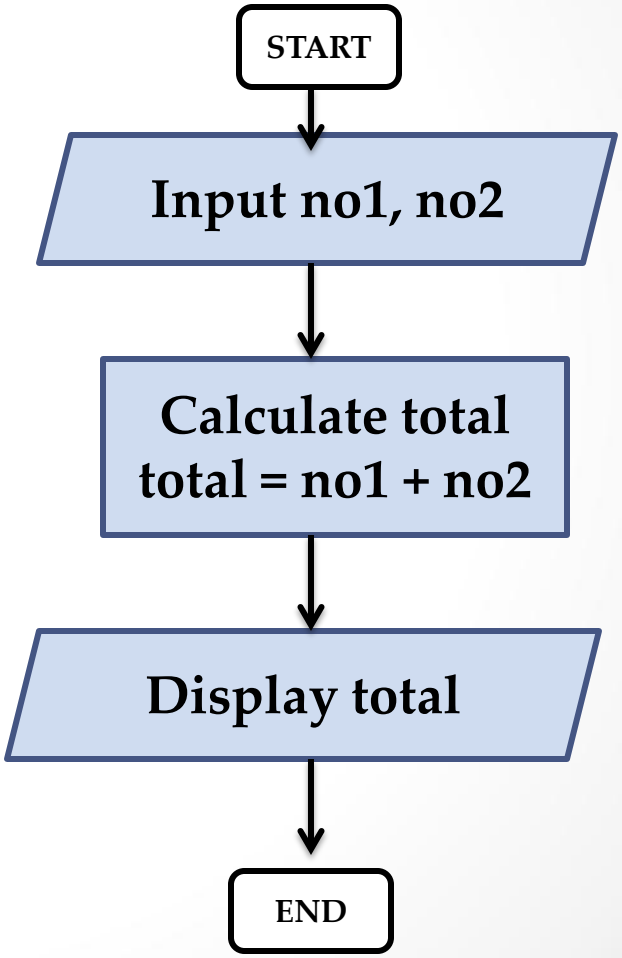
8.2.3.1

Algorithm

Solve a problem using an Algorithm

- Math's Problem → **Calculate total of two numbers.**
  - Examples of **Flow Chart**
    - a ***graphical representation*** of algorithm in relation to its sequence of functions





8.2.3.1

Algorithm

Solve a problem using an Algorithm

- Math's Problem →
  Finding an Area of a Rectangle
  - Examples of
    Pseudocode
    - a ***notation*** resembling a simplified programming language, used in program design

START

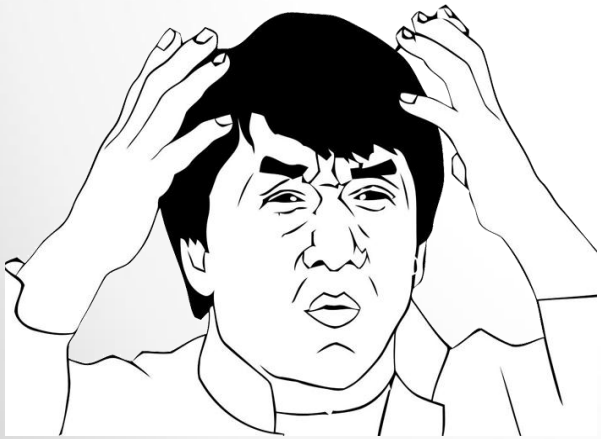
enter width, height

Calculate area of rectangle

area of rectangle = width x height

Display area of rectangle

END



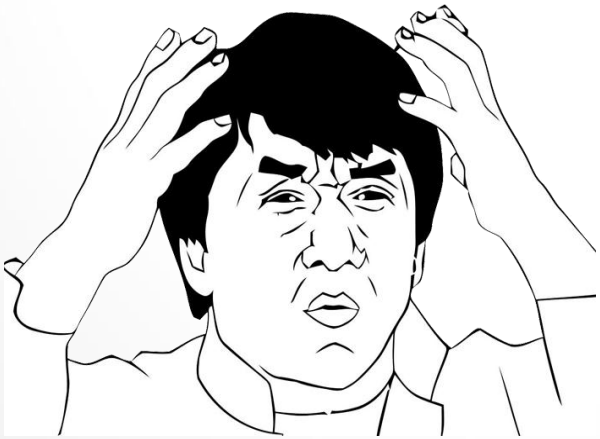
Note:
*** input/enter/key in/ read*
*** output / display / print*

