# 19CSE100 Problem Solving and Algorithmic Thinking

## Sequence , Selection and Looping

# Algorithmic Thinking

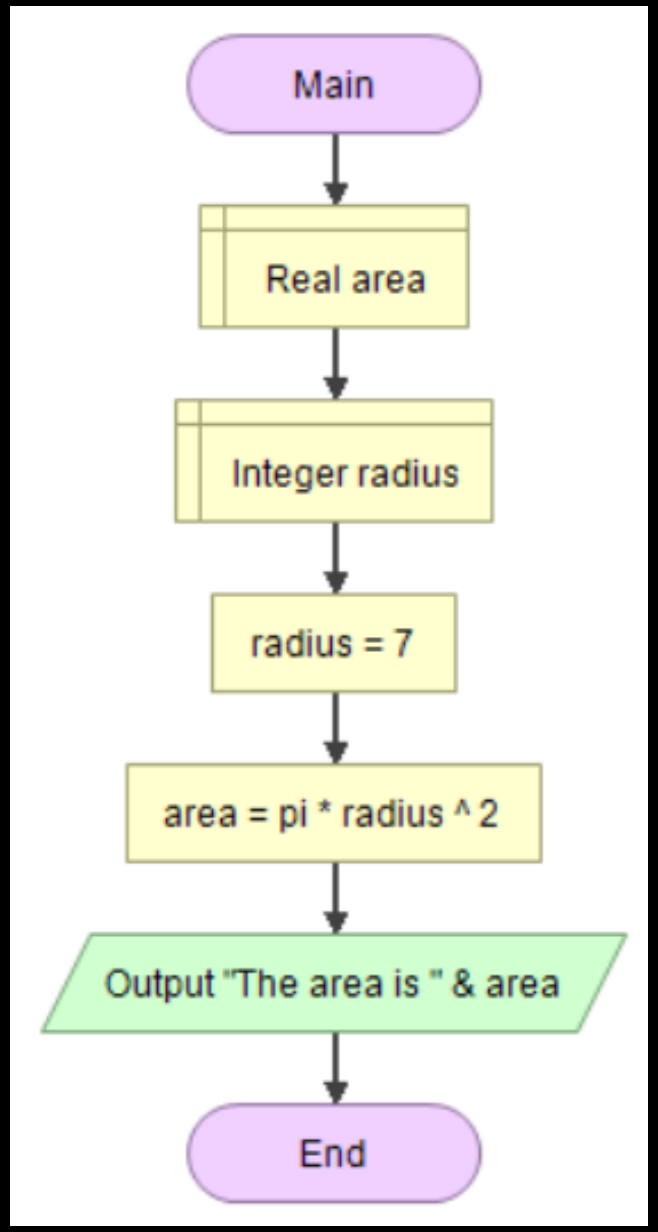
Designing Algorithms
*is*
problem solving

