KRAKATAU posco

# Algorithm & Programming Fundamental

Muhammad Verly

26 July 2024

KRAKATAU posco

# Muhammad Verly

EIC Technology Team - P/C (L2)

## Work Experiences

### Software Engineer

tokopedia

DOKU

### AI Engineer/Data Scientist (Intern)

nodeflux

LIPI

## Background

Telkom University - Informatics

## Academic Experiences

### Teaching Assistant

- Artificial Intelligence (AI)
- Machine Learning (ML)
- Object Oriented Programming (OOP)

### Lab Work Assistant

- Object Oriented Programming (OOP)
- Software Analysis & Design

## Publication (ML)

**Custom Clustering And Load Balancing Implementation For Graph Database**

https://www.linkedin.com/in/verlymv/

# Today's Objectives

- To practice thinking algorithmically
- To understand and be able to implement proper program development
- To start learning about control structures
- To be able to express an algorithm using a flow chart

# What is an Algorithm?

- Steps used to solve a problem

- Problem must be
  - Well defined
  - Fully understood by the programmer

- Steps must be
  - Ordered
  - Unambiguous
  - Complete

KRAKATAU posco

# Developing an Algorithm

KRAKATAU POSCO

# Program Development

1. Understand the problem
2. Represent your solution (your algorithm)
   - Pseudocode
   - Flowchart
3. Implement the algorithm in a program
4. Test and debug your program

KRAKATAU POSCO

# Step 1: Understanding the Problem

- Input
  - What information or data are you given?
- Process
  - What must you do with the information/data?
  - **This is your algorithm!**
- Output
  - What are your deliverables?

KRAKATAU POSCO

# "Weekly Pay" Example

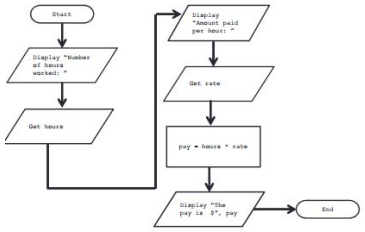
- Create a program to calculate the weekly pay of an hourly employee
  - What is the input, process, and output?

- Input: pay rate and number of hours
- Process: multiply pay rate by number of hours
- Output: weekly pay

KRAKATAU POSCO

# Step 2: Represent the Algorithm

- Can be done with flowchart or pseudocode
- Flowchart




  - Symbols convey different types of actions
- Pseudocode
  - A cross between code and plain English
- One may be easier for you – use that one

KRAKATAU POSCO

# Step 2A: Pseudocode

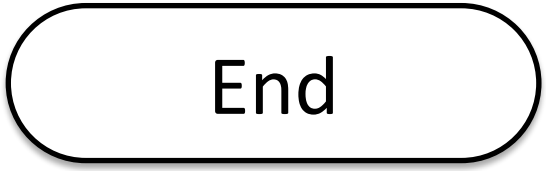
- Start with a plain English description, then…

1. `Display "Number of hours worked: "`
2. `Get the hours`
3. `Display "Amount paid per hour: "`
4. `Get the rate`
5. `Compute pay = hours * rate`
6. `Display "The pay is $" , pay`