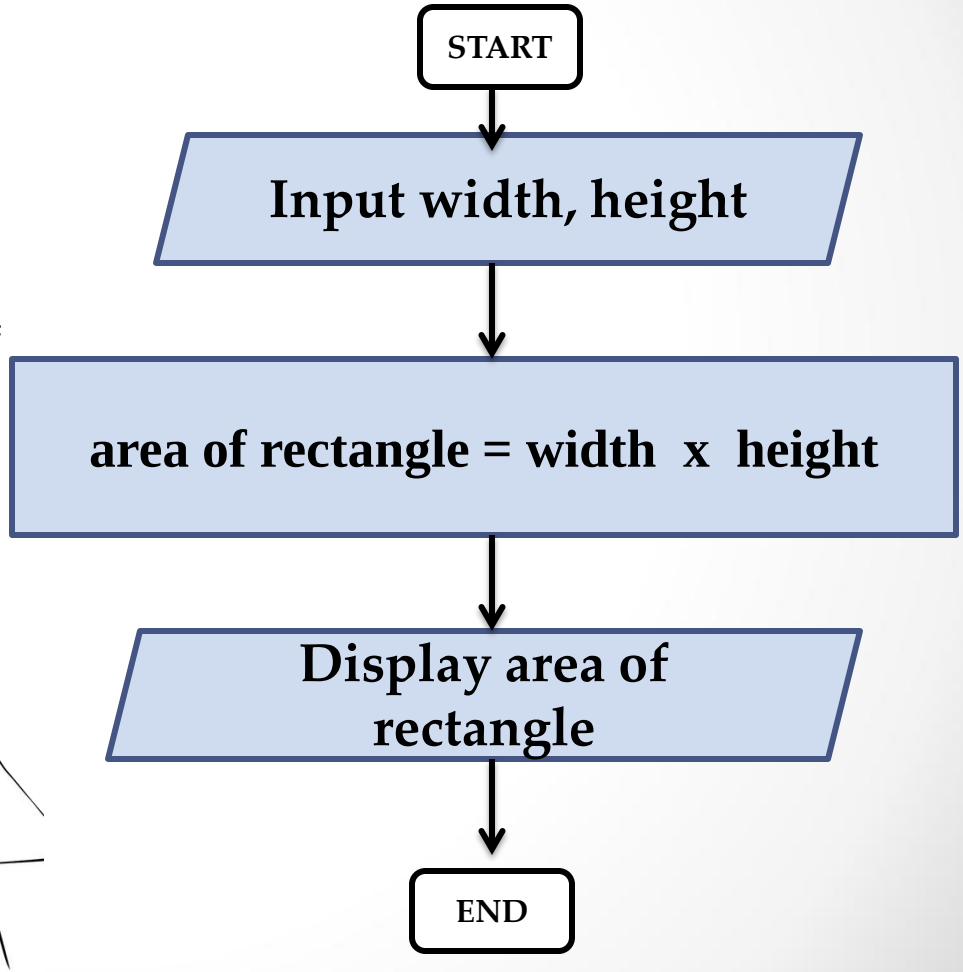
8.2.3.1

Algorithm

Solve a problem using an Algorithm

- Math's Problem → **Finding an Area of Rectangle**
  - Examples of **Flow Chart**
    - a ***graphical representation*** of algorithm in relation to its sequence of functions





8.2.3.1

Algorithm

Solve a problem using an Algorithm

- Math's Problem → **Mean for 2 Marks**
  - Do Problem Analysis (prepare an IPO)

Input	Processing	Output

8.2.3.1

Algorithm

Solve a problem using an Algorithm

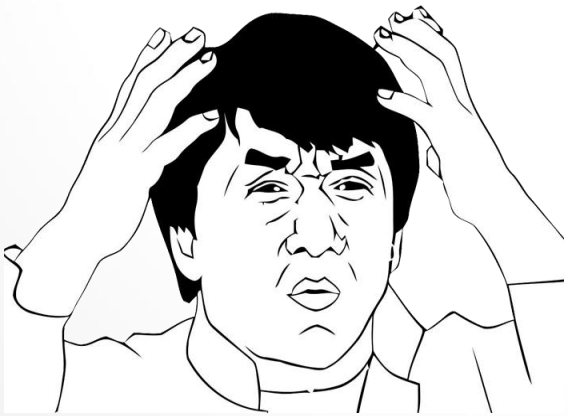
- Math's Problem →

  Mean for 2 Marks

  - Examples of

    Pseudocode

    - a ***notation*** resembling a simplified programming language, used in program design



START

..

..

..

..

..

END

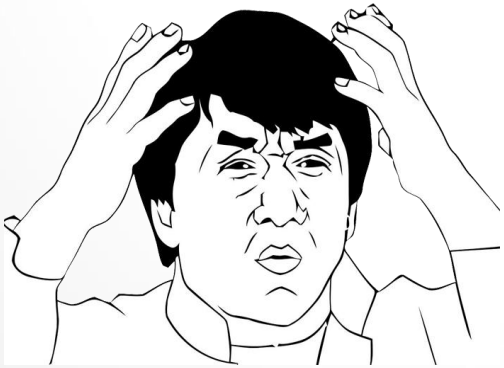
8.2.3.1

Algorithm

Solve a problem using an Algorithm

- Math's Problem → **Mean for 2 Marks**
  - Examples of **Flow Chart**
    - a ***graphical representation*** of algorithm in relation to its sequence of functions

START



END

8.2.3.2

Algorithm Control Structures

Sequence	Selection	Repetition
series of statements that **execute one after another**	statement is used to determine which of two different statements to execute **depending on** certain **conditions**	statement is used to **repeat statements while certain conditions are met**

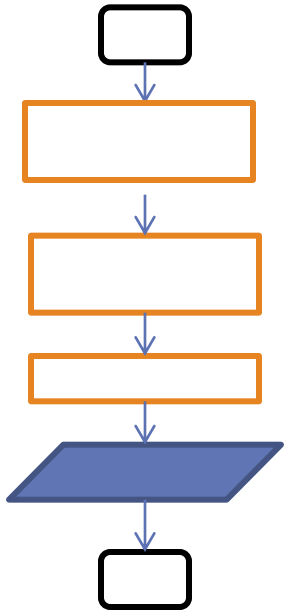
8.2.3.2

An Overview of Control Structures

Sequence

series of statements that **execute one after another**

- all examples from previous slide have already shown a sequence control structures.