# Why Algorithms?

Algorithms express solutions to solvable problems

# SEQUENCE

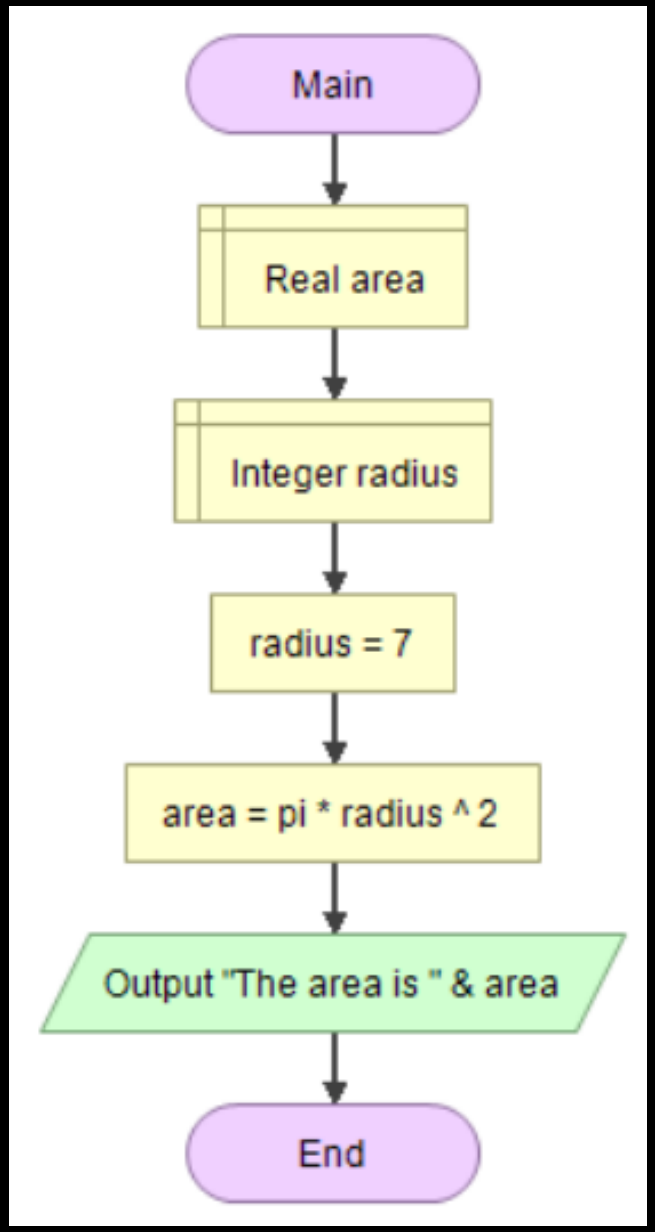
# A Simple Sequence Algorithm

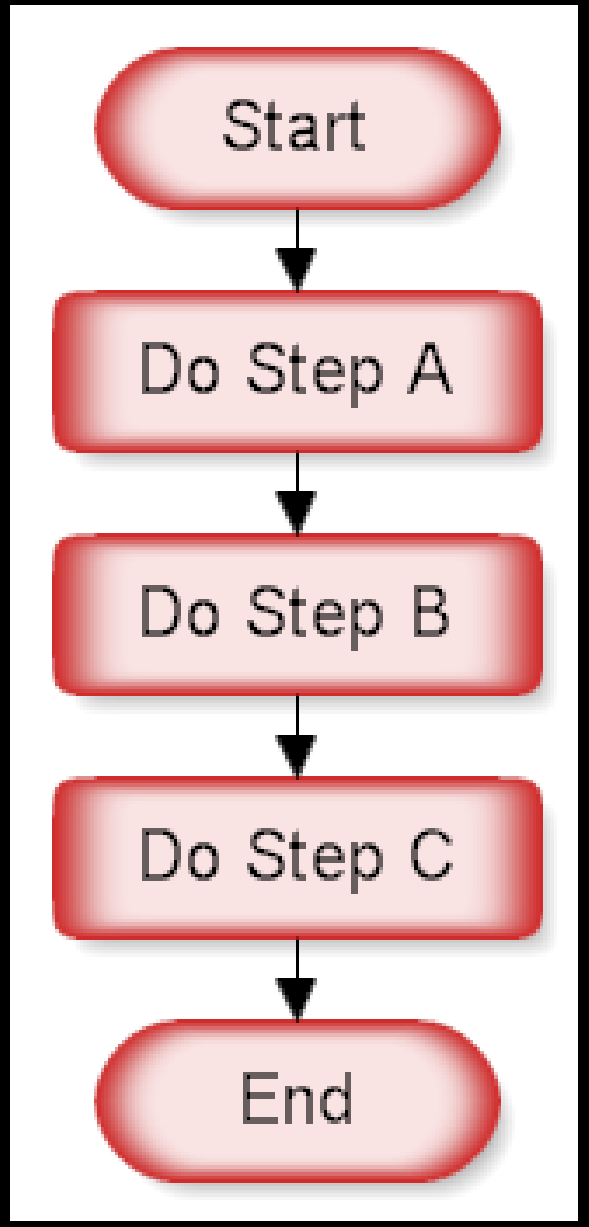



Observe the flow of the algorithm

http://flowgorithm.org/documentation/assignment.htm

# Flow of Algorithm



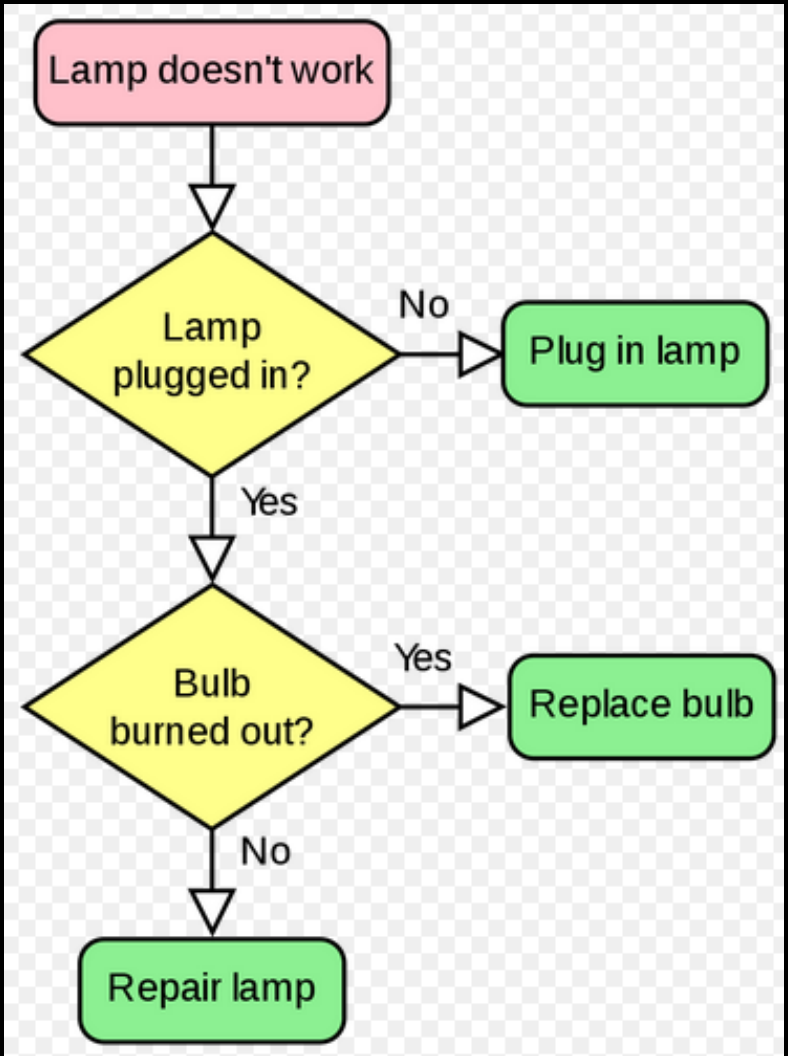




flow of the algorithm is sequential

https://www.rff.com/structured_flowchart.php

# SELECTION

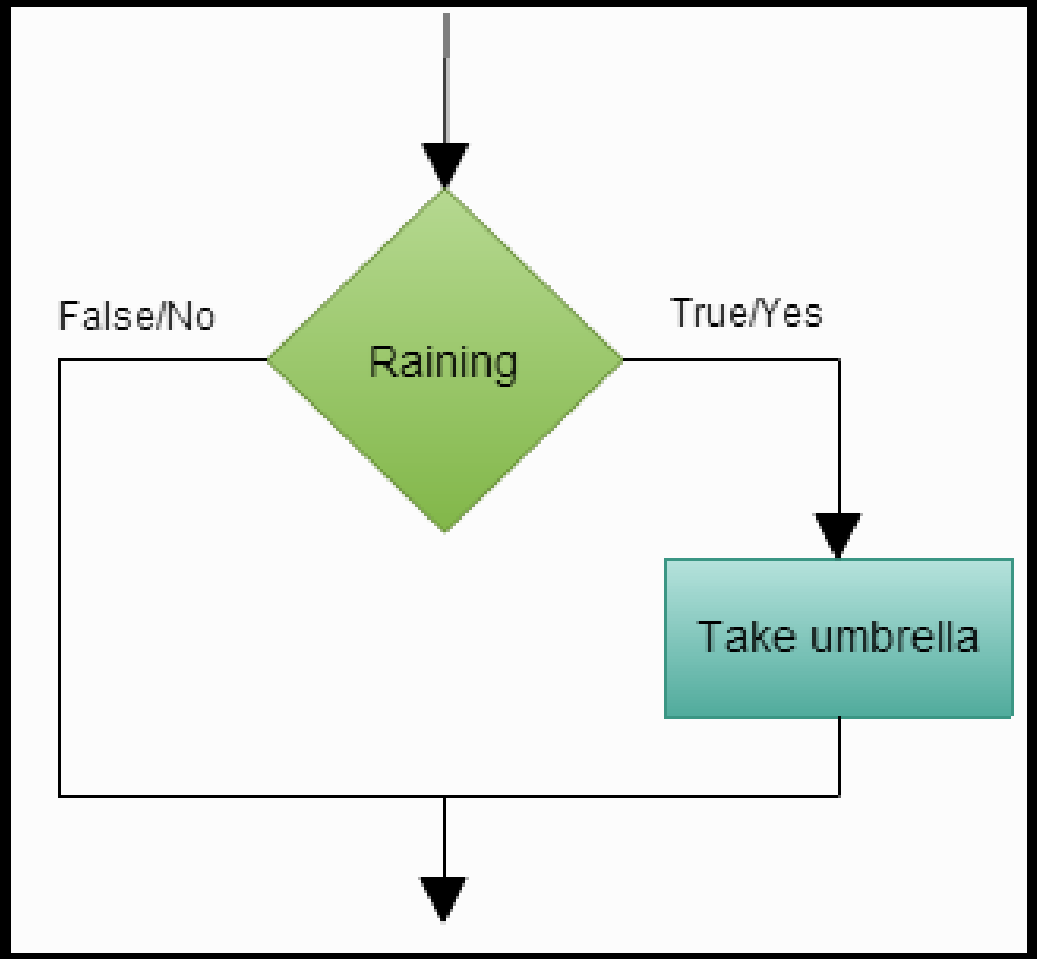
# Representing Selection Algorithms




1. Lamp doesn't work
2. If lamp not plugged in
   Then plug in lamp
   Else if the bulb burned out
   Then replace bulb
   Else repair lamp

https://en.wikipedia.org/wiki/Flowchart#/media/File:LampFlowchart.svg

# One-way Selection



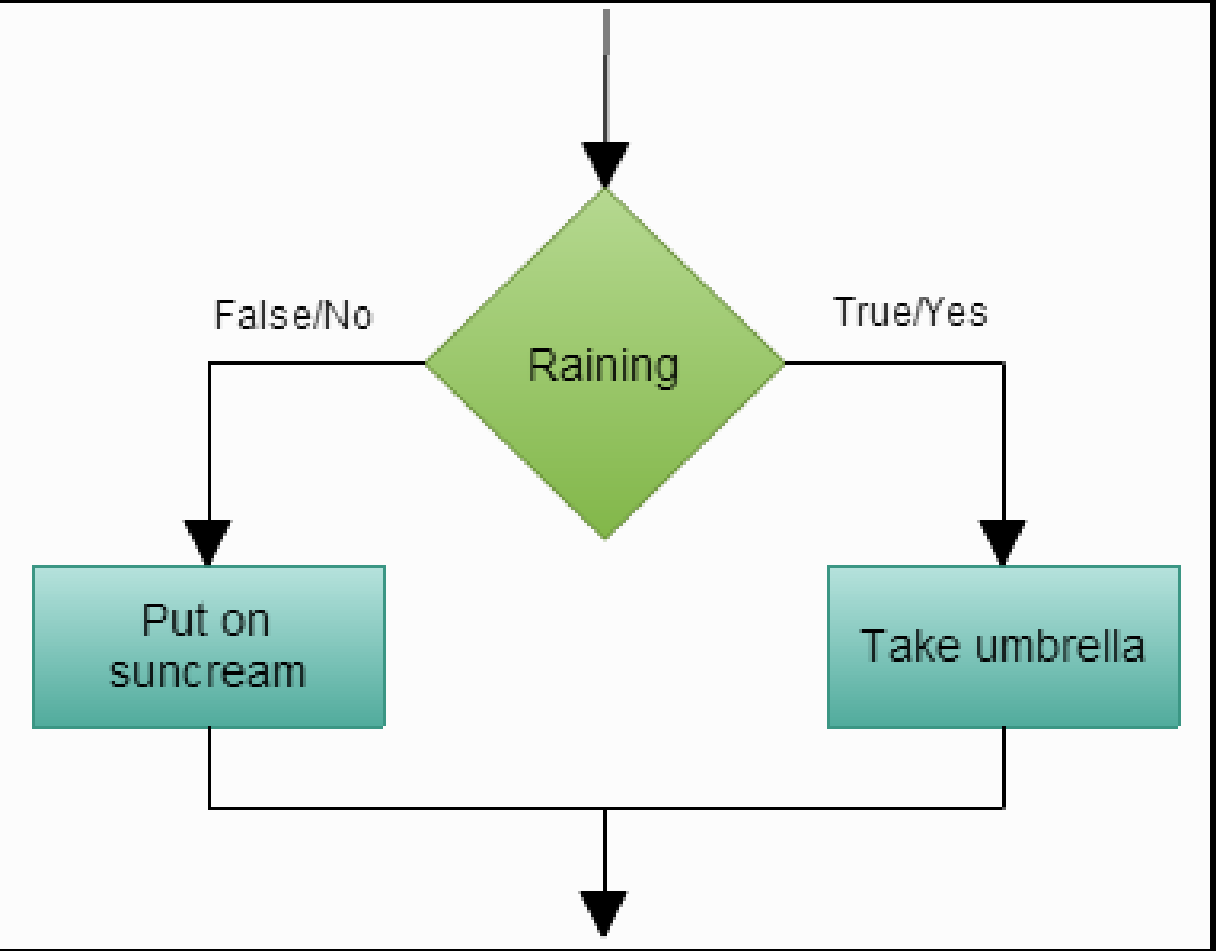

**IF** (*condition*) **THEN**

*statements*

**END IF**

https://courses.p2pu.org/en/courses/2544/content/5290/

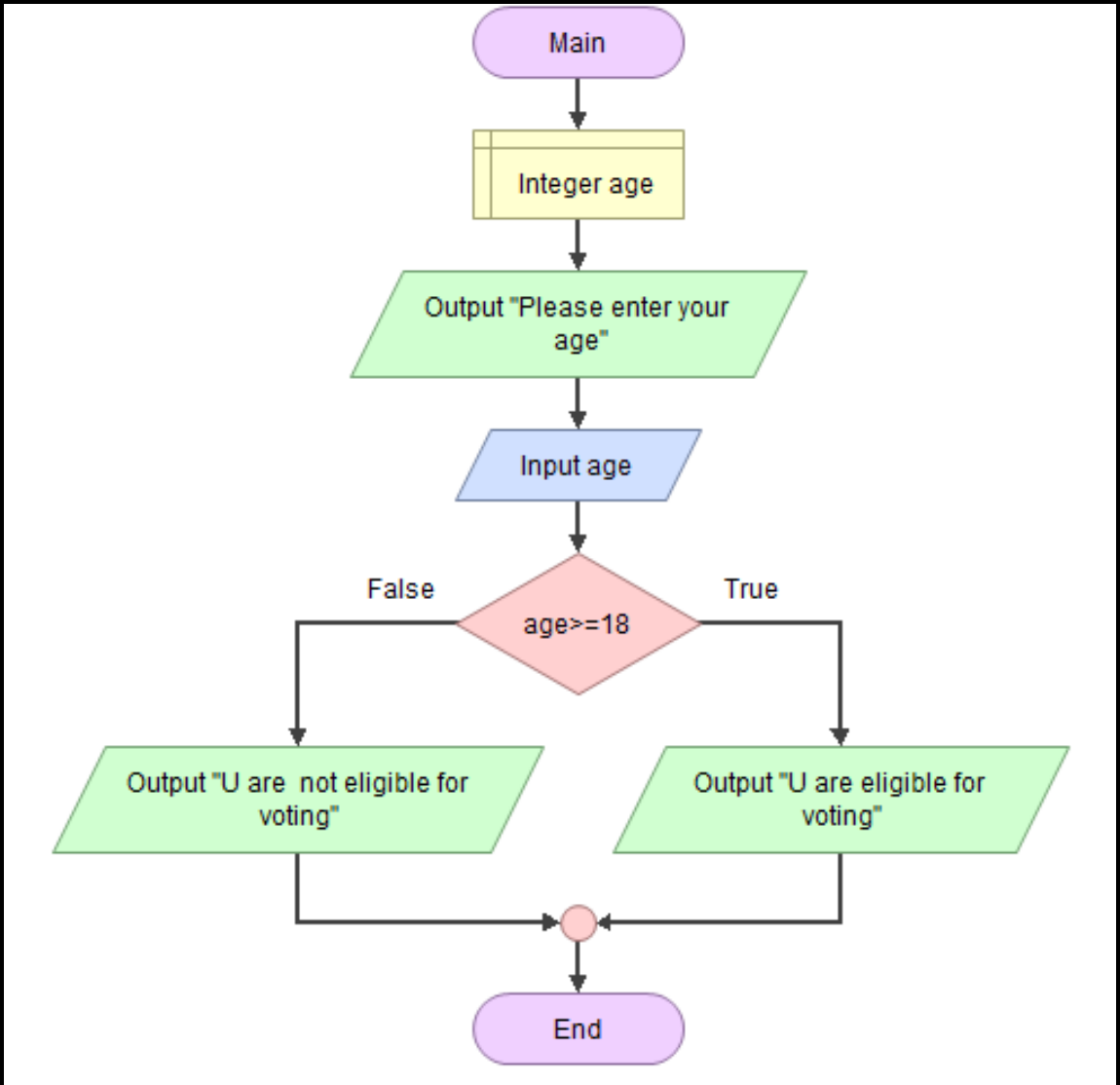
# Two-way Selection




```
IF (condition) THEN
        statements1
ELSE
        statements2
END IF
```