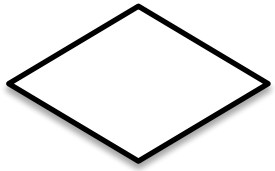
# Flowchart Symbols


Start Symbol

Input/Output




End Symbol

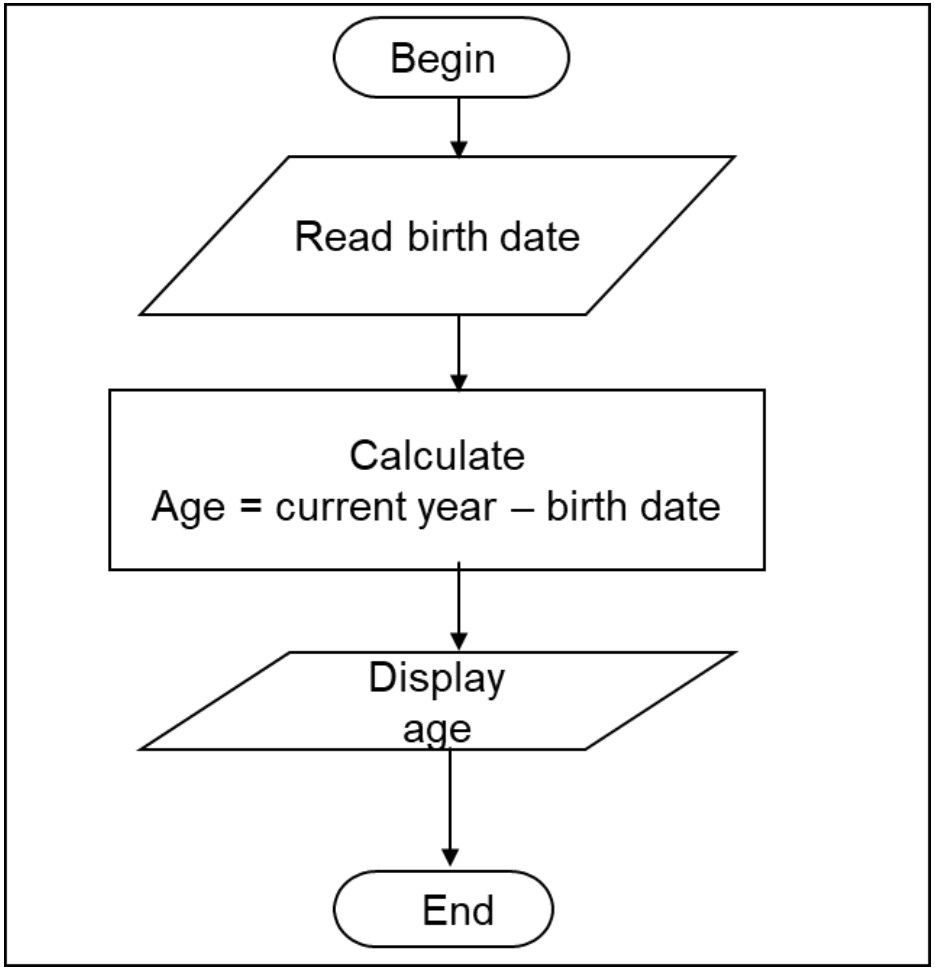


Decision Symbol

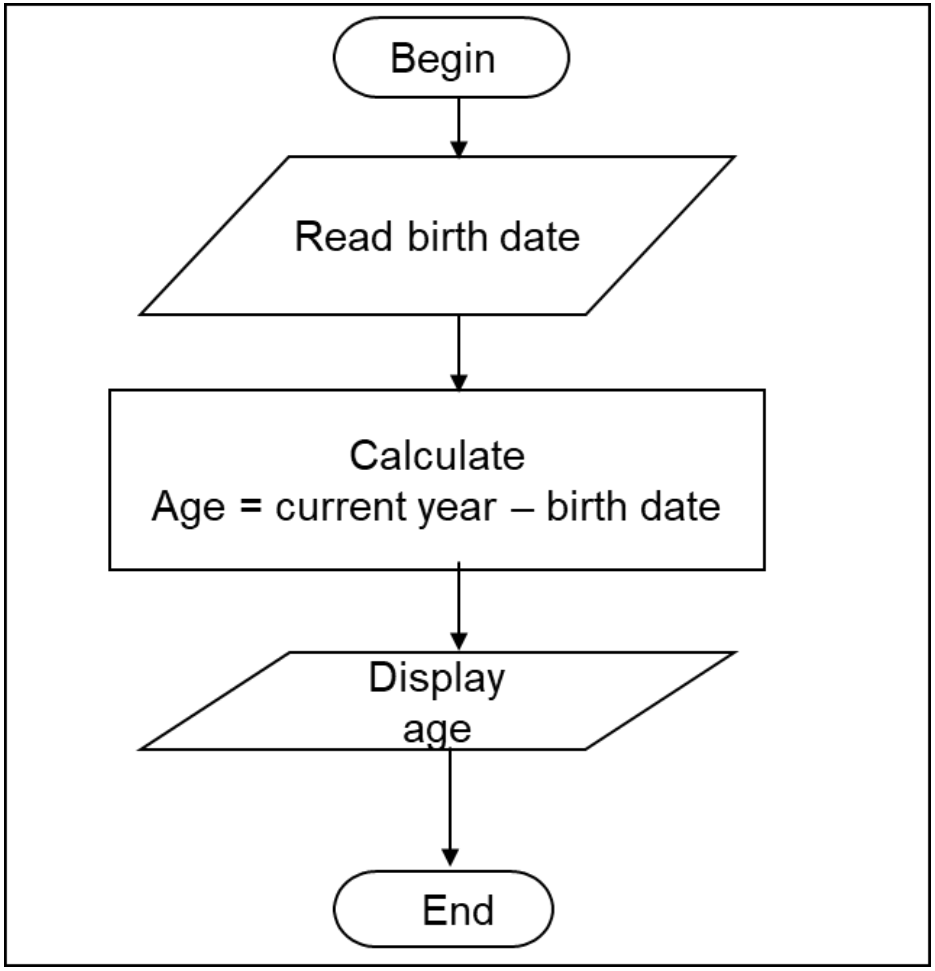
Data Processing Symbol

Flow Control Arrows

# Flowchart - Example 1

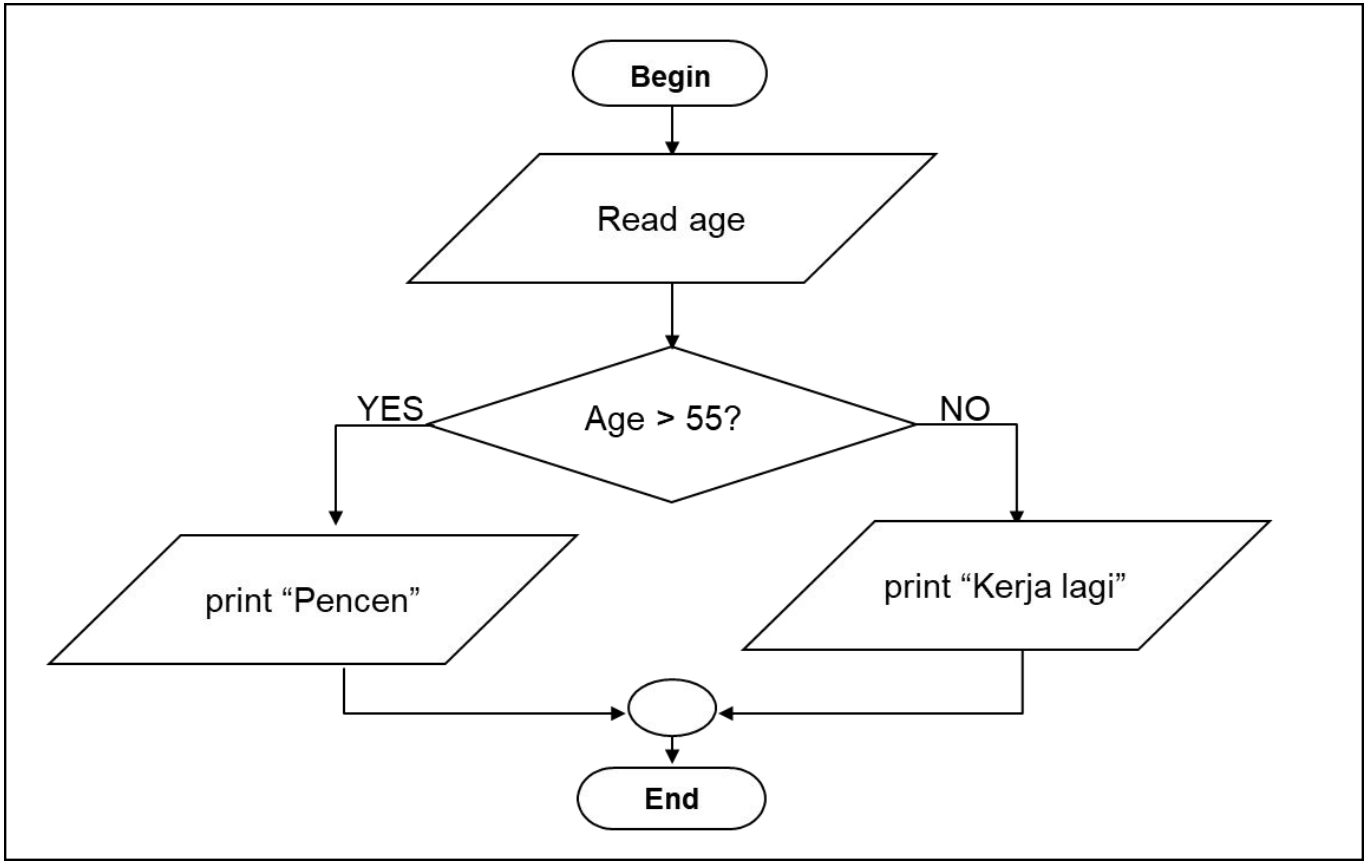


# Flowchart - Example 1



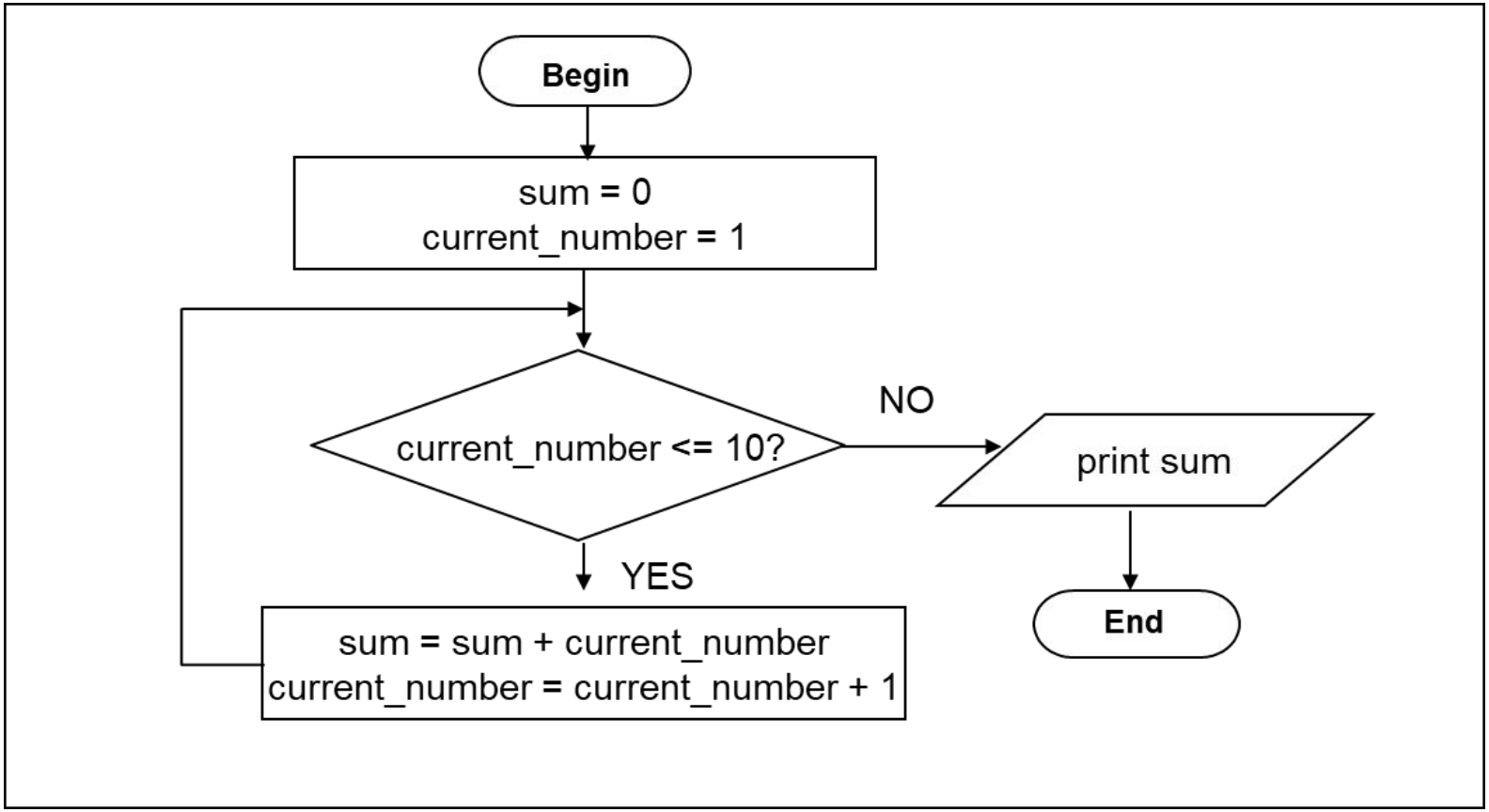
KRAKATAU POSCO

# Flowchart - Example 2

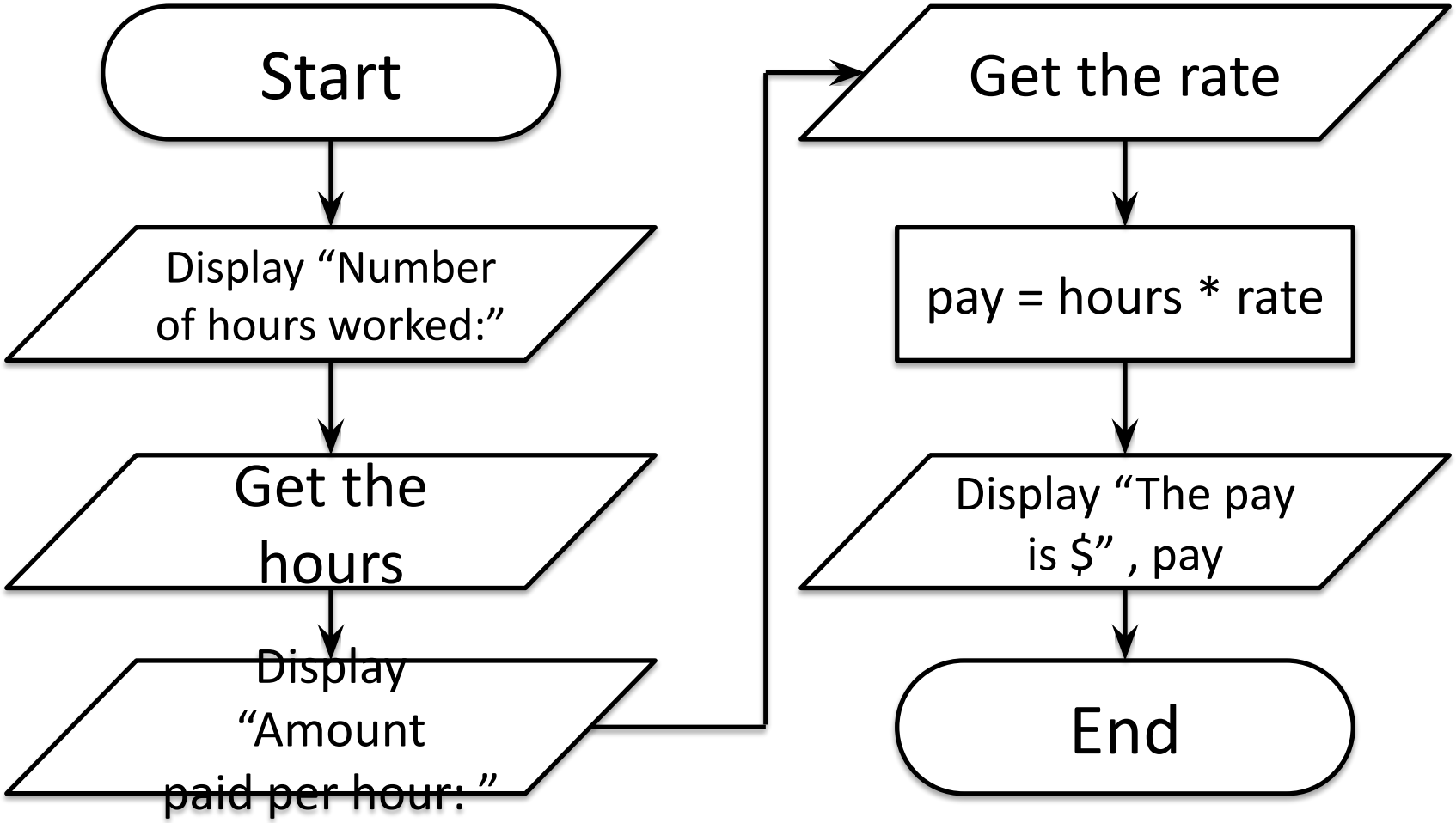


KRAKATAU POSCO

# Flowchart - Example 3



# Step 2B: Flowchart



KRAKATAU POSCO

