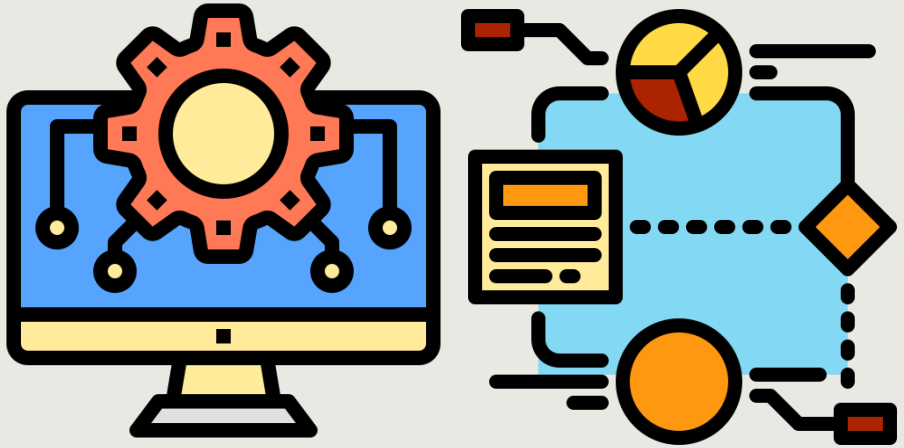
Praktikum Algoritma Pemrograman

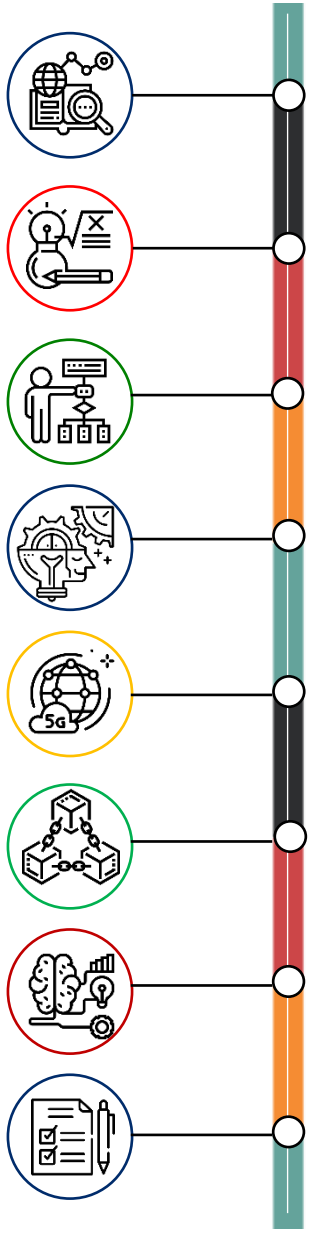
Dr. Sandi Rahmadika, S.T., M.T., M.Eng.
NIP. 199103242022031008

E-mail: sandi@ft.unp.ac.id



Lecture 1 – Introduction (Algorithm & Flowchart)

Outline



What is Algorithm?

- Definition
- Objective & Goals

<u>Properties of Algorithm</u>

Correct and Incorrect Algorithms

- How to define?
- Hot to select?