https://courses.p2pu.org/en/courses/2544/content/5290/

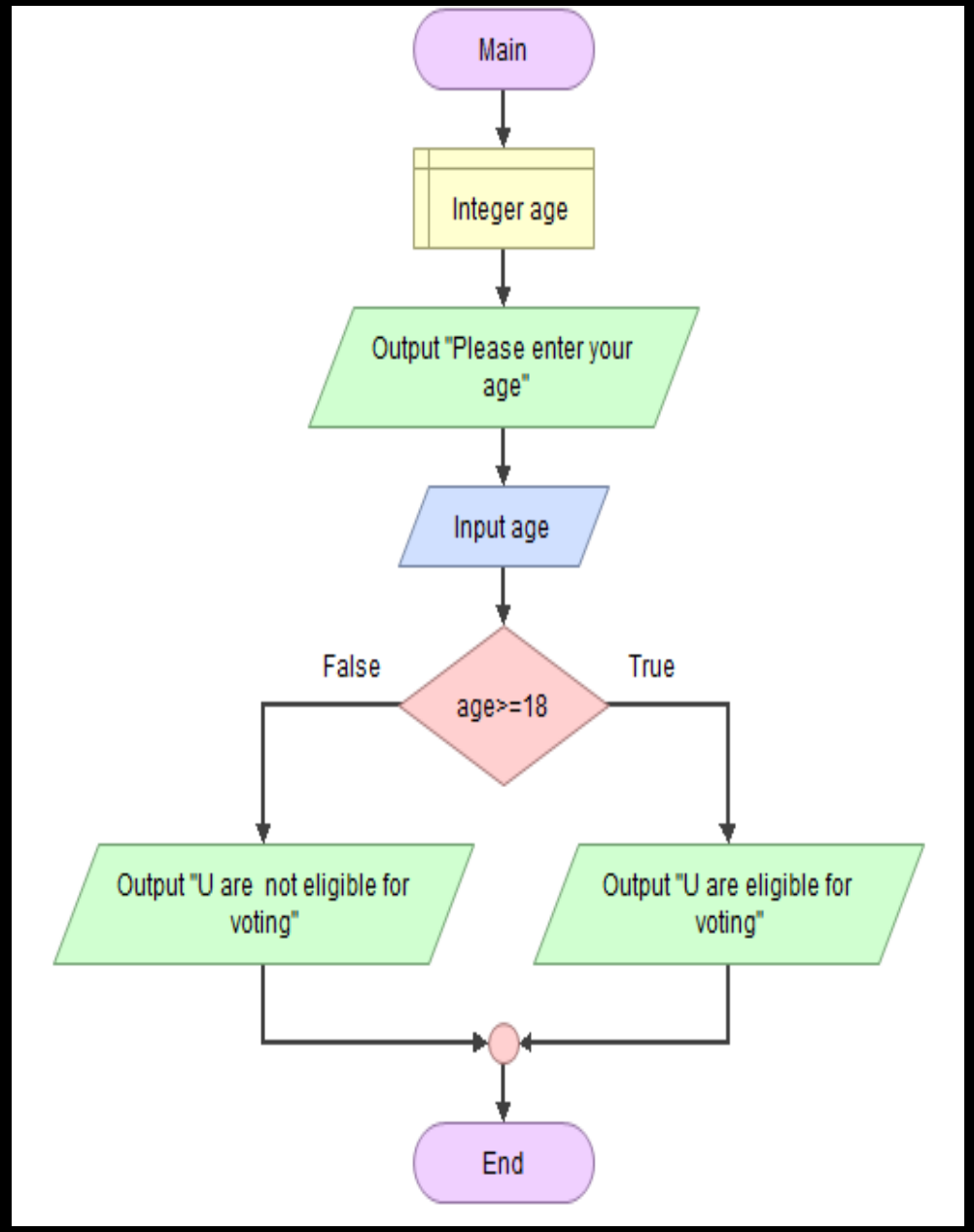
# A Simple Algorithm

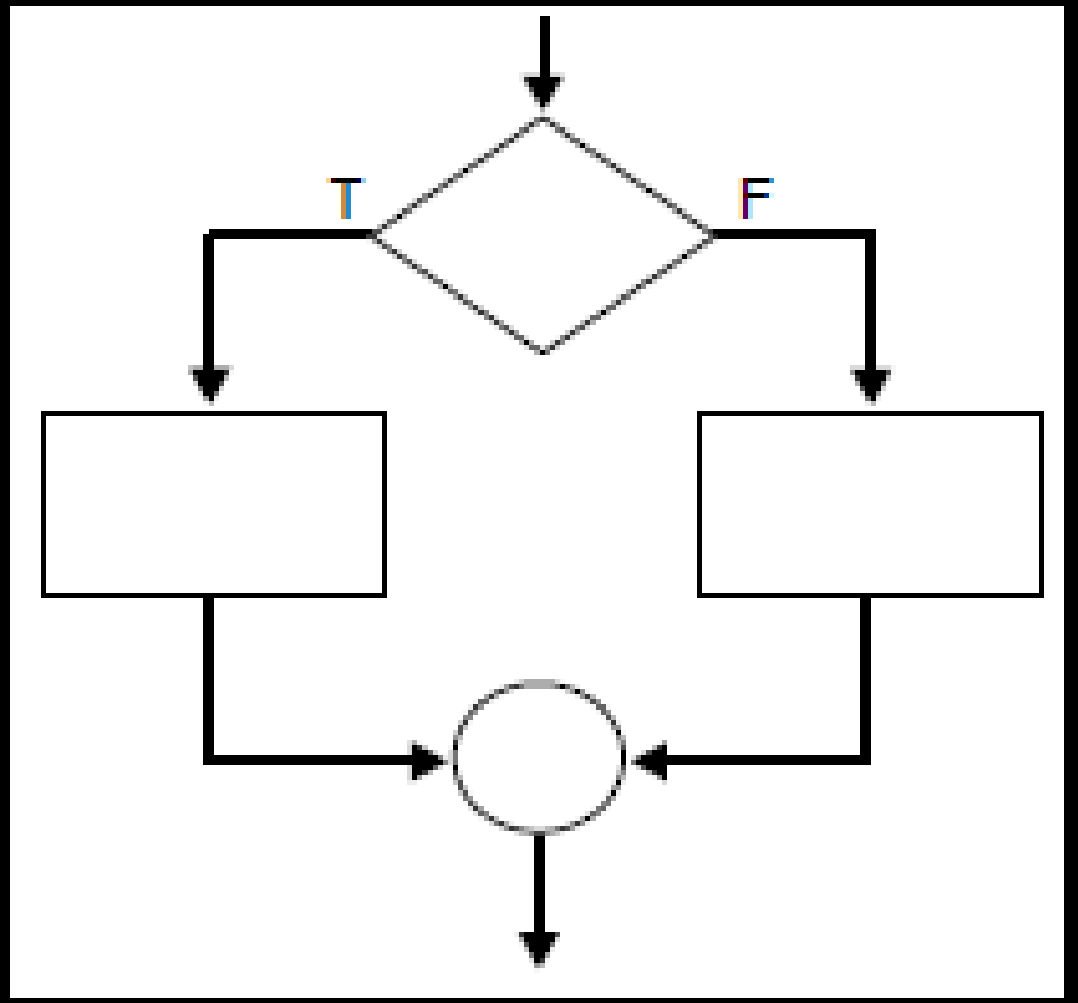



Observe the flow of the algorithm

http://flowgorithm.org/documentation/if.htm

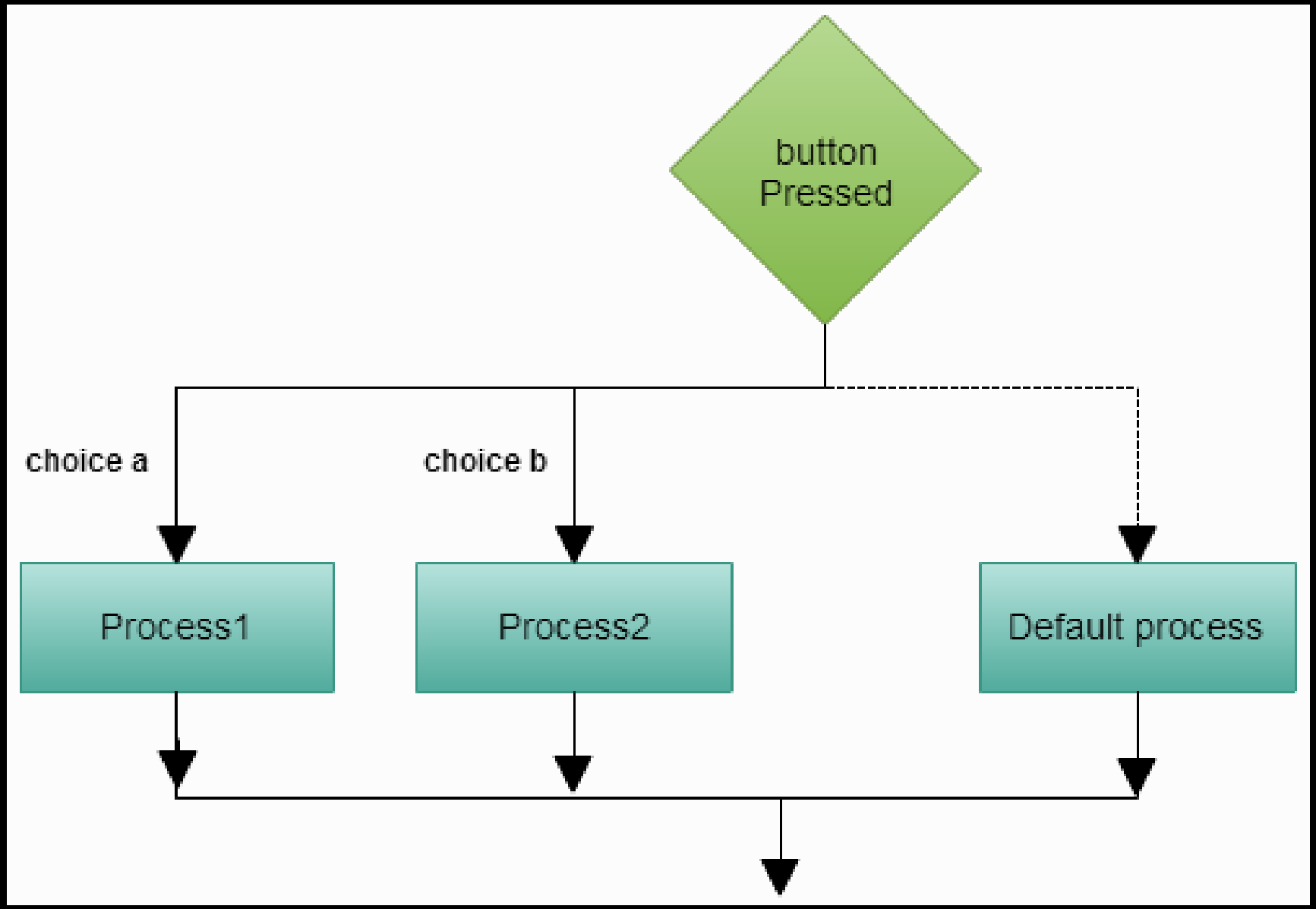
# Selection



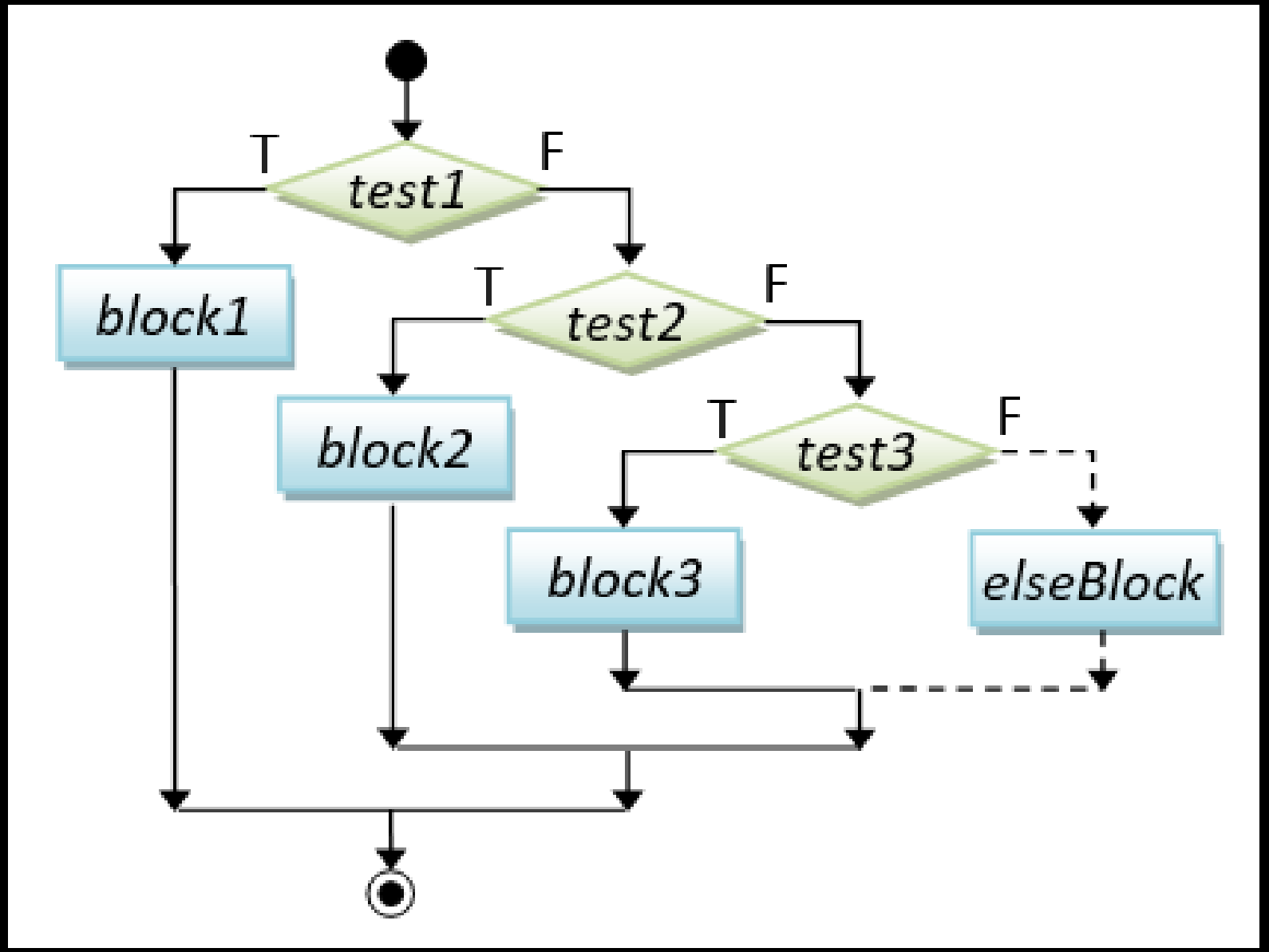

https://cdn.intellipaat.com/wp-content/uploads/2015/09/control-structures.png

# Multi-way Selection




https://courses.p2pu.org/en/courses/2544/content/5290/