# Steps 3 and 4: Implementation and Testing/Debugging

- We'll cover implementation in detail next class

- Testing and debugging your program involves identifying errors and fixing them
  - We'll talk about this later today

# More example…

**Example 1:** Write an algorithm to determine a student's final grade and indicate whether it is passing or failing. The final grade is calculated as the average of four marks.

KRAKATAU POSCO

# Pseudocode & Algorithm

**Pseudocode**:

*Input a set of 4 marks*

*Calculate their average by summing and dividing by 4*

*if average is below 50*

*Print "FAIL"*

*else*

*Print "PASS"*

# Pseudocode & Algorithm

**Detailed Algorithm**

Step 1: Input M1,M2,M3,M4

Step 2: GRADE ¬ (M1+M2+M3+M4)/4

Step 3: if (GRADE < 50) then

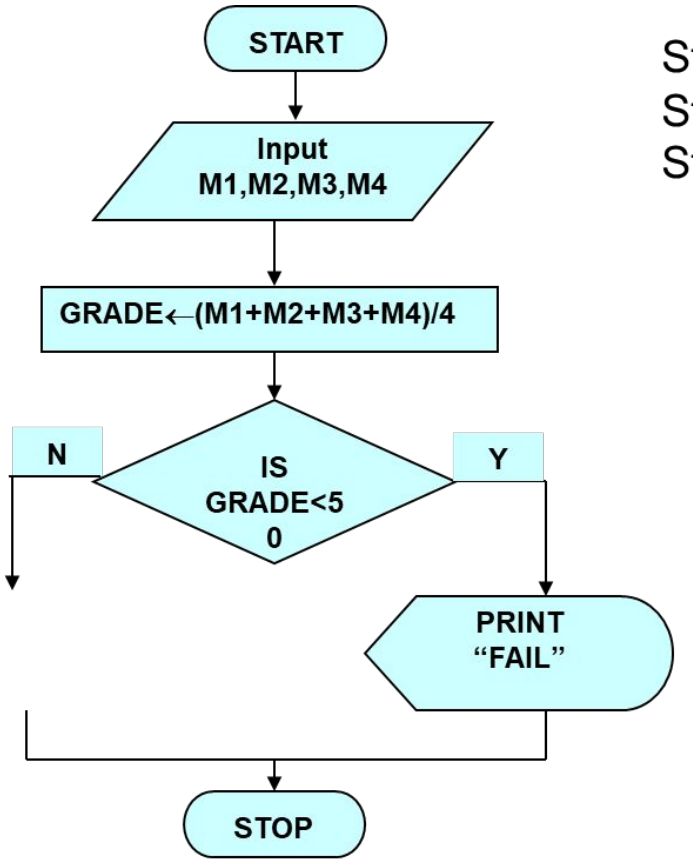
Print “FAIL”

else

Print “PASS”

endif

# Pseudocode & Algorithm




Step 1: Input M1,M2,M3,M4
Step 2: GRADE ← (M1+M2+M3+M4)/4
Step 3: if (GRADE <50) then
Print “FAIL”
else
Print “PASS”
endif

# Example 2

Write an algorithm and draw a flowchart to convert the length in feet to centimeter.