Fungsi

- Manfaat & Why it Matters

Basic Implementation

- Use Cases
- Example of Algorithms

Flowchart

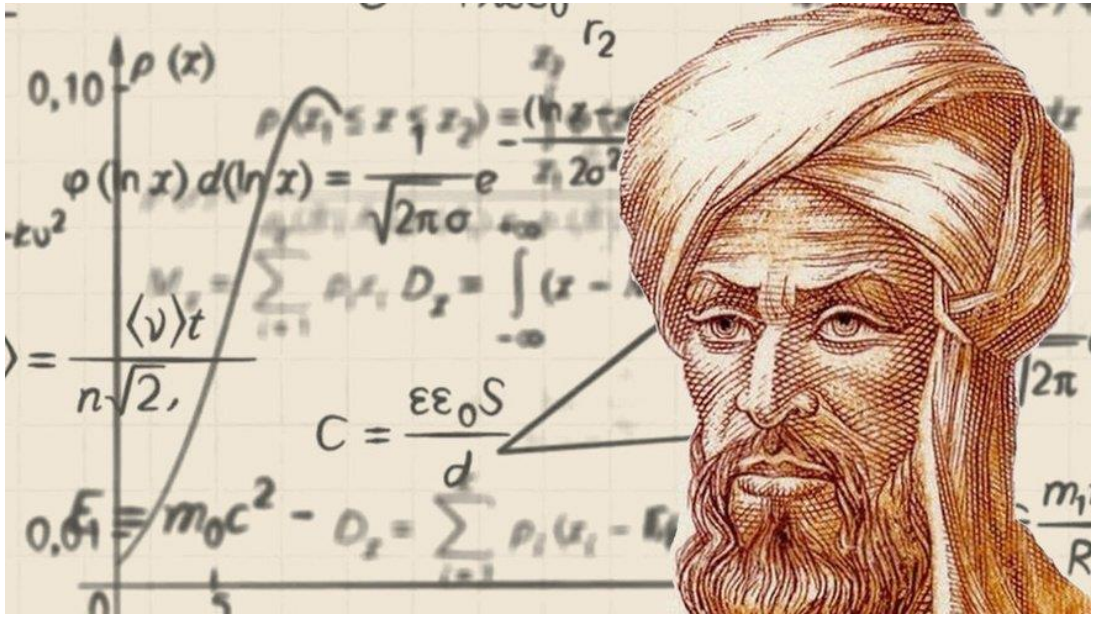
Symbols of Flowchart

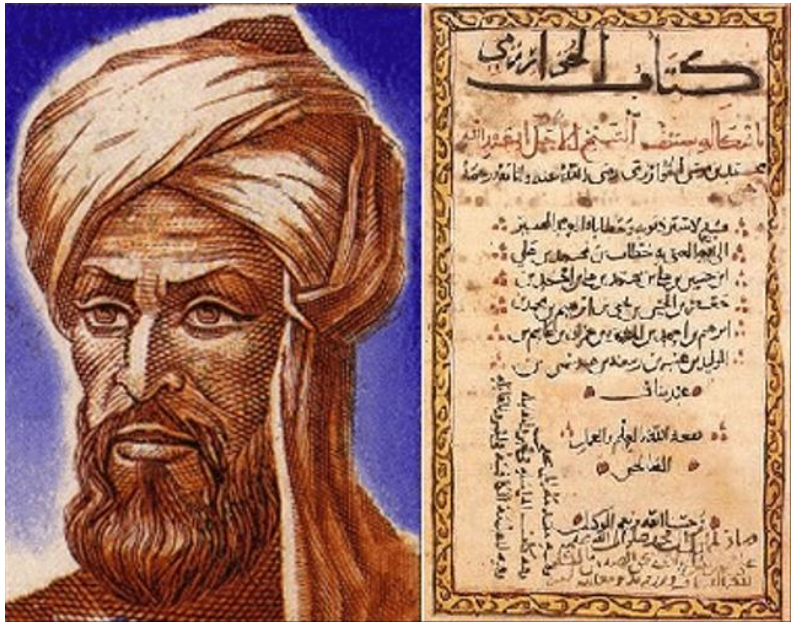
- Examples

Next Week Prerequisites

What is an Algorithm?

- Named after Persian Mathematician **Mohammad Al-Khwarizmi**
  - A sequential solution of any problem
  - Written in human understandable form
  - Requires a clear understanding of the problem





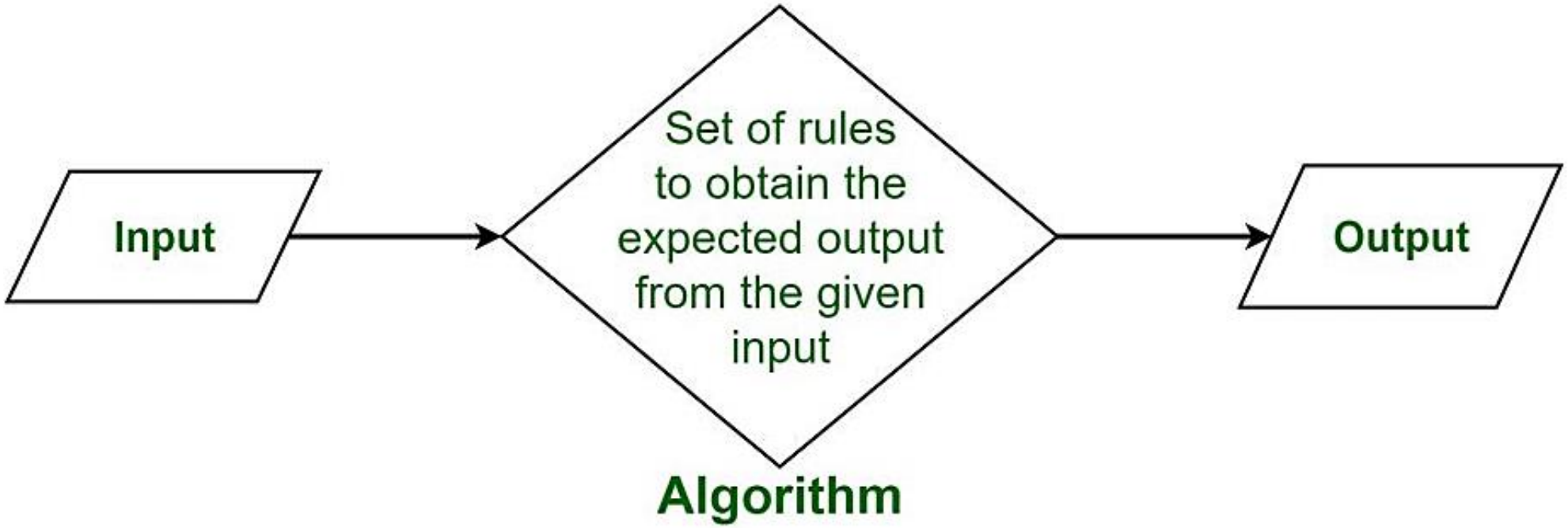
Definisi

- **Algoritma** – Urutan langkah untuk menyelesaikan masalah secara sistematis dan logis

- **Program** – Kumpulan instruksi / perintah computer dengan Bahasa tertentu yang berfungsi menghubungkan user dengan computer.

Atau bisa juga disebut implementasi dari Bahasa pemrograman

What is Algorithm



- Algorithm must be:
  - **Correct:** For each input produce an appropriate output
  - **Efficient**: Run as quickly as possible, and use as little memory as possible

What is Algorithm … (2)

- A well-defined computational procedure that takes some value, or set of values, as input and produces some value, or set of values, as output:

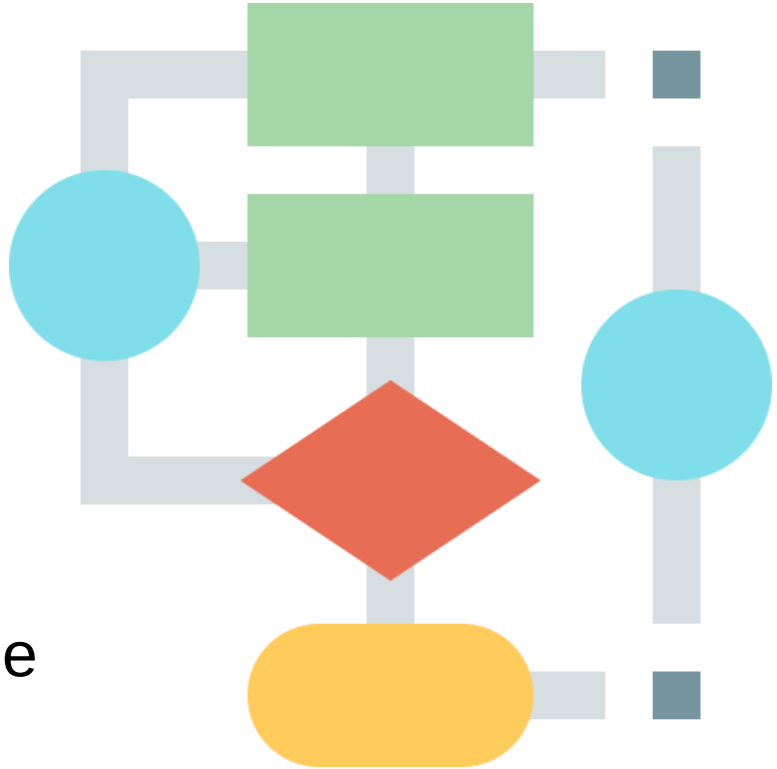
- Written in a pseudo code which can be implemented in the language of programmer's choice (ex. C++, C#, Java, Python, etc.)


```
Data: this text
Result: how to write algorithm with LaTeX2e
initialization;
while not at end of this document do
    read current;
    if understand then
        go to next section;
        current section becomes this one;
    else
        go back to the beginning of current section;
    end
end
```