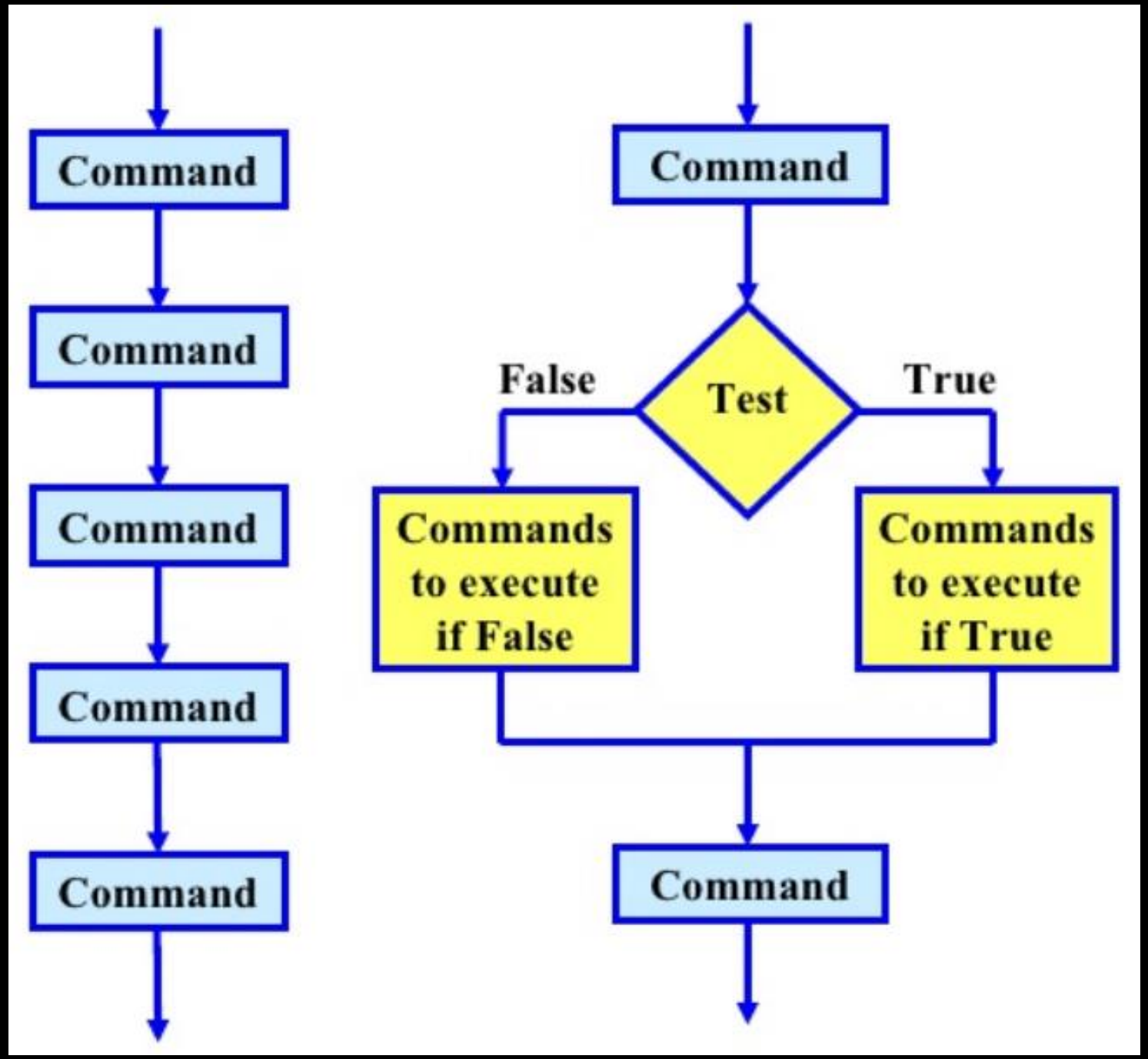
# Multi-way Selection




https://www.ntu.edu.sg/home/ehchua/programming/java/J2_Basics.html

# Multi-way Selection

```
IF (condition1) THEN
    Statements1
ELSE  IF (condition2) THEN
            Statements2
        ELSE   IF (Condition3) THEN
                    Statements3
                ELSE IF (Condition4) THEN
                            Statements4
                END IF
        END IF
 END IF
```