**Pseudocode**:

*Input the length in feet (Lft)*

*Calculate the length in cm (Lcm) by multiplying LFT with 30*

*Print length in cm (LCM)*

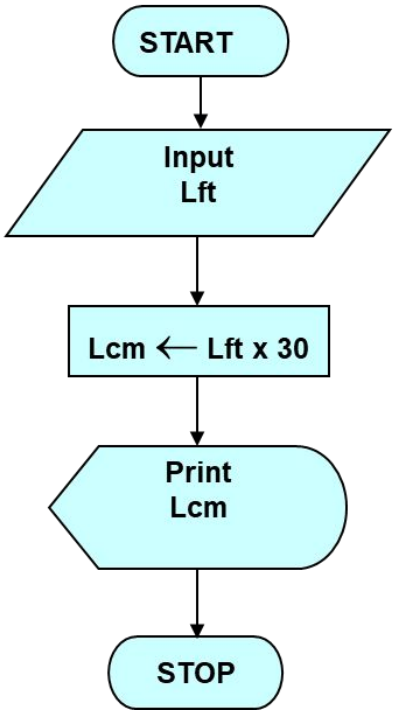
KRAKATAU POSCO

# Example 2

**Algorithm**

- Step 1: Input Lft
- Step 2: Lcm ← Lft x 30
- Step 3: Print Lcm

**Flowchart**



# Example 3

**Write an algorithm and draw a flowchart that will read the two sides of a rectangle and calculate its area.**

**Pseudocode**

*Input the width (W) and Length (L) of a rectangle*

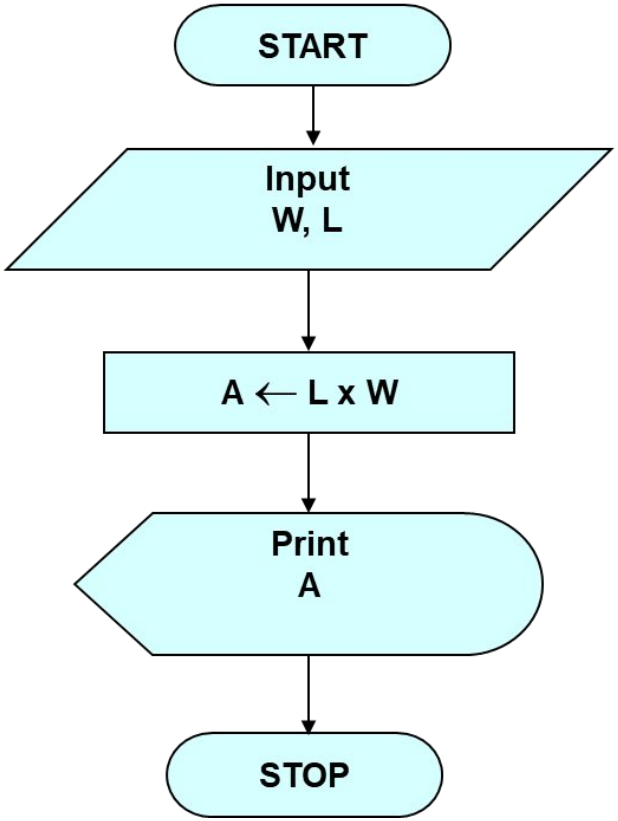
*Calculate the area (A) by multiplying L with W*

*Print A*

# Example 3

**Algorithm**

- Step 1: Input W,L
- Step 2: A ← L x W
- Step 3: Print A



# Algorithms and Language

- Notice that developing the algorithm didn't involve any Python at all
  - Only pseudocode or a flowchart was needed
  - An algorithm can be coded in any language
- All languages have 3 important control structures we can use in our algorithms

KRAKATAU posco

# Programming Principles

# Programming Principles

- Structures that control how the program "flows" or operates, and in what order

- Sequence
- Decision Making
- Looping

# Sequence

- One step after another, with no branches
- Already wrote one for “Weekly Pay” problem
- What are some real life examples?
  - Dialing a phone number
  - Purchasing and paying for groceries

# Decision Making

- Selecting one choice from many based on a specific reason or condition
  - If something is true, do ***A*** … if it's not, do ***B***

- What are some real life examples?
  - Walking around campus (construction!)
  - Choosing where to eat lunch