Algorithm 1: How to write algorithms

Properties of Algorithm

- Finiteness
- Properly defined
- Input
- Output
- Effectiveness
- Independent to any other programming language

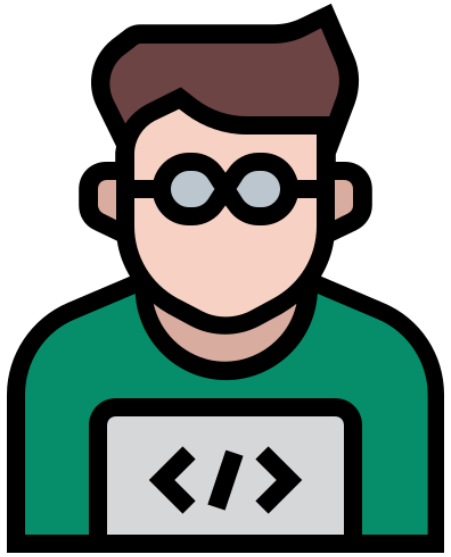


Correct and Incorrect Algorithms

- Algorithm is correct if, for every input instance, it ends with the correct output. We say that a correct algorithm solves the given computational problem

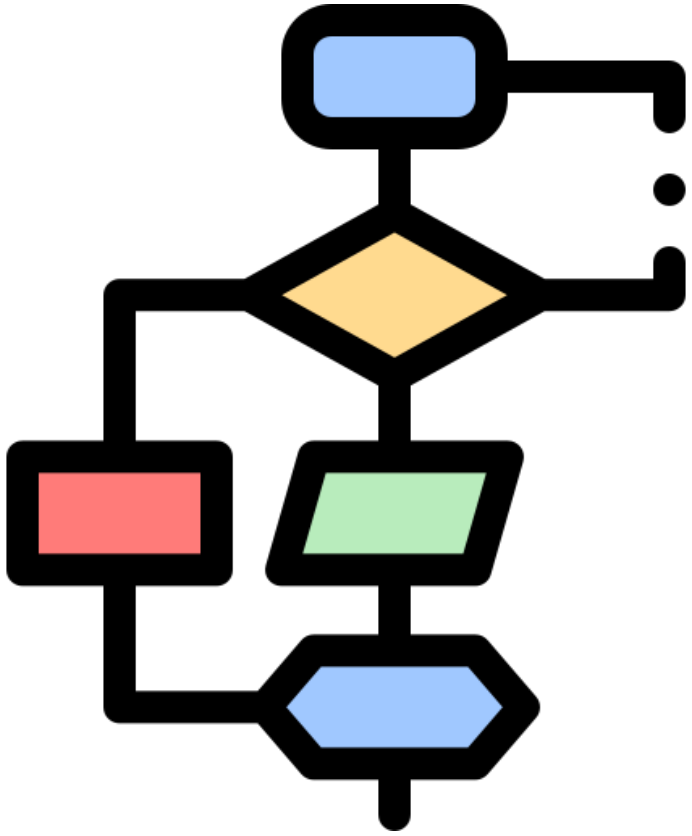
- An incorrect algorithm might not end at all on some input instances, or it might end with an answer other than the desired one

- We shall be concerned only with correct algorithms

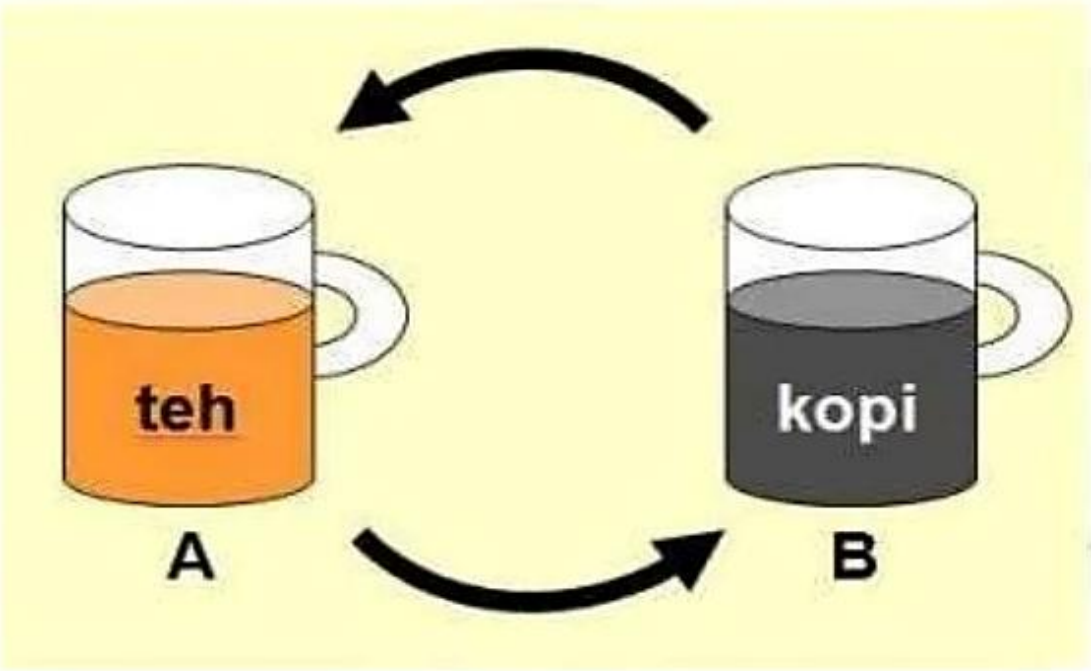


Fungsi Algoritma Pemrograman

- Memudahkan dalam pembuatan program
- Bisa mengatasi segala masalah dengan logika dan urut
- Program yang ada menjadi lebih terstruktur dengan rapi sehingga dapat lebih mudah untuk dipahami ataupun dikembangkan
- Meminimalisir penulisan program yang berulang-ulang
- Dokumentasi lebih mudah