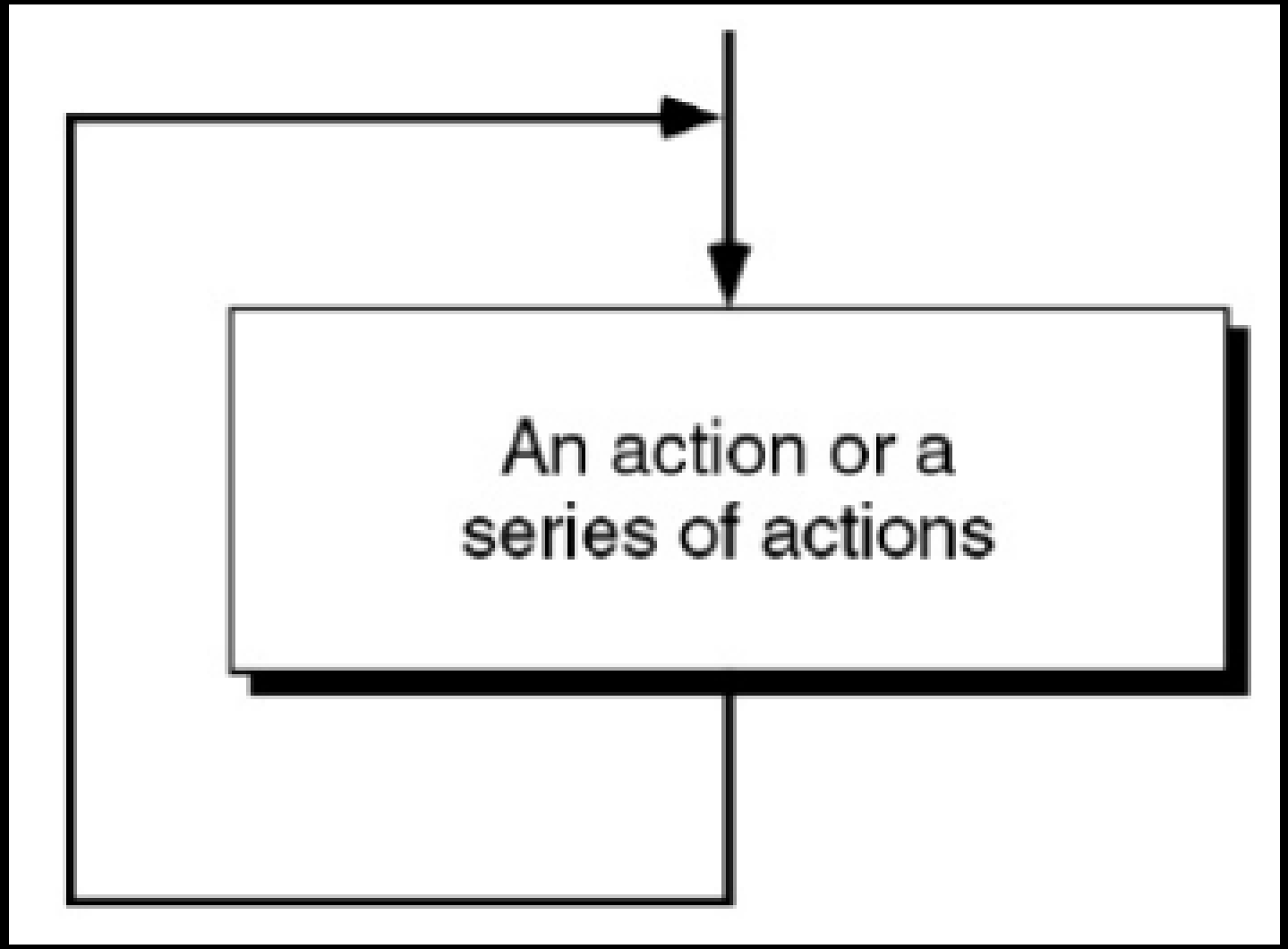



# Sequence and Selection

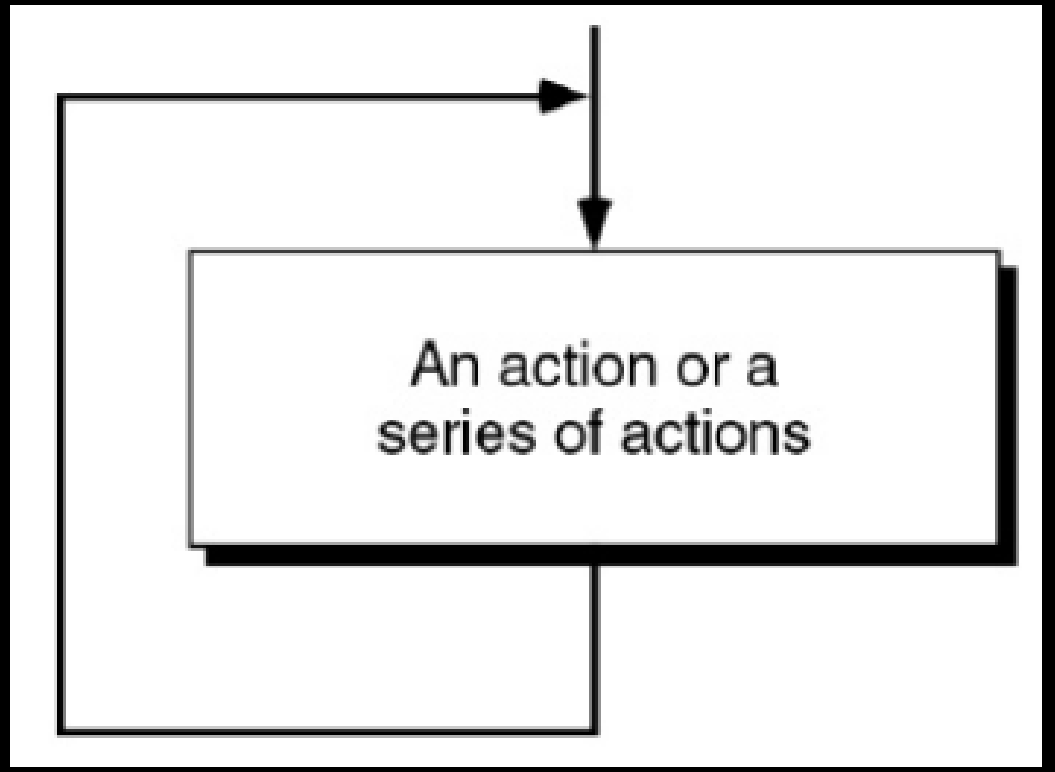
https://www.slideshare.net/ldunphy/iteration