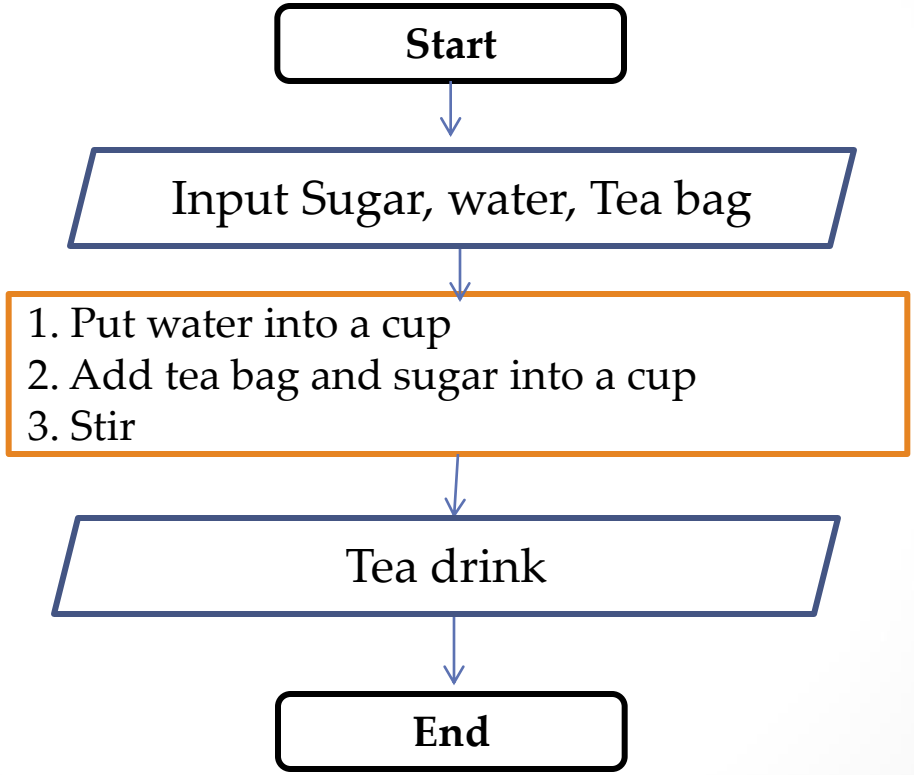


An arrow shows the execution of every block (in a flow chart), one after another, this is a **SEQUENCE** control structure to solve a problem.

8.2.3.2

An Overview of Control Structures

- Example: How to make a tea drink

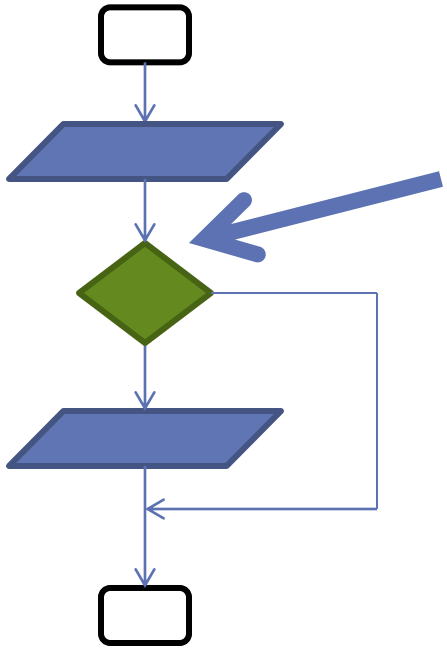


8.2.3.2

An Overview of Control Structures

Selection

statement is used to determine which of two different statements to execute **depending on** certain **conditions**



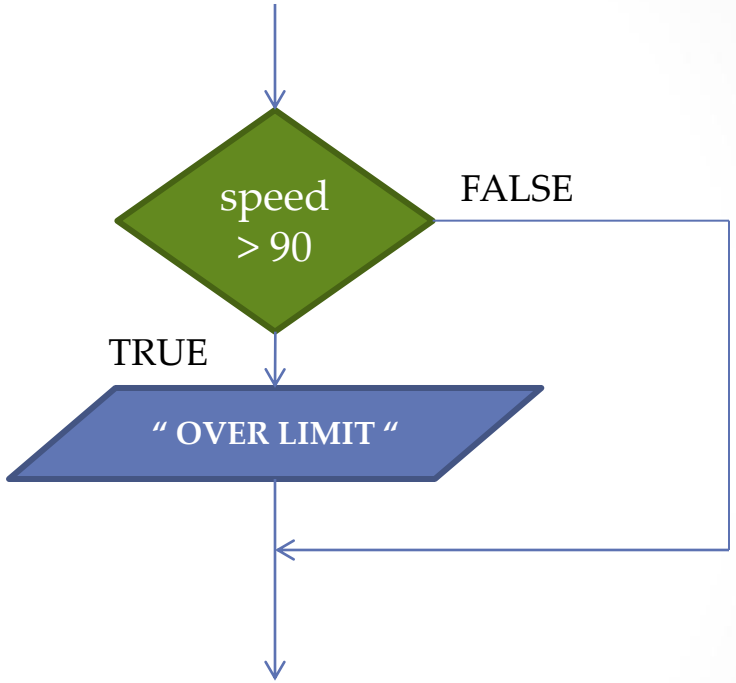
A diamond (in this case is a **SELECTION**) determine whether the resulting process is TRUE or FALSE – and it will flow to one direction

8.2.3.2

An Overview of Control Structures

Selection

statement is used to determine which of two different statements to execute **depending on** certain **conditions**



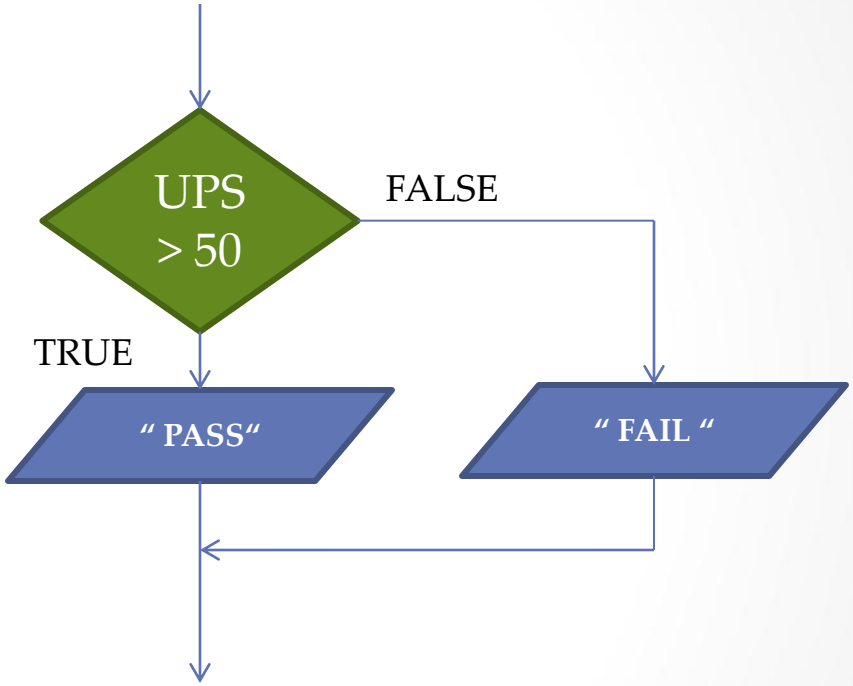
The program will display the message "Over limit" if speed > 90