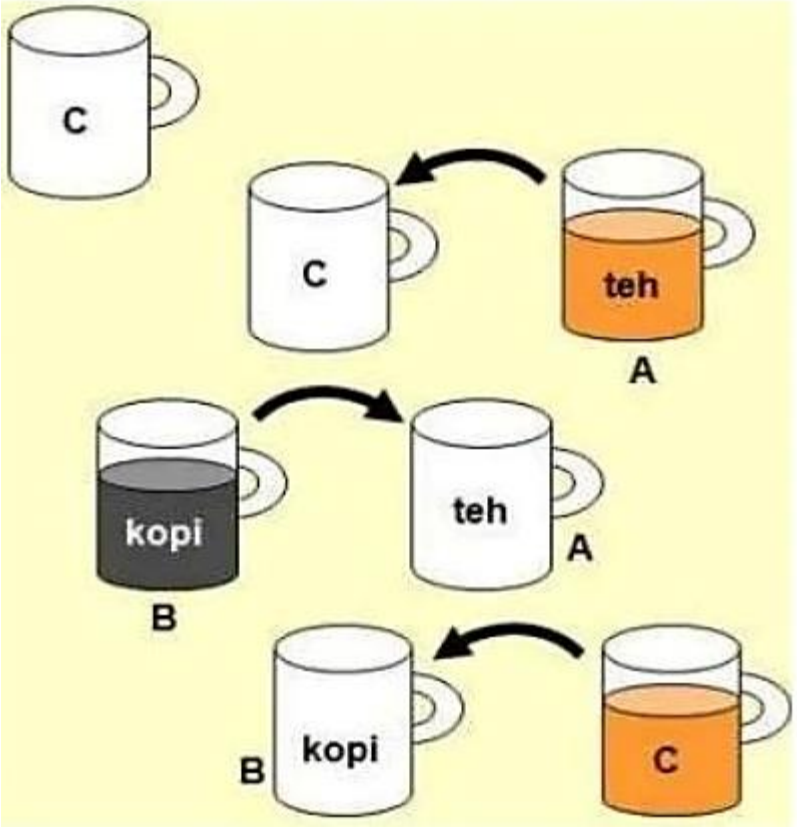
Basic Implementation



Bagaimana cara menukarkan isi gelas A yang semula berisi air teh menjadi berisi air kopi dan gelas B yang semula air kopi menjadi air teh ?

Penjelasan

Basic Implementation … (2)



X Close

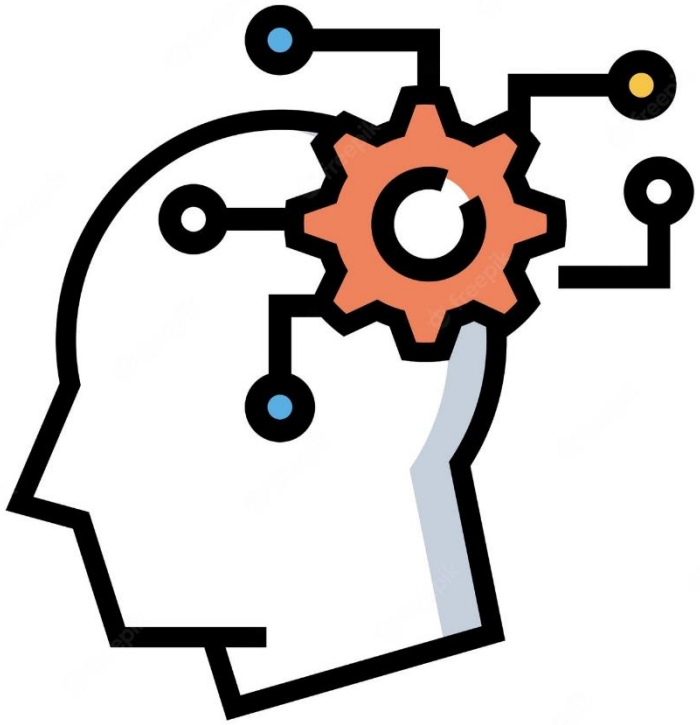
Cara penyelesaian permasalahan gambar tersebut yaitu diperlukan gelas tambahan yang kita namakan gelas C sebagai tempat penampungan sementara.

Berikut detail Algoritmanya:

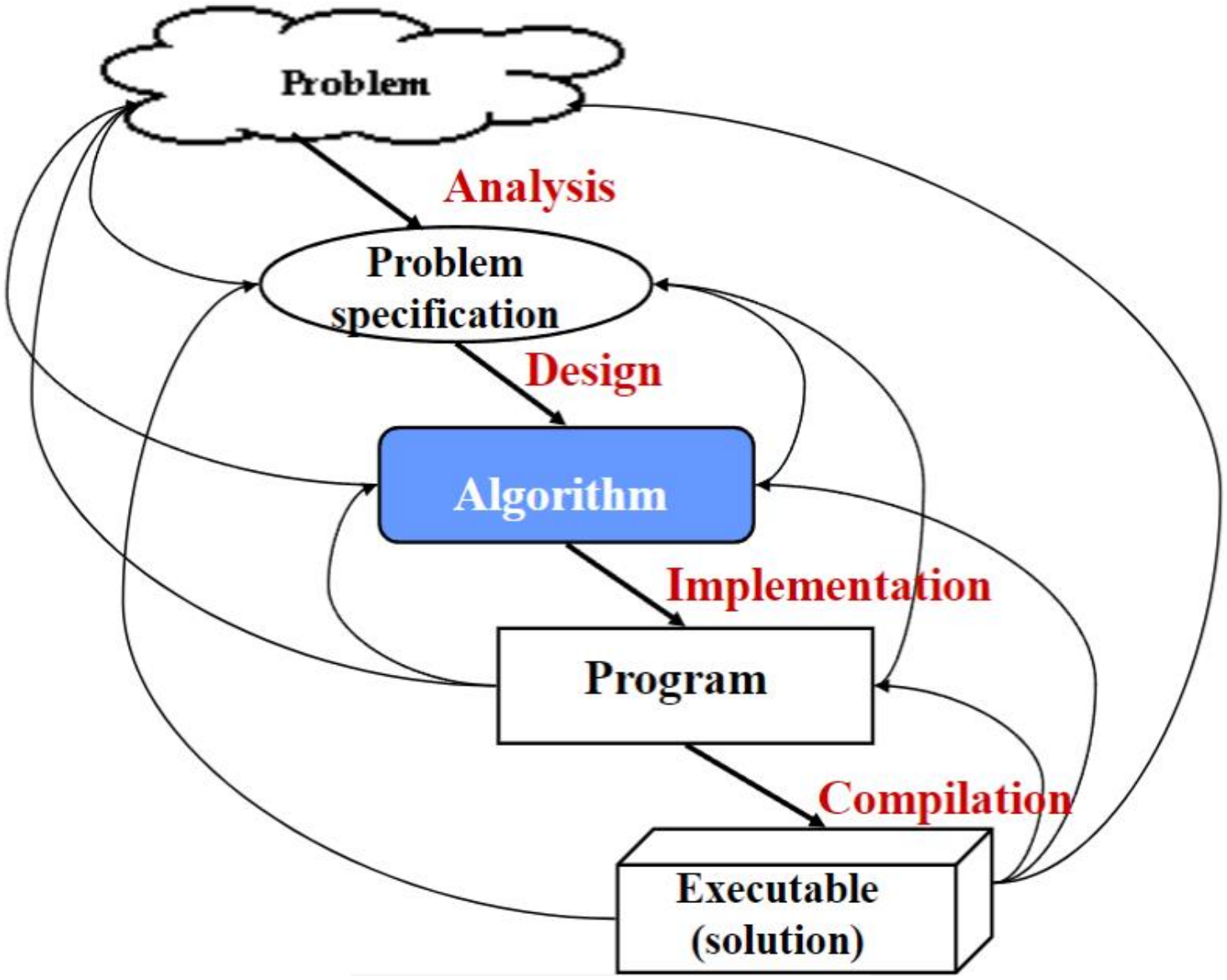
1. Siapkan gelas cadangan C
2. Tuangkan air teh dari gelas A ke dalam gelas C (gelas A menjadi kosong).
3. Tuangkan air kopi dari gelas B ke dalam gelas A (gelas B menjadi kosong).
4. Tuangkan air teh dari gelas C ke dalam gelas B.

Problems

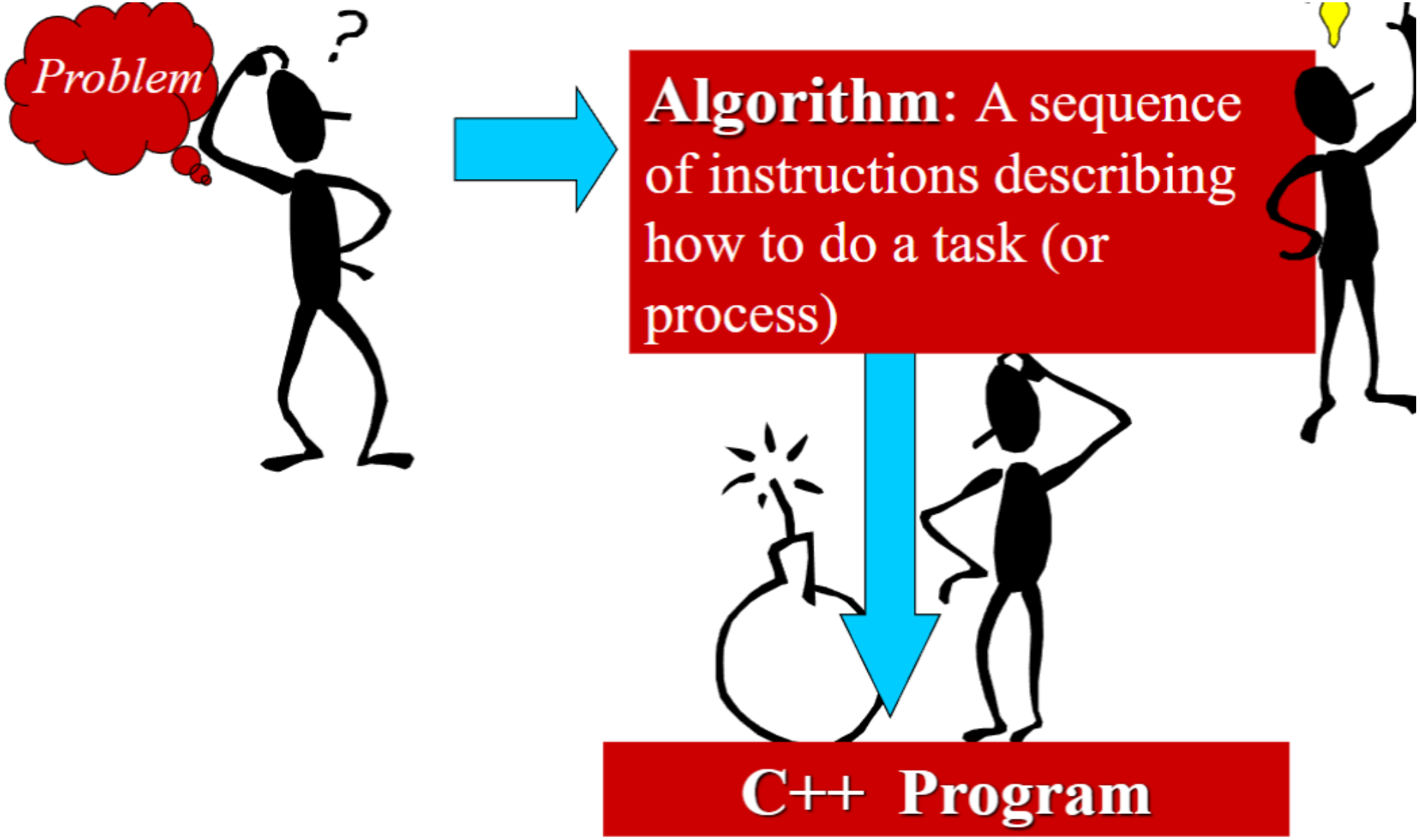
- We need to solve a computational problem
  - "Convert a weight in pounds to Kg"
- An algorithm specifies how to solve it, e.g.:
  - 1. Read weight-in-pounds
  - 2. Calculate weight-in-Kg = weight-in-pounds * 0.455
  - 3. Print weight-in-Kg
- A computer program is a computer-executable description of an algorithm