# REPETITION or LOOPING

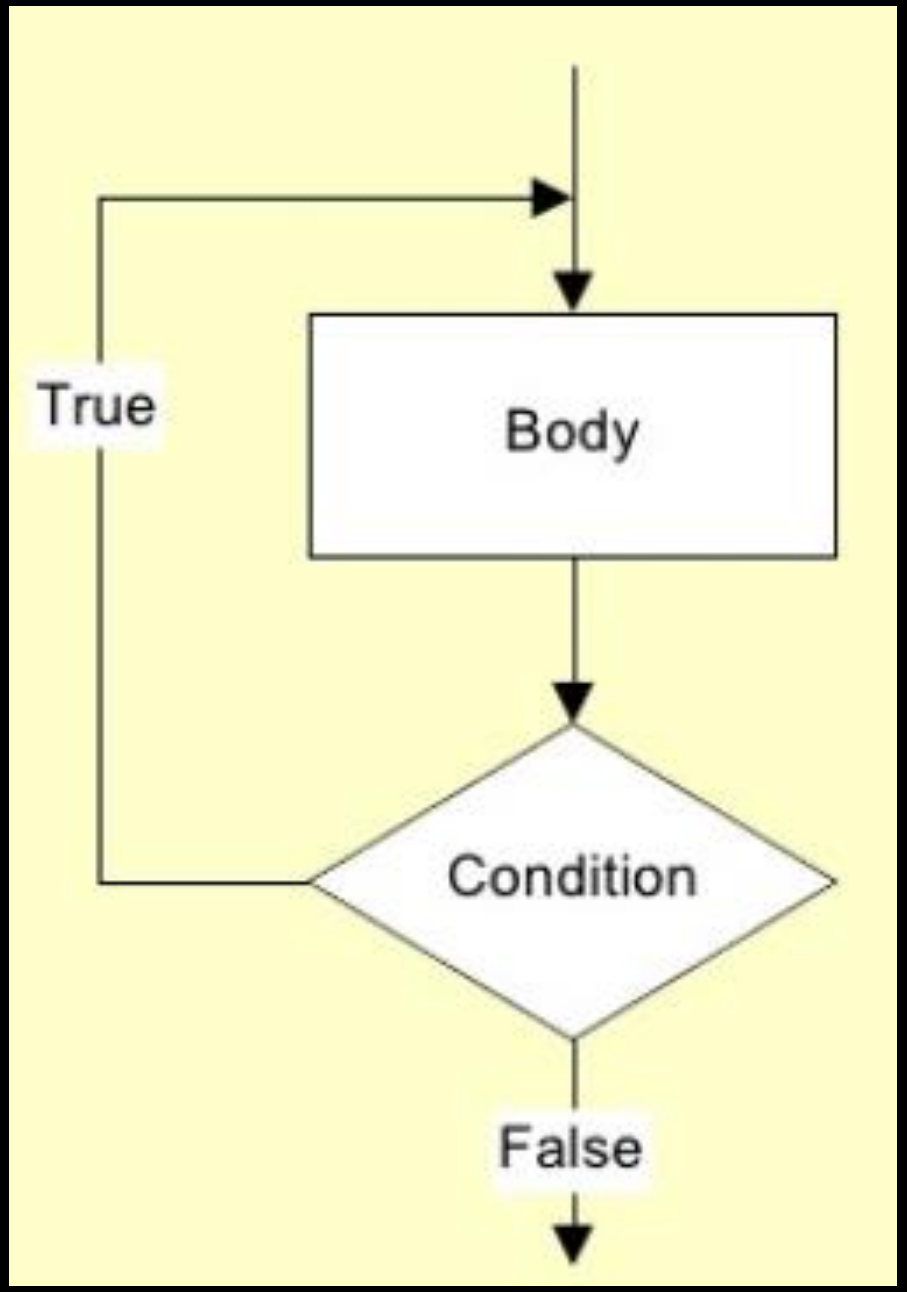
# Repetition

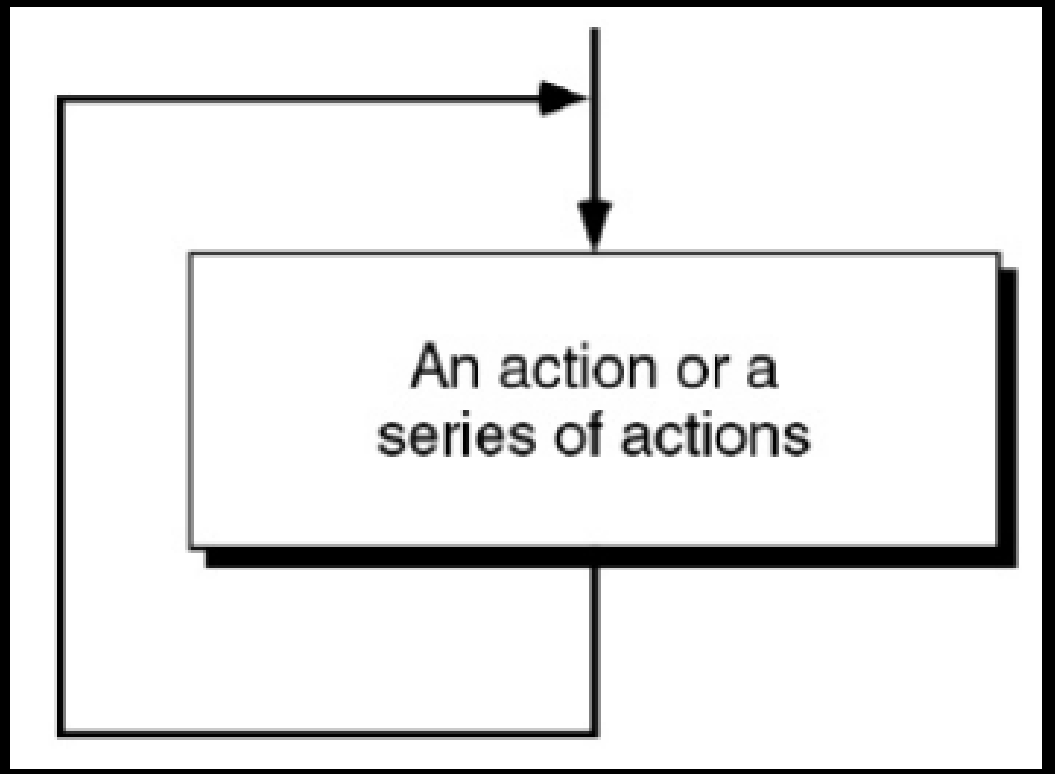


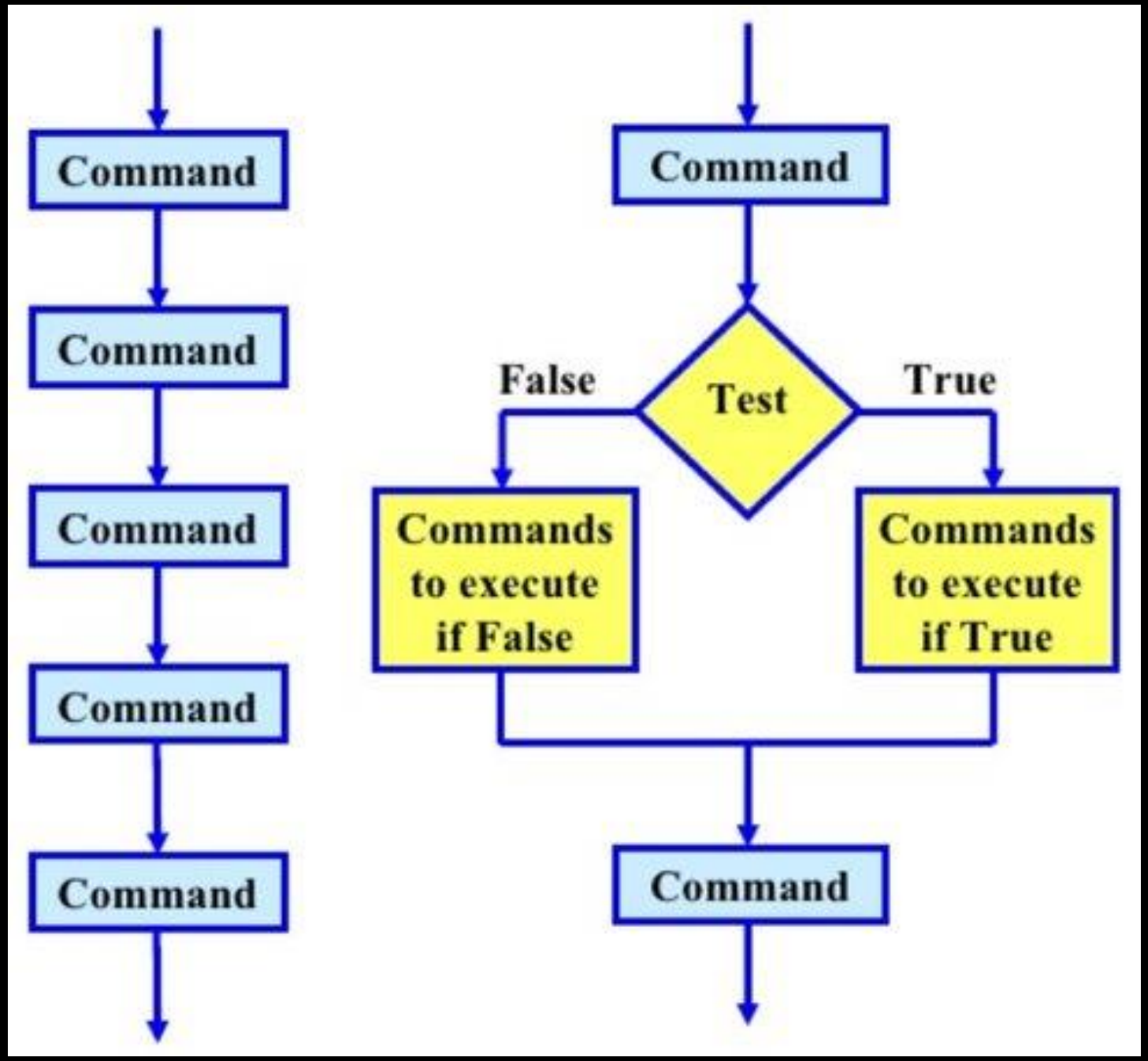
# Not forever!!