8.2.3.2

An Overview of Control Structures

Selection

statement is used to determine which of two different statements to execute **depending on** certain **conditions**



The program will display the message "PASS" if UPS > 50, otherwise, display message "FAIL".

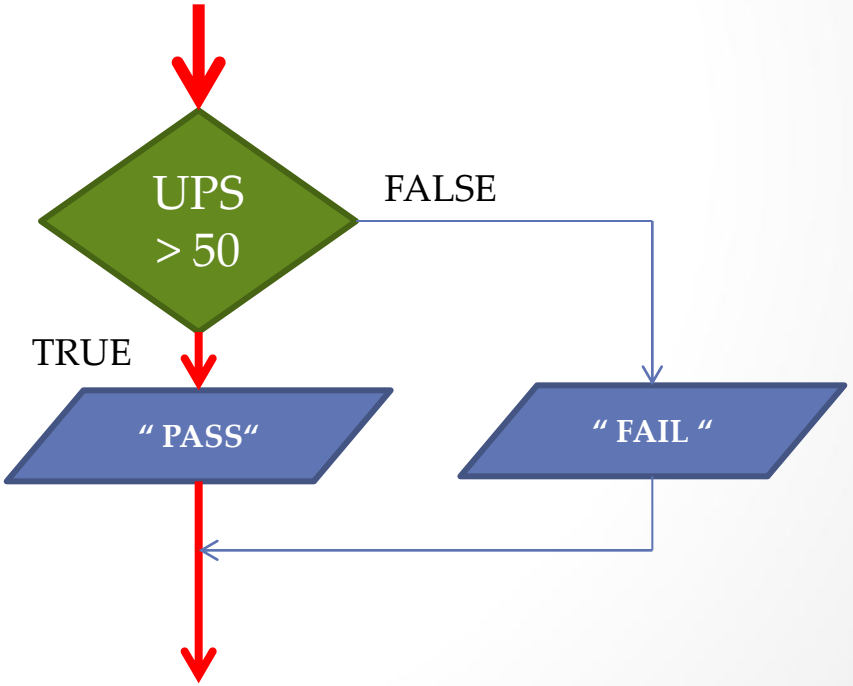
8.2.3.2

An Overview of Control Structures

Selection

statement is used to determine which of two different statements to execute **depending on** certain **conditions**

If the program receive an input of **UPS = 60** , then the data flow is presented by the dotted line ..