The Problem-Solving Process



From Algorithms to Programs

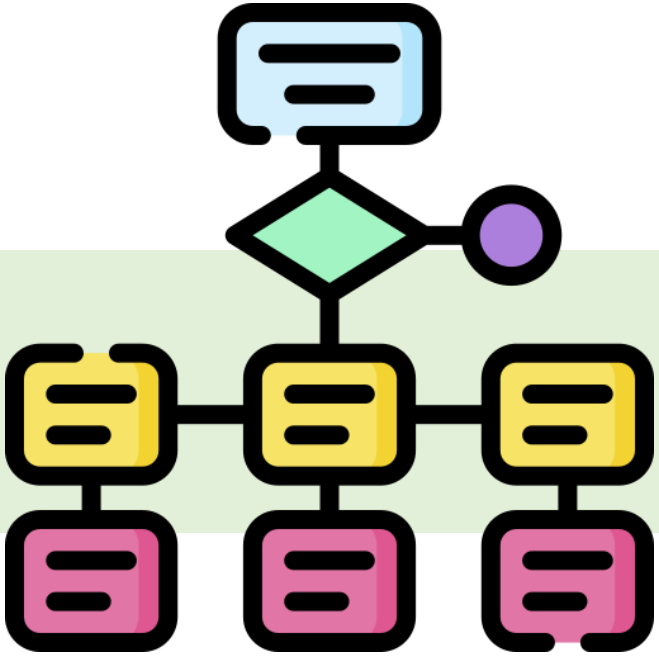


Practical Examples

- **Internet and Networks**
  - The need to access large amount of information with the shortest time
  - Problems of finding the best routes for the data to travel
  - Algorithms for searching this large amount of data to quickly find the pages on which particular information resides

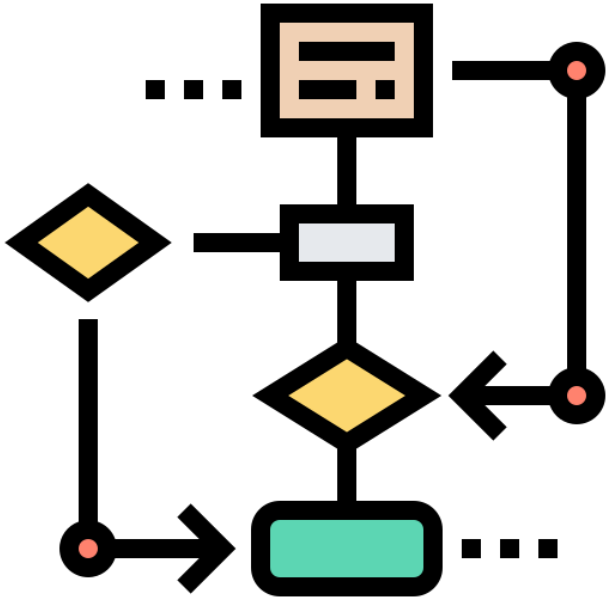
- **Electronic Commerce**
  - The ability of keeping the information (credit card numbers, passwords, bank statements) private, safe, and secure
  - Algorithms involves encryption / decryption techniques

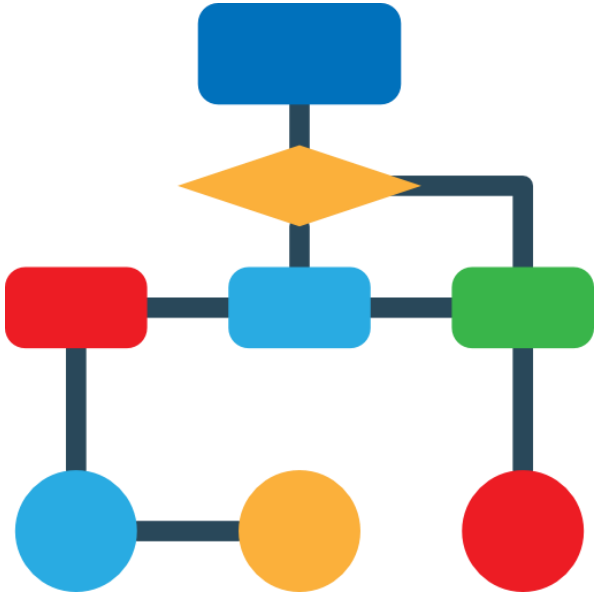
Flowchart



What is a Flowchart?

- Diagrammatic representation of algorithm
  - An important programming tool
  - Solving a problem using figures
  - Different figures having different functions