# Not forever!!



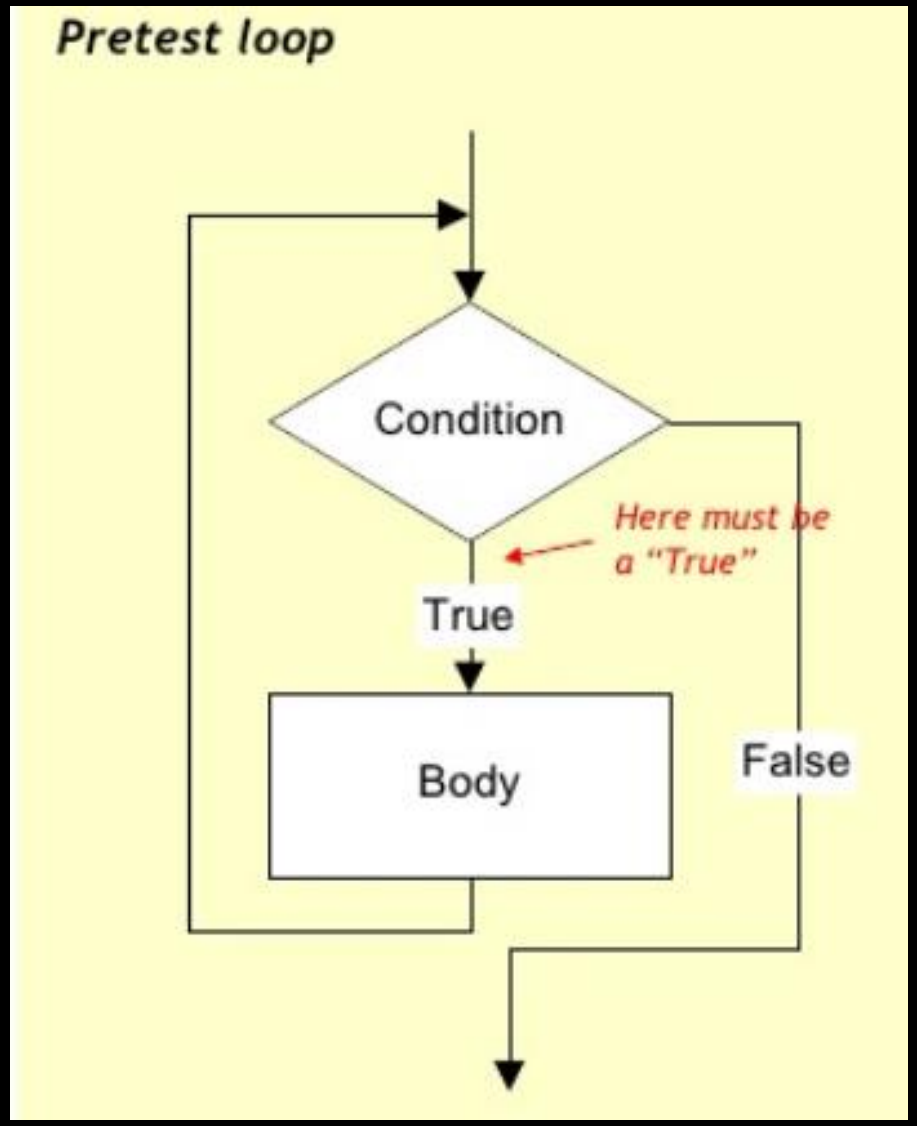



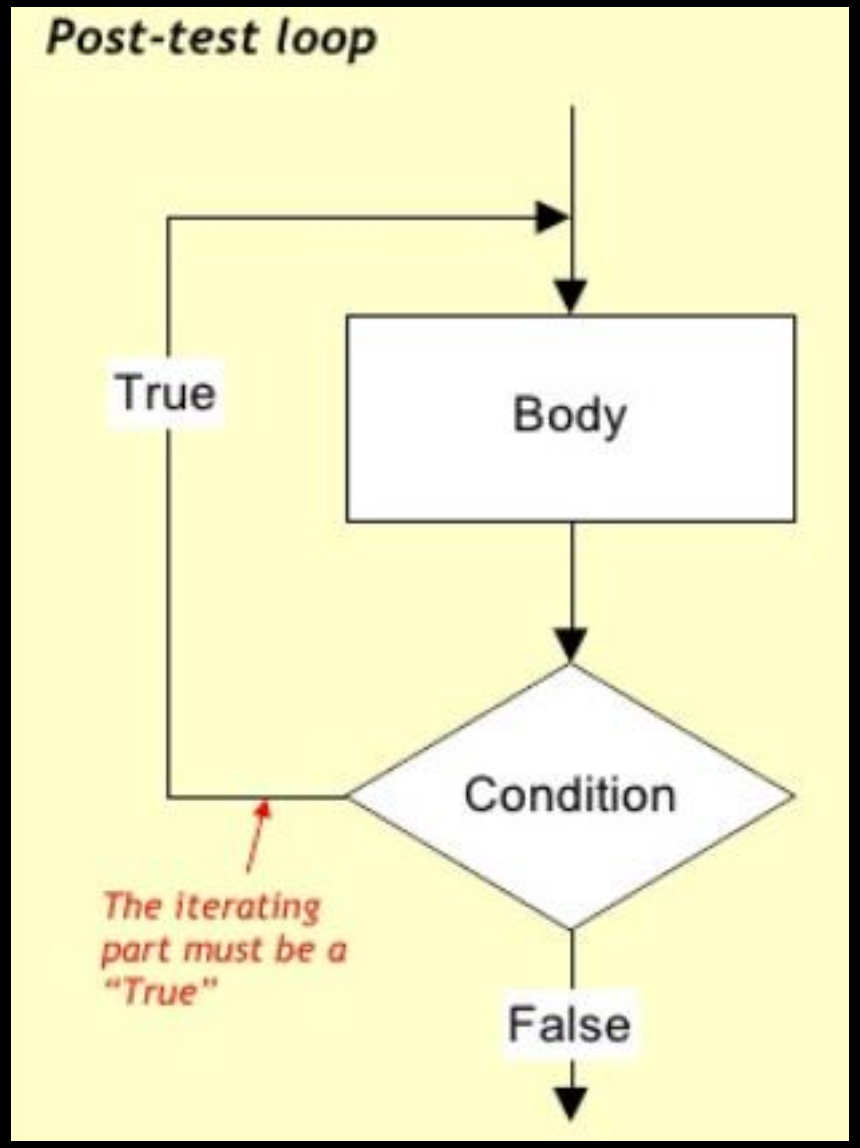



# Repetition
# Can you realize with these control structures?

https://www.slideshare.net/ldunphy/iteration

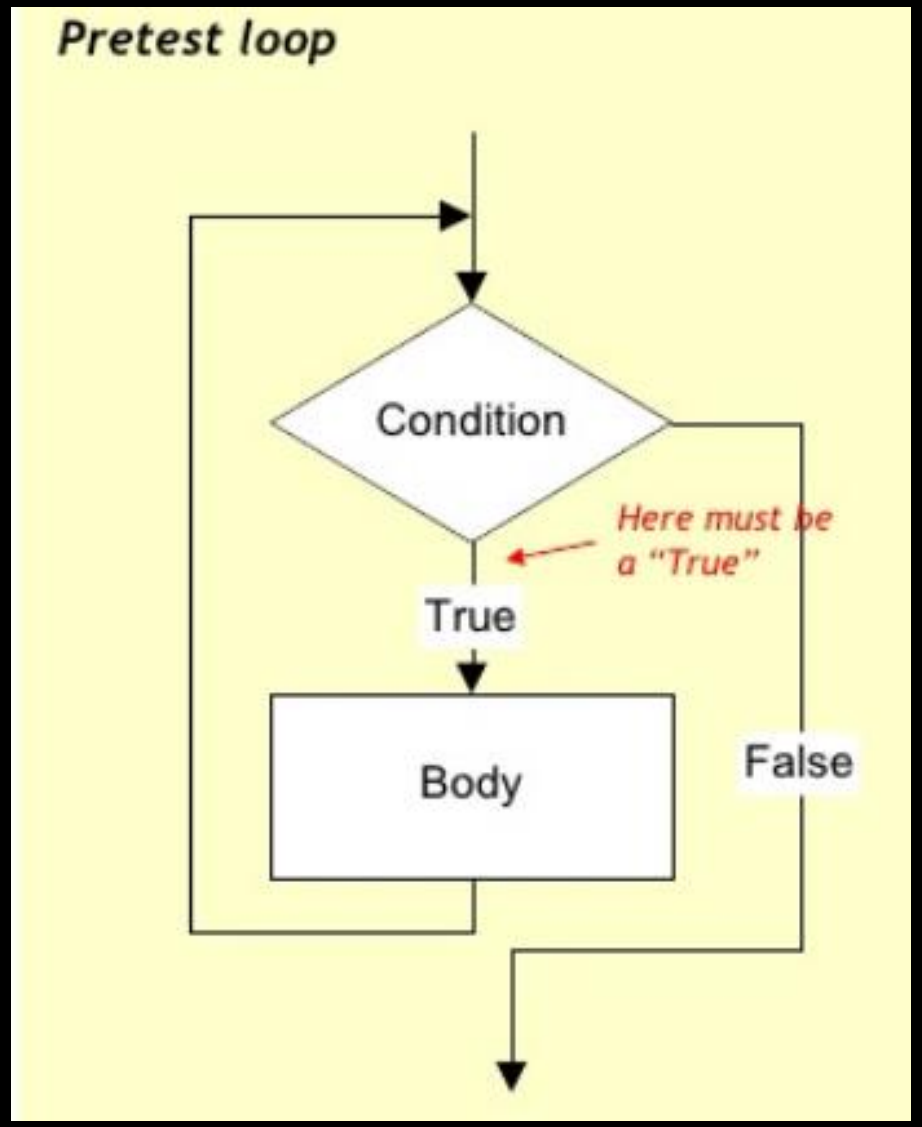
# Types of Loops

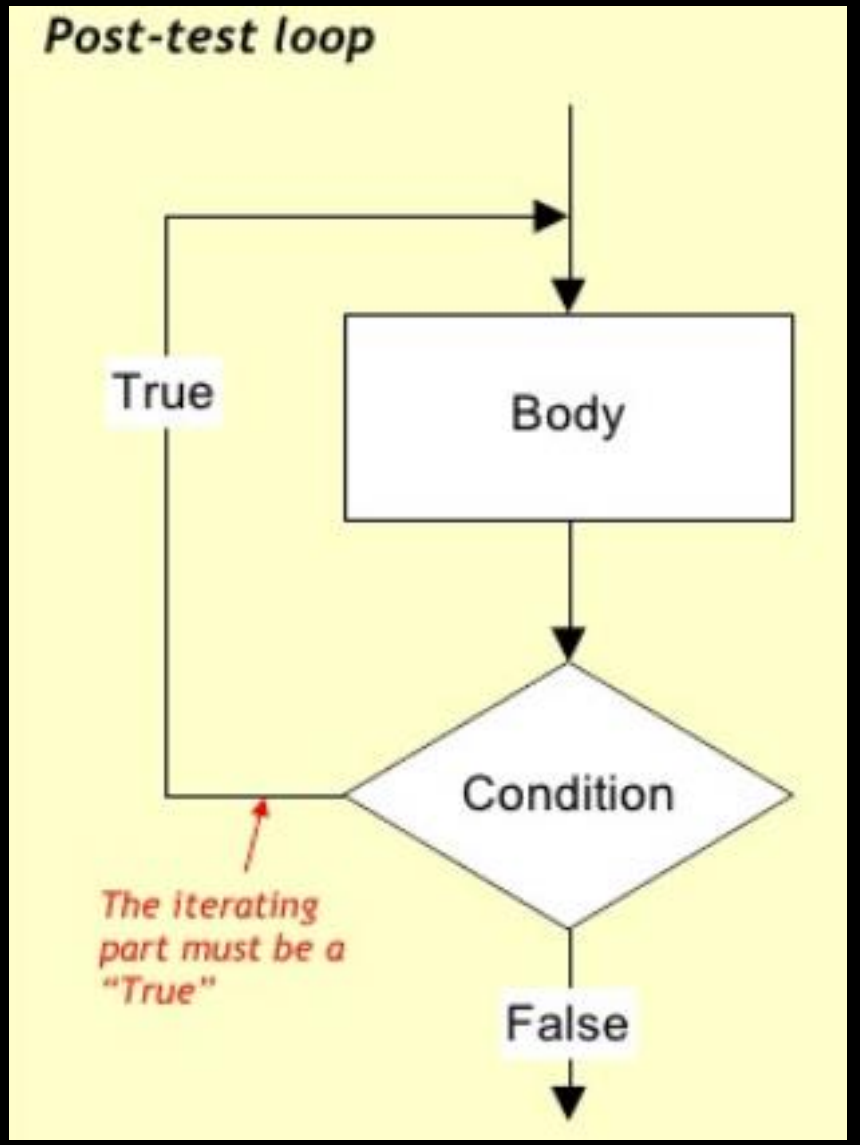



https://www.slideshare.net/dinakan1/06-1-repetition

# Types of Loops

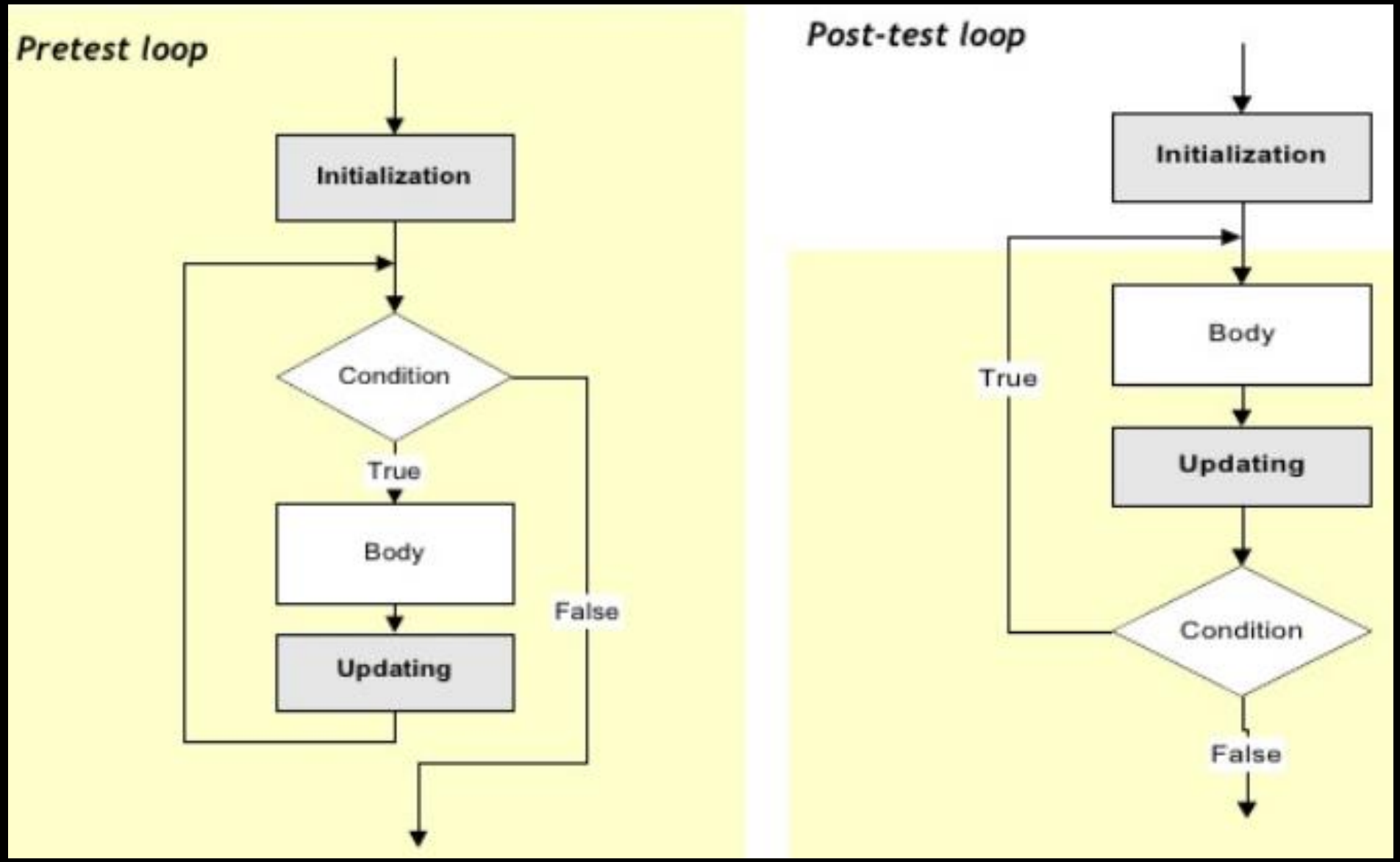



Does a condition guarantee finiteness?!