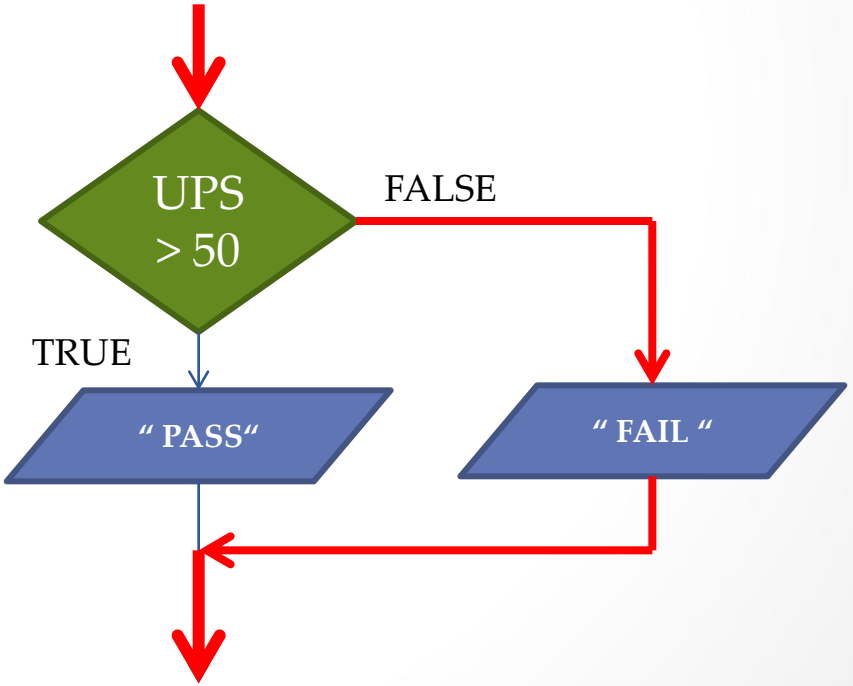
8.2.3.2

An Overview of Control Structures

Selection

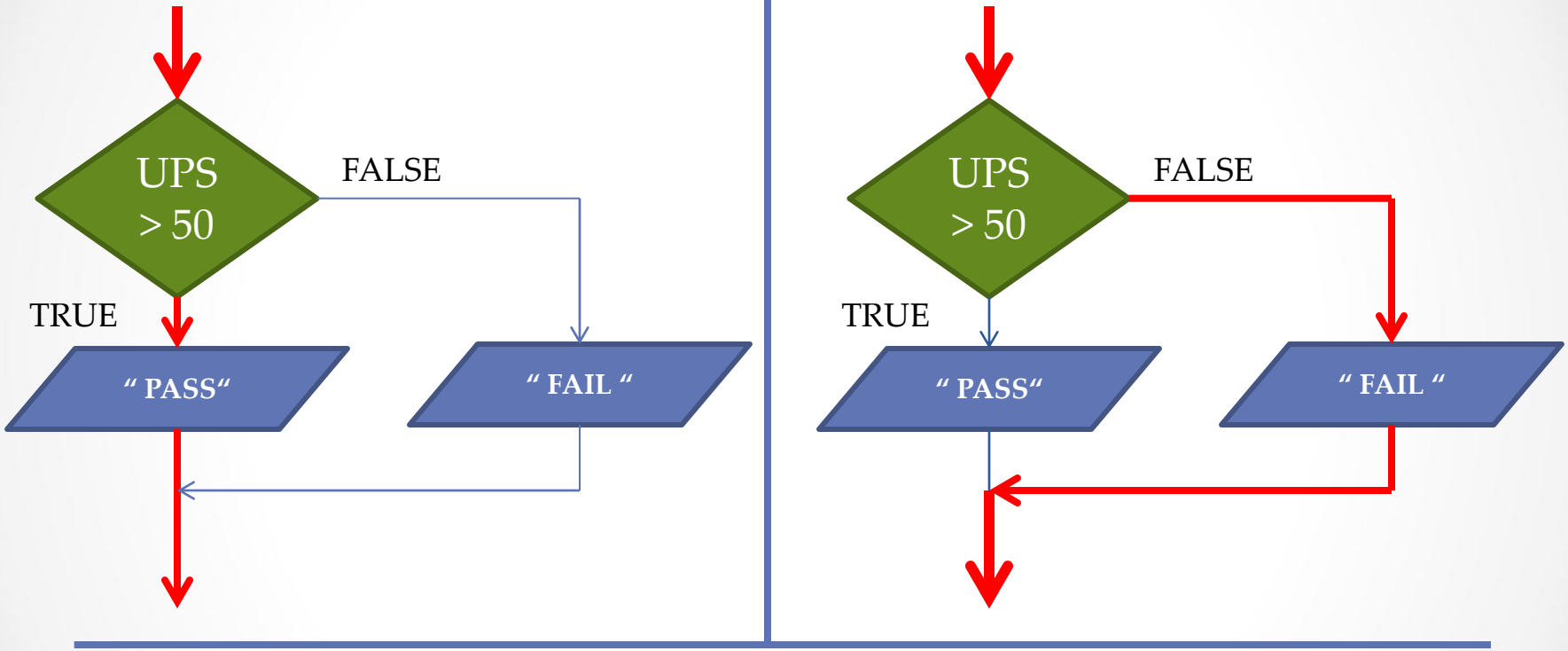
statement is used to determine which of two different statements to execute **depending on** certain **conditions**

If the program receive an input of **UPS = 30** , then the data flow is presented by the red line ..



8.2.3.2

An Overview of Control Structures



Conclusion For Control Structure : **SELECTION**

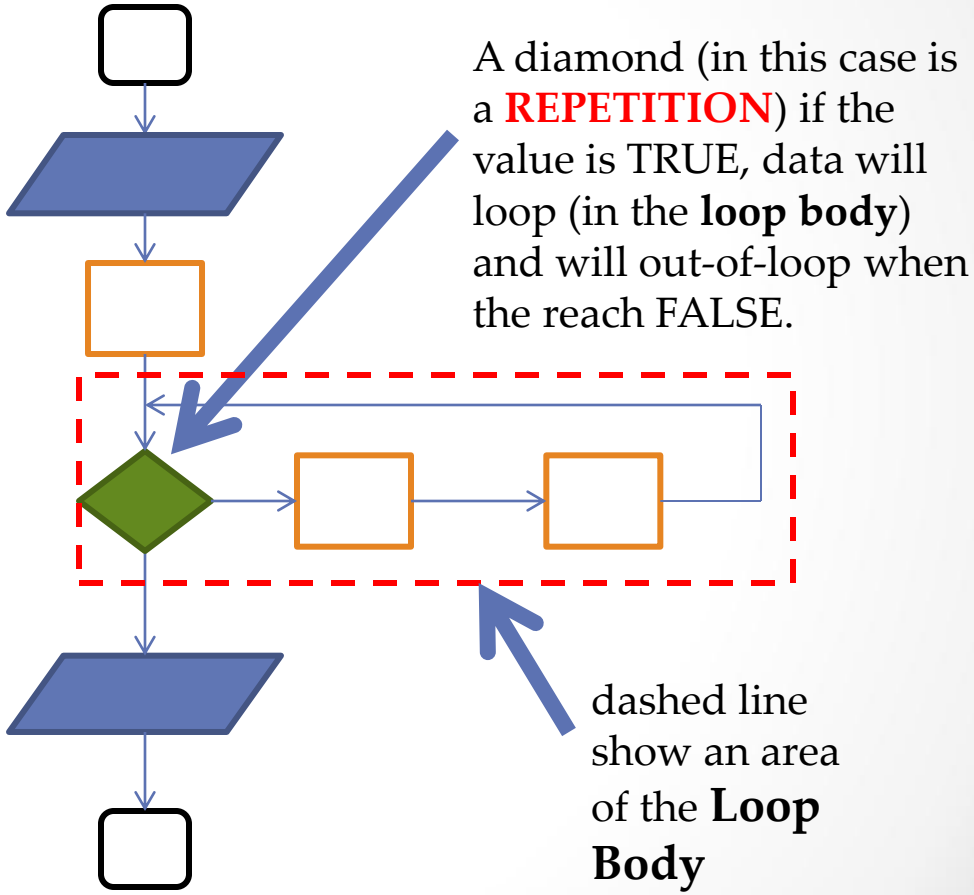
decisions (selections): statement(s) is (are) executed if certain condition gives TRUE or FALSE value.