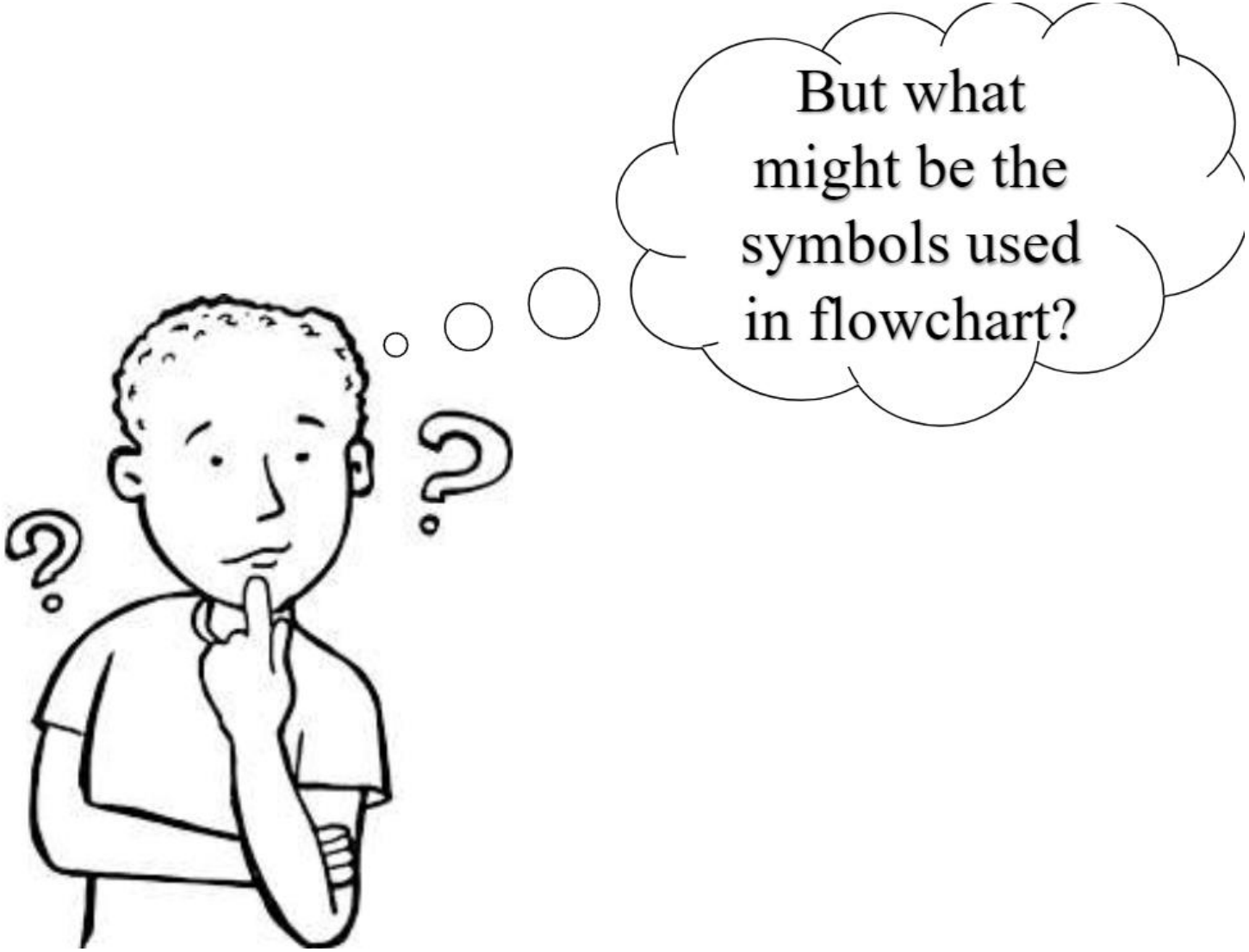




Merits & Demerits of Flowchart

Merits	Demerits
▶ Easy to explain program logic	▶ Time consuming monotonous
▶ Makes coding effective and faster	▶ Monotonous job
▶ Effective communication	▶ Difficult to maintain
▶ Different symbols used	▶ Occupies space while documentation
▶ Serve as documentation	▶ Translation to computer is difficult
▶ Easy to detect, locate, and remove bugs in a program	

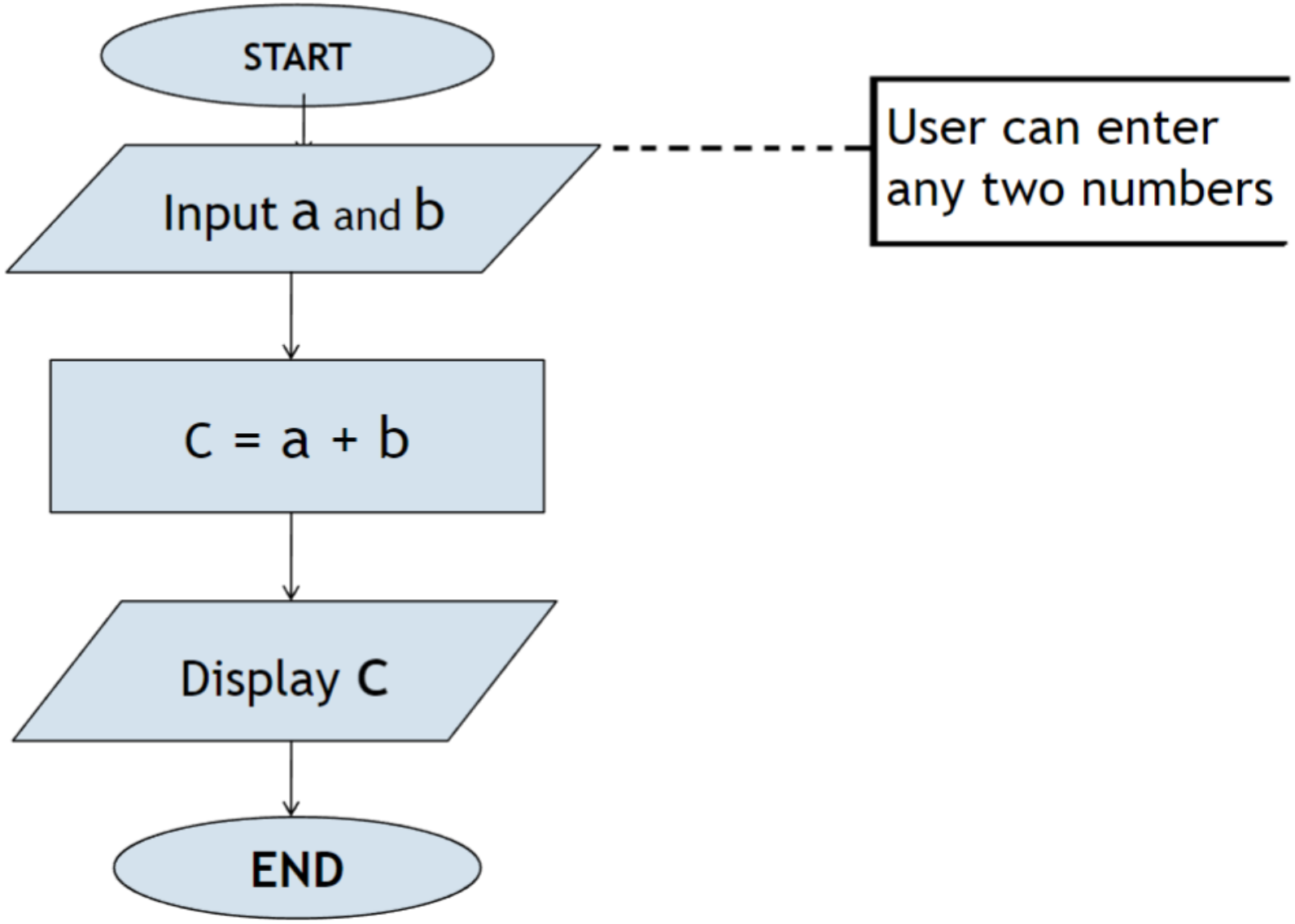
Flowchart's Symbols



Flowchart's Symbols …(2)