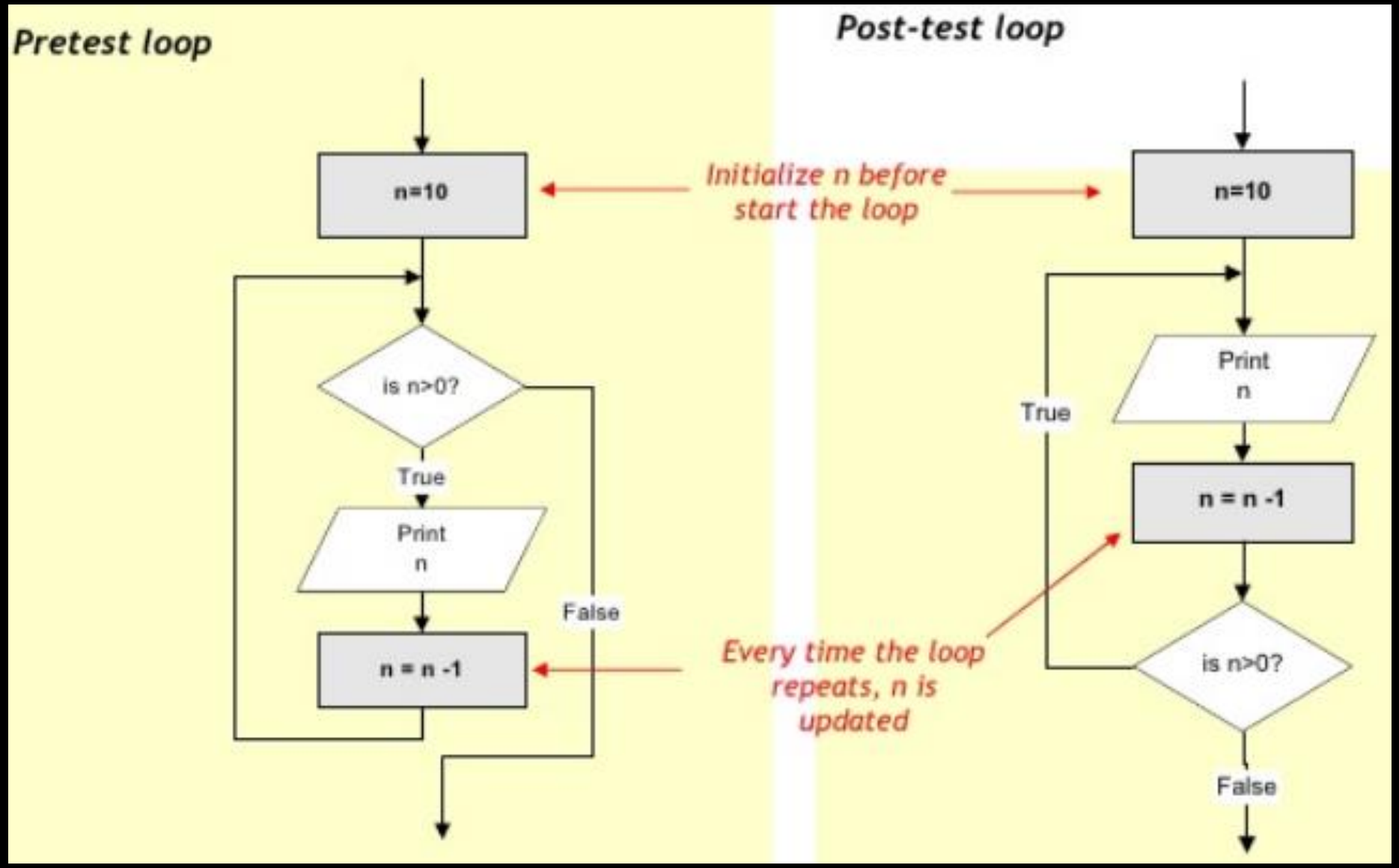
https://www.slideshare.net/dinakan1/06-1-repetition

# Types of Loops

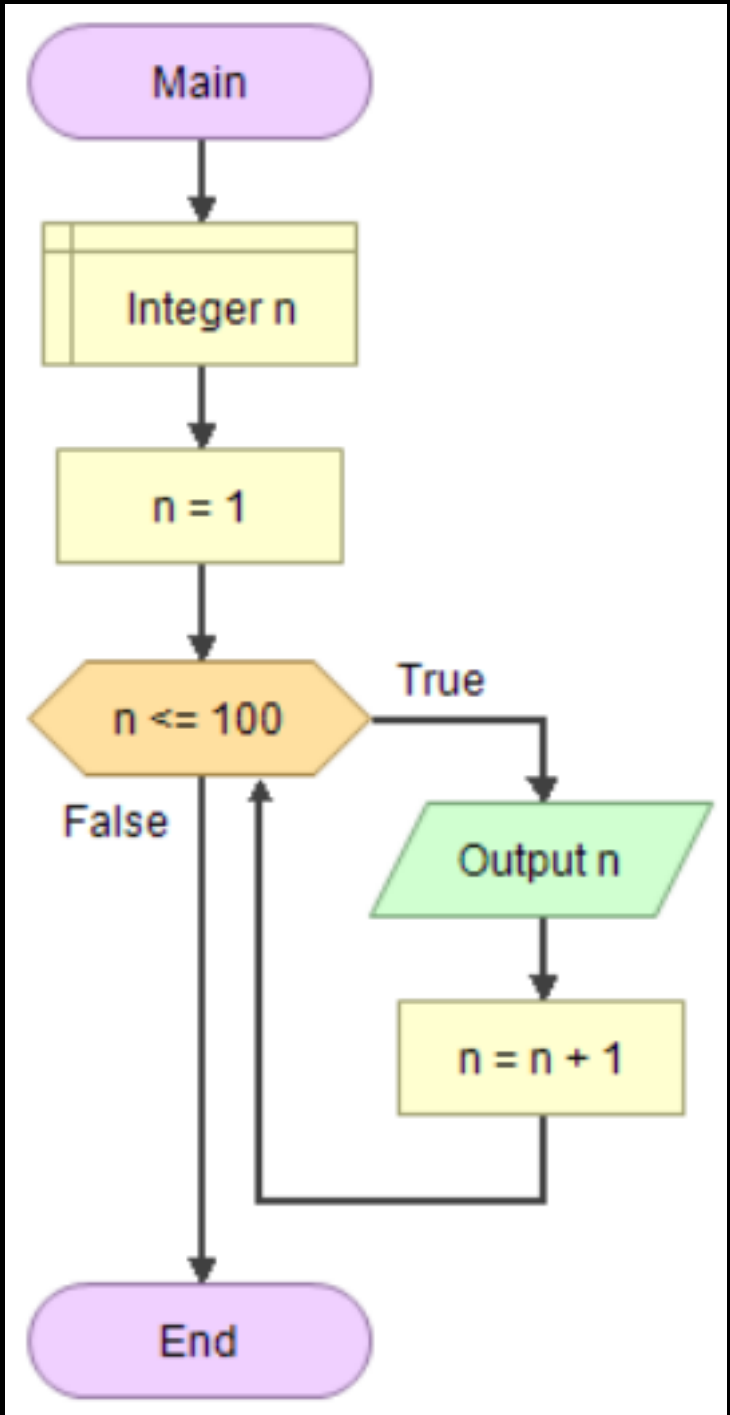



https://www.slideshare.net/dinakan1/06-1-repetition

# Types of Loops

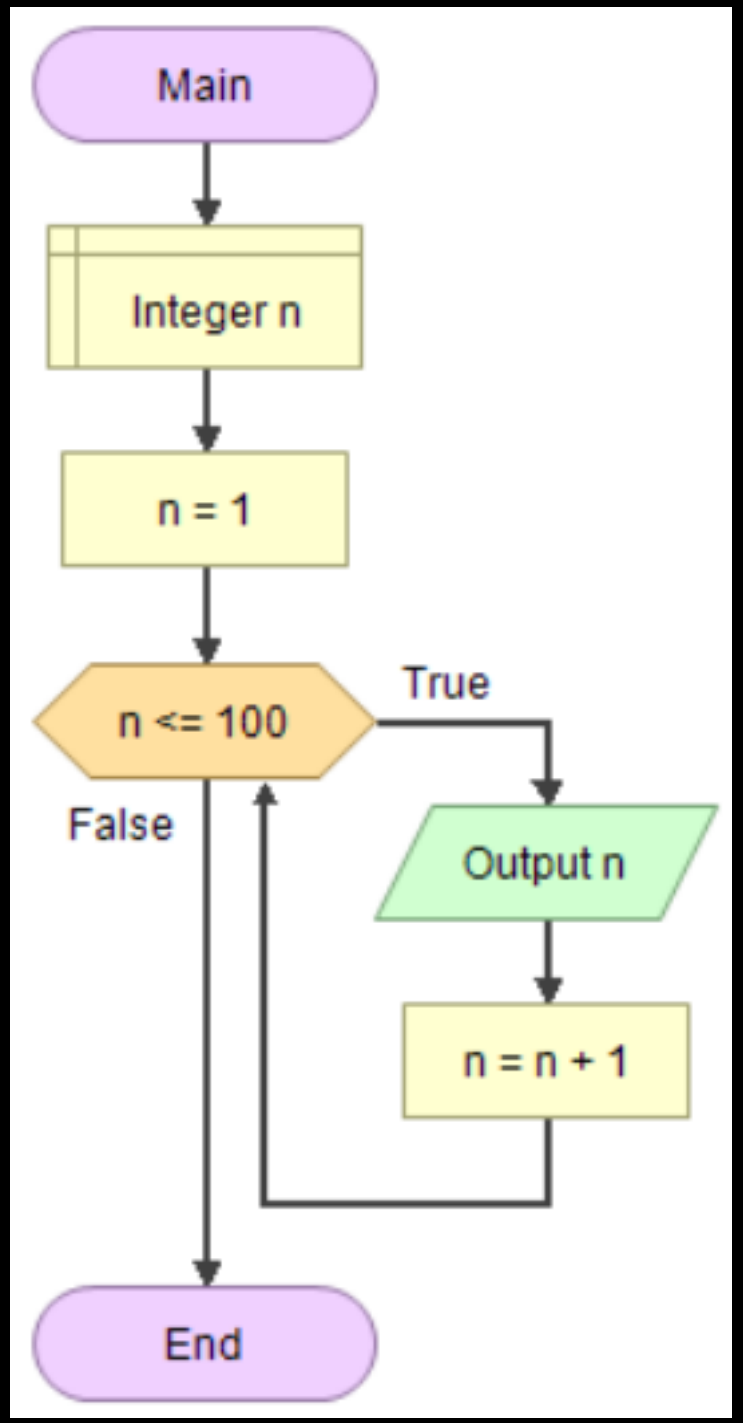