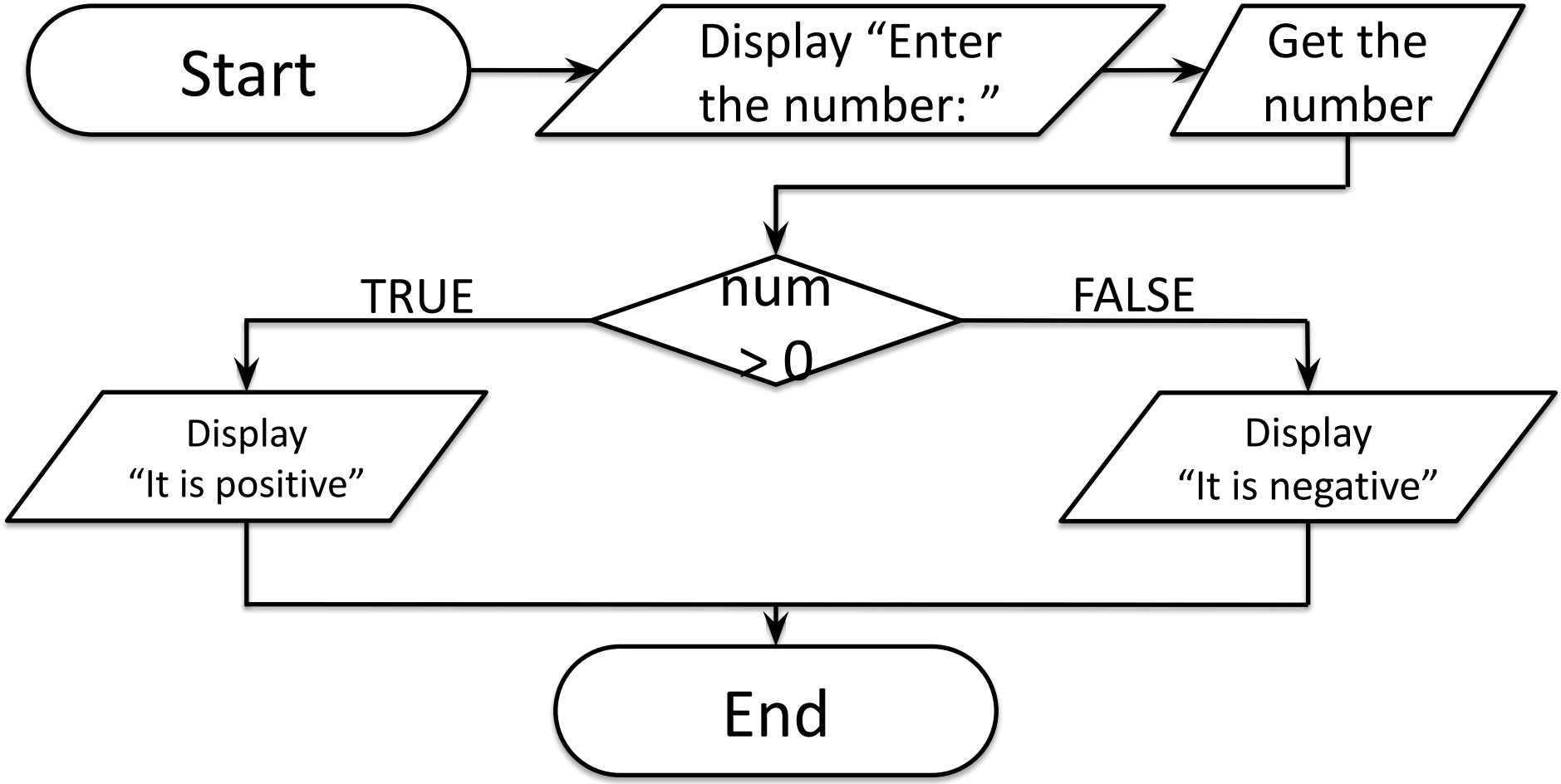
KRAKATAU POSCO

# Decision Making: Pseudocode

- Answer the question “Is a number positive?”
  - Start with a plain English description

```
1.  Display “Enter the number: ”
2.  Get the number (call it num)
3.  If num > 0
4.      Display “It is positive”
5.  Else
6.      Display “It is negative”
```

# Decision Making: Flowchart



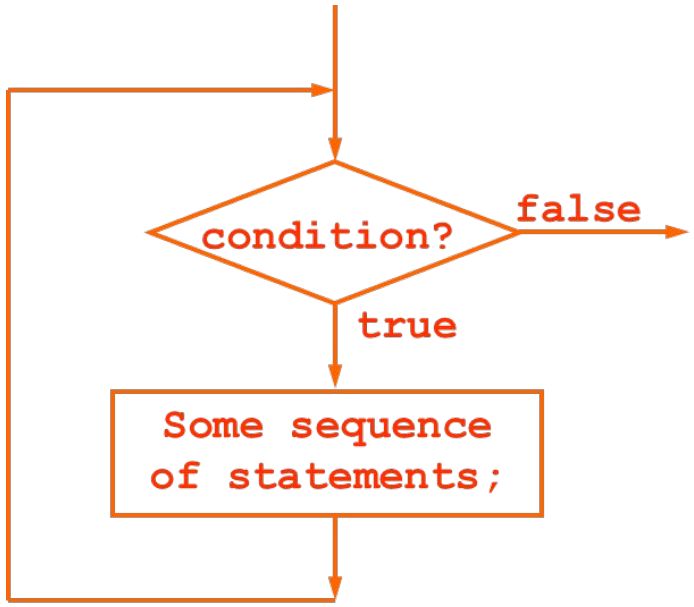
# Looping

- Doing something over and over again
- Used in combination with decision making
  - Otherwise we loop forever
    - This is called an “infinite loop”
- What are some real life examples?
  - Jogging, Assignment grading,
  - Walking up steps