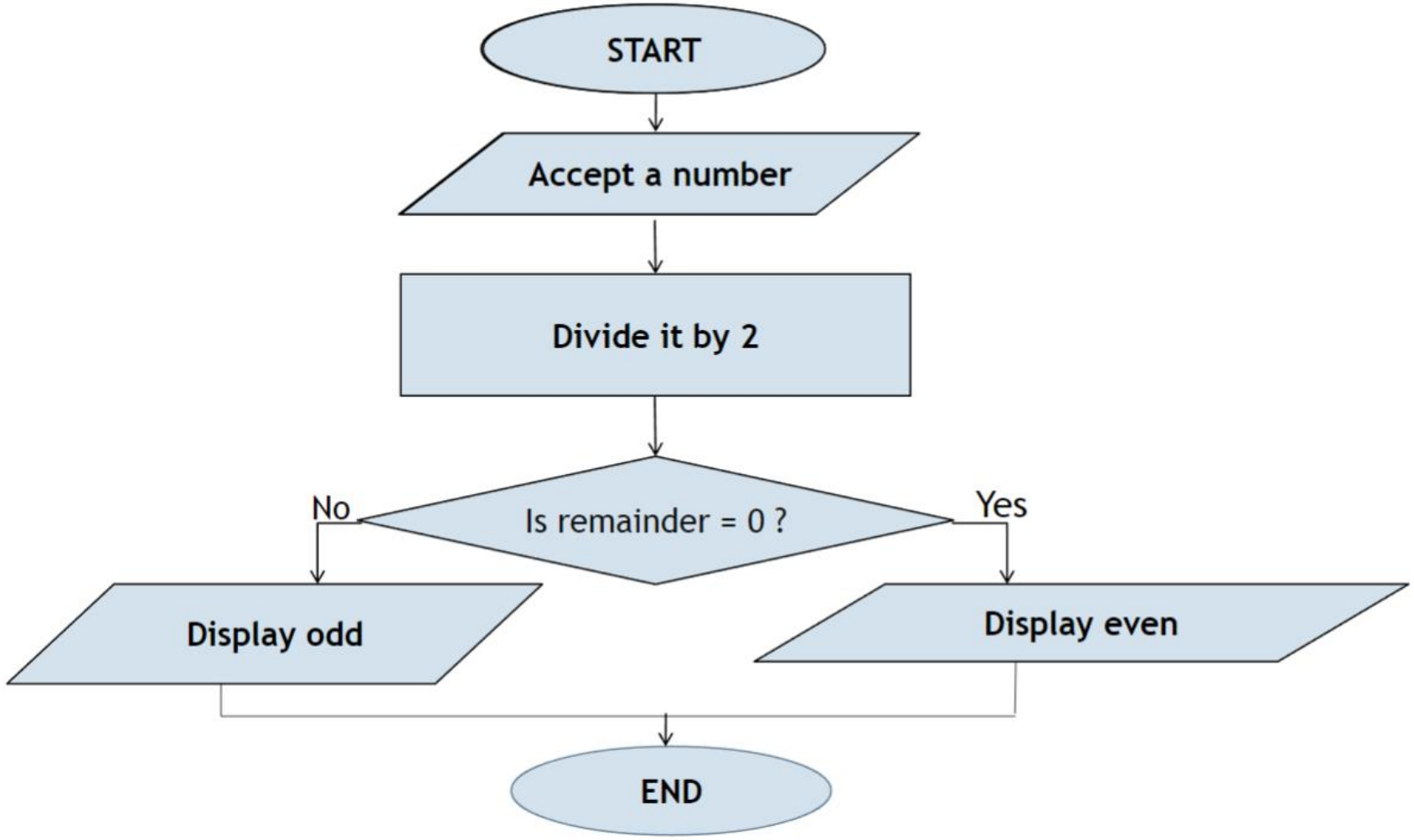
Symbol	Function	Description
	Start/End	Start and end point
	Input/output	Input and output operations
	Processing operation	Editing and calculation of data
	Decision	Check logical condition
	Connector	Indicates logical flow from one page to another
	Direction of logic	Direction of flow of logic
	Comment	Indicates any comments for explanatory notes

Example 1: Flowchart add sum of two numbers



Example 2:

Flowchart to check whether the input number is even or odd:



Using Algorithm and Flowchart in Coding

Algorithm

Step 1: Start

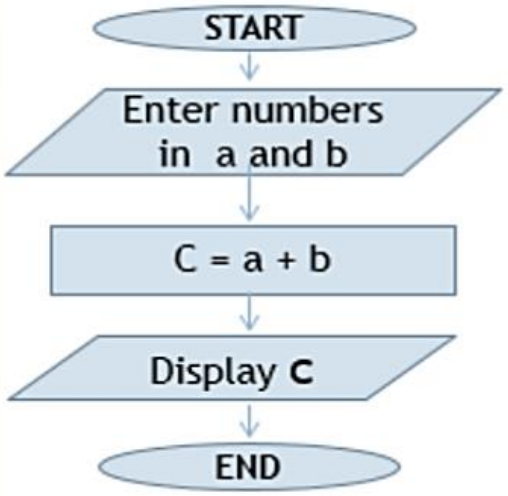
Step 2: Enter numbers in a and b

Step 3: Add a and b and store in c

Step 4: Display c

Step 5: End

Flowchart



Coding

```
#include<stdio.h>
#include<conio.h>
main()
{
    int a,b,c;
    printf("Enter numbers in a and b:  ");
    scanf("%d%d",&a,&b);
    c=a+b;
    printf("Displayed c: %d",c);
   getch();
}
```

```
Enter numbers in a and b: 5
6
Displayed c: 11
```

Exercises

Next Week …

- Fully understanding – Algorithms and Flowchart
  - Objective & Goals
  - Structure, and how to make it

- Install:
  - Code::Blocks (https://www.codeblocks.org/downloads/)

Thank You!

Questions?

sandi@ft.unp.ac.id