8.2.3.2

An Overview of Control Structures

Repetition

statement is used to **repeat statements while certain conditions are met**



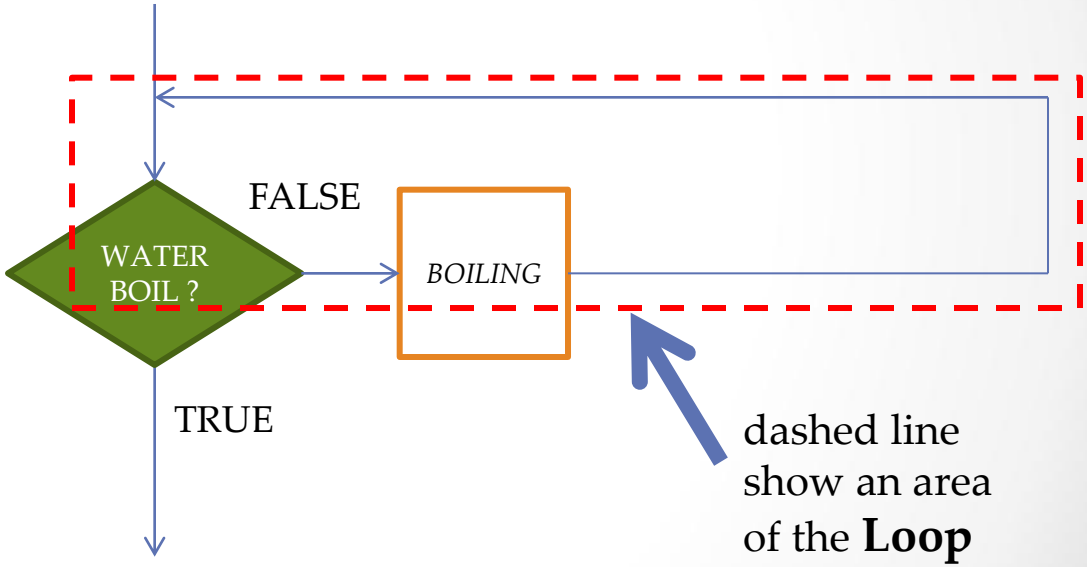
8.2.3.2

An Overview of Control Structures

Repetition

statement is used to **repeat statements while certain conditions are met**

If water boil is FALSE (meaning not yet boil to 100 degree Celsius, temp < 100), keep it boiling until the water is boiled (temp = 100 deg. Celsius)



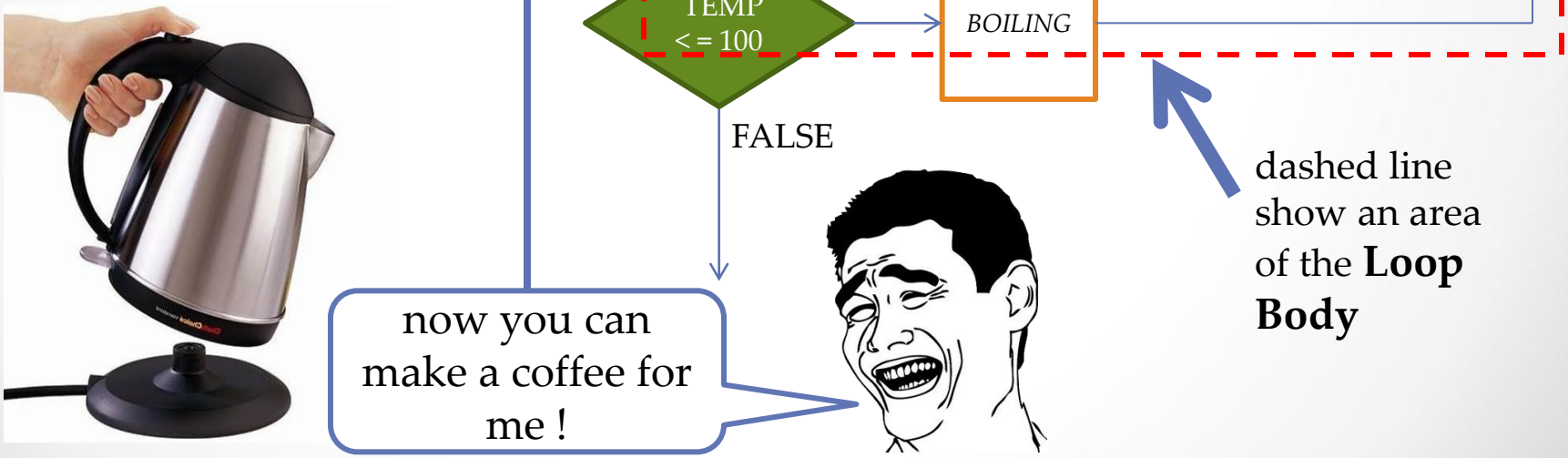
8.2.3.2

An Overview of Control Structures

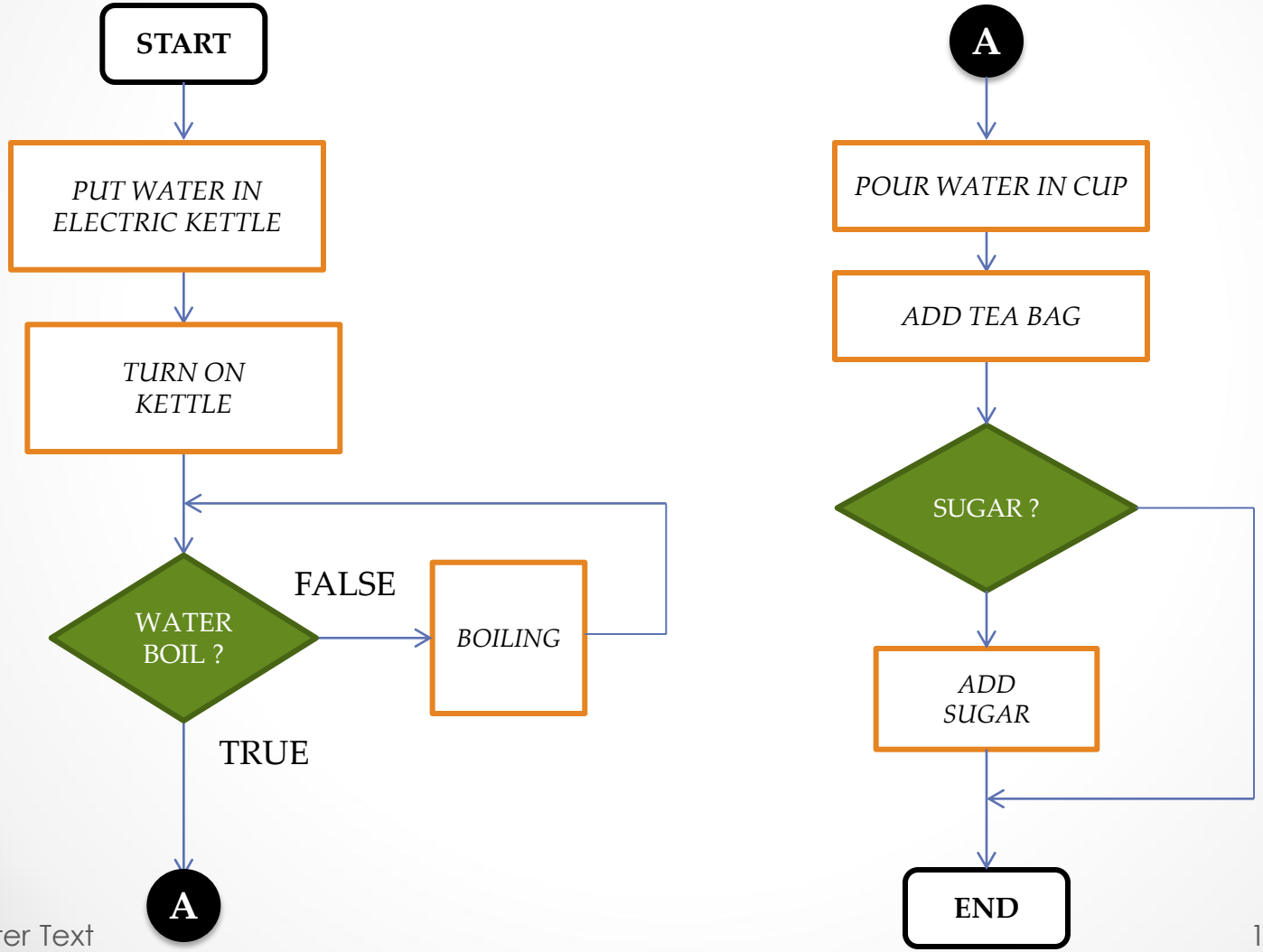
Repetition

statement is used to **repeat statements while certain conditions are met**

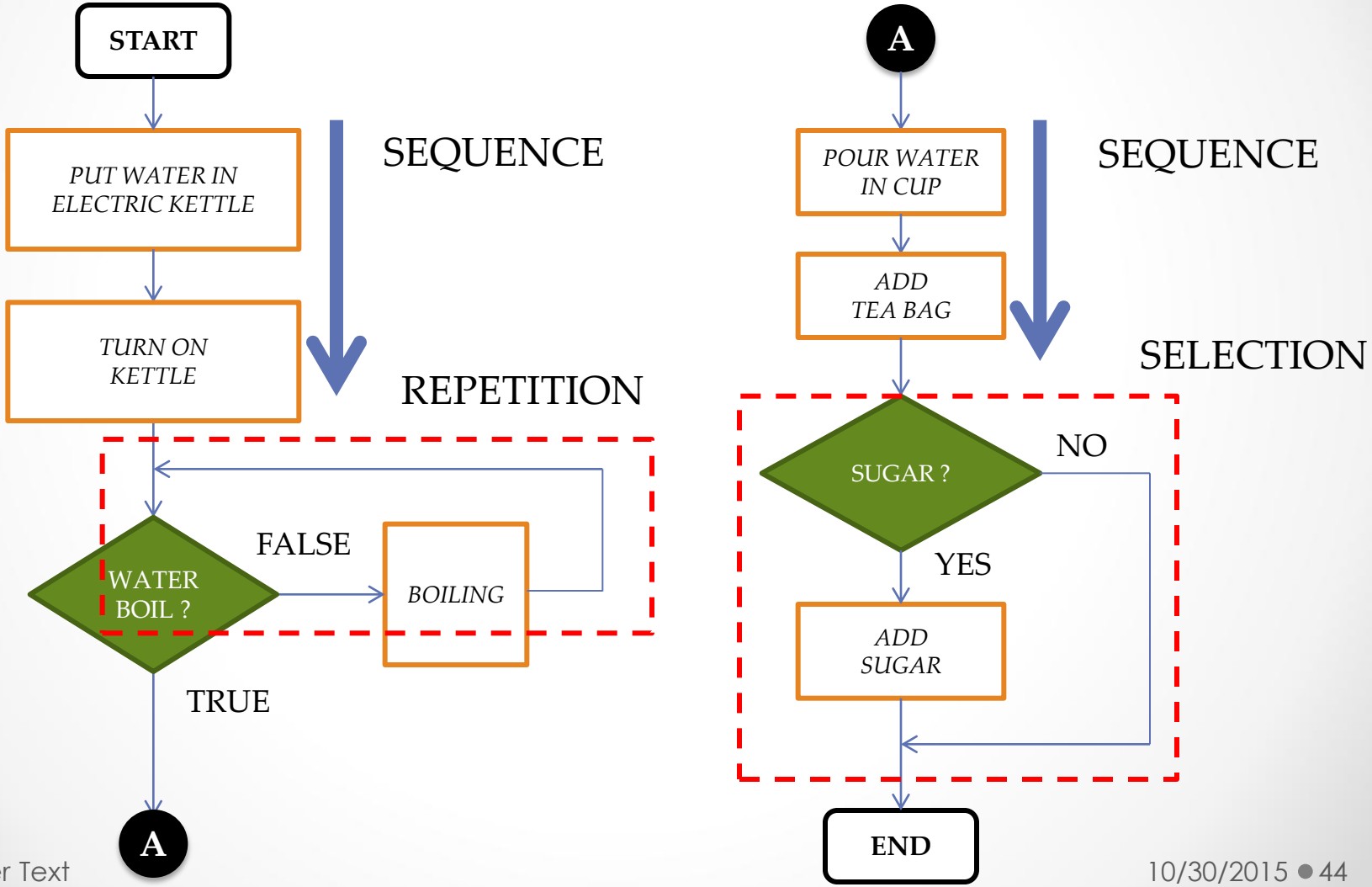
If temp < = 100 is TRUE (meaning water still not boil) keep it boiling until the temp reach 100, then it will stop.



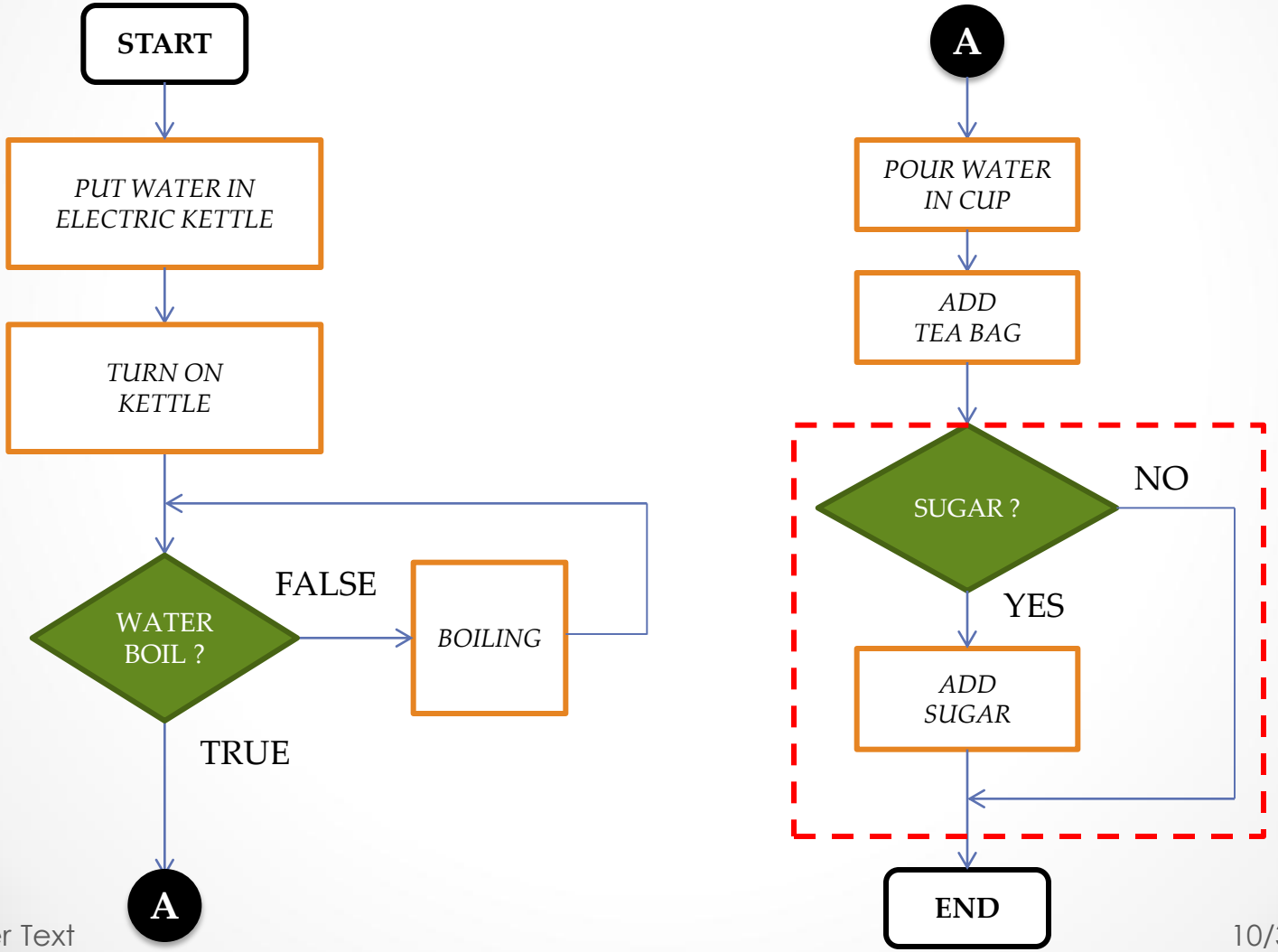
8.2.3.2 Apply Appropriate Control Structures



8.2.3.2 Apply Appropriate Control Structures



8.2.3.2 Apply Appropriate Control Structures



EXERCISE

- 1. (a) Write a pseudocode to display " Kolej Matrikulasi Perak".

  (b) Convert the pseudocode info flowchart.

- 2. (a) Write a pseudocode to calculate average of two numbers.

  (b) Convert the pseudocode info flowchart.