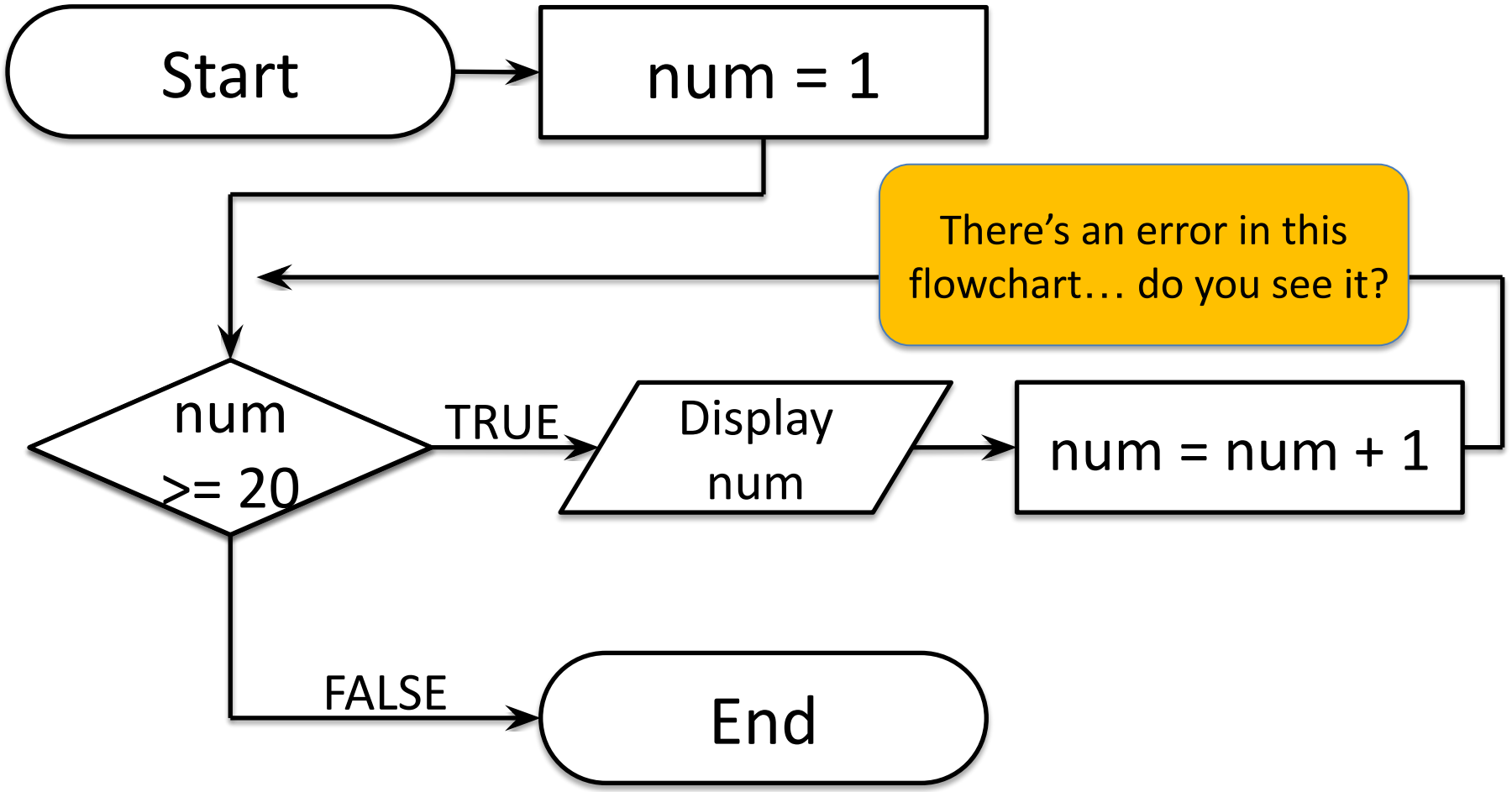
# Looping: Pseudocode

- Write an algorithm that counts from 1-20
  - Start with a plain English description

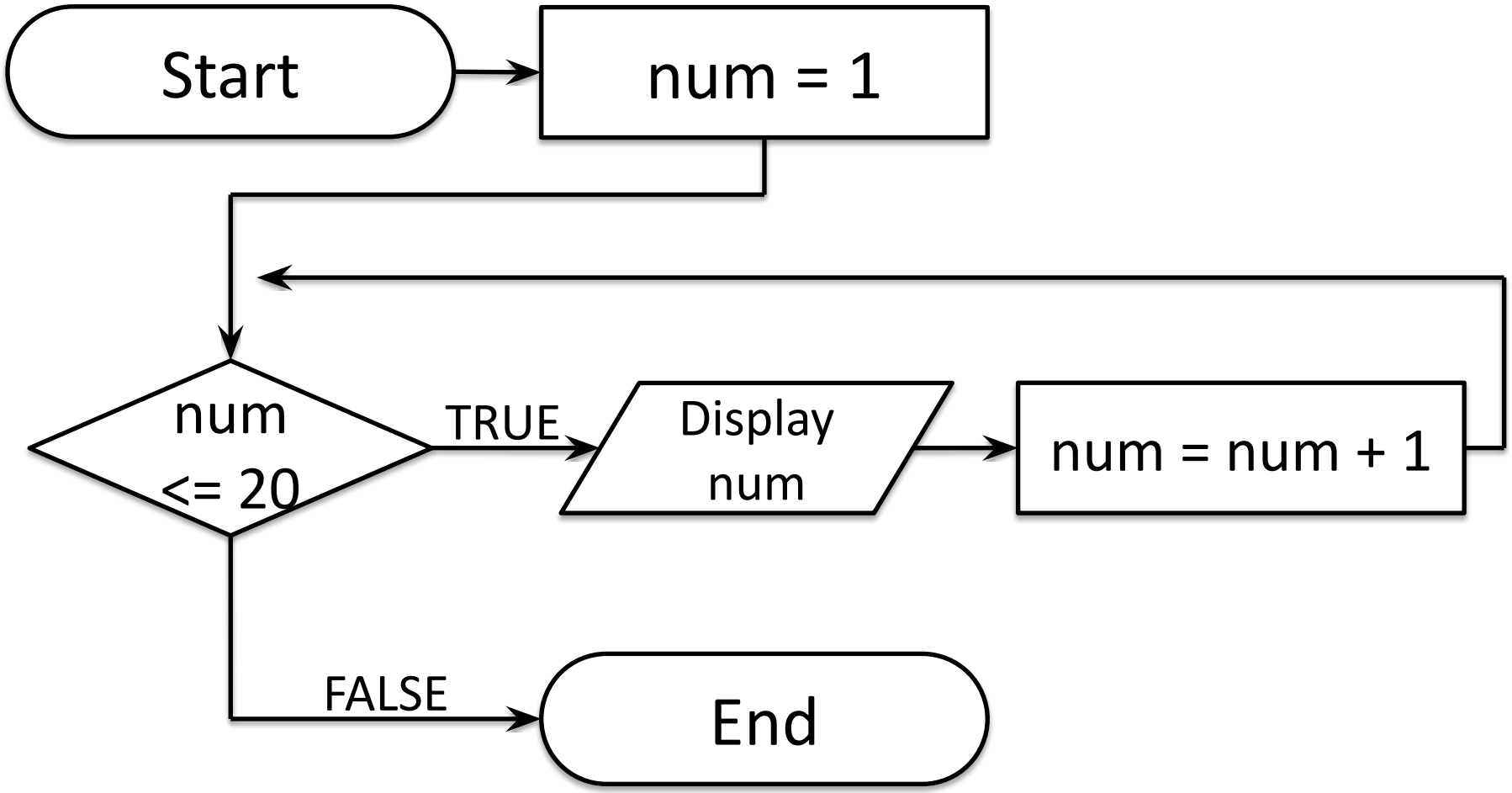
```
1.  Set num = 1
2.  While num <= 20
3.       Display num
4.       num = num + 1
5.  (End loop)
```



# Looping: Flowchart



# Looping: Flowchart



KRAKATAU posco

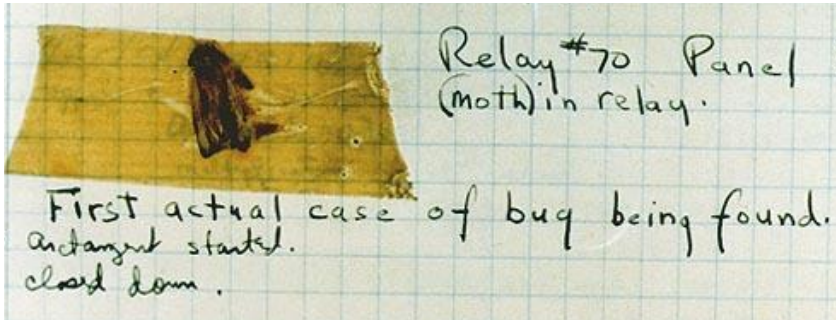
# Debugging

KRAKATAU POSCO

# A Bit of History on "Bugs"

- US Navy lab – September 9, 1947
- Grace Hopper and her colleagues were working on the Harvard Mark II
  - Or trying to… it wasn't working right

- They found a literal bug inside the machine
  - Taped the bug (a moth) into their log book



# Errors (“Bugs”)

- Two main classifications of errors

- Syntax errors
  - Prevent Python from understanding what to do
- Logical errors
  - Cause the program to run incorrectly, or to not do what you want

# Syntax Errors

- "Syntax" is the set of rules followed by a computer programming language
  - Similar to grammar and spelling in English
- Examples of Python's syntax rules:
  - Keywords must be spelled correctly

    **`True`** and **`False`**, not **`Ture`** or **`Flase`** or **`Truu`**
  - Quotes and parentheses must be closed:

```
("Open and close")
```

KRAKATAU POSCO

# Syntax Error Examples

- Find the syntax errors in each line of code below:

```
1    prnit("Hello")
2    print("What"s up?")
3    print("Aloha!)
4    print("Good Monring")
```

KRAKATAU POSCO

# Syntax Error Examples

- Find the syntax errors in each line of code below:

```
1    prnit("Hello")
2    print("What"s up?")
3    print("Aloha!)
4    print("Good Monring")
```

not actually a syntax error

# Logical Errors

- Logical errors don't bother Python at all… they only bother you!

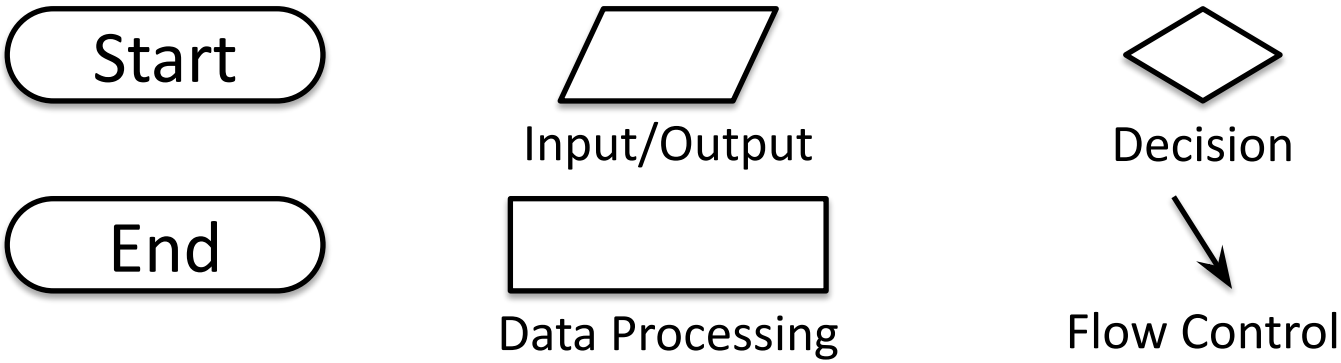
- Examples of logical errors:
  - Using the wrong value for something

```
currentYear = 2013
```

  - Doing steps in the wrong order
    - "Close jelly jar.  Put jelly on bread.  Open jelly jar."

# Exercise

- Write an algorithm that asks a user for their name, then responds with “Hello NAME”
- You can use a flowchart or pseudocode



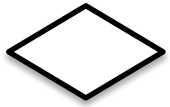
KRAKATAU POSCO

# Exercise #2

- Write an algorithm that asks a user for their grade, and tells them their letter grade.

A: 100 - 90
B: <90 - 80
C: <80 - 70
D: <70 - 60
F: <60 - 0


Input/Output



Decision


Data Processing

Flow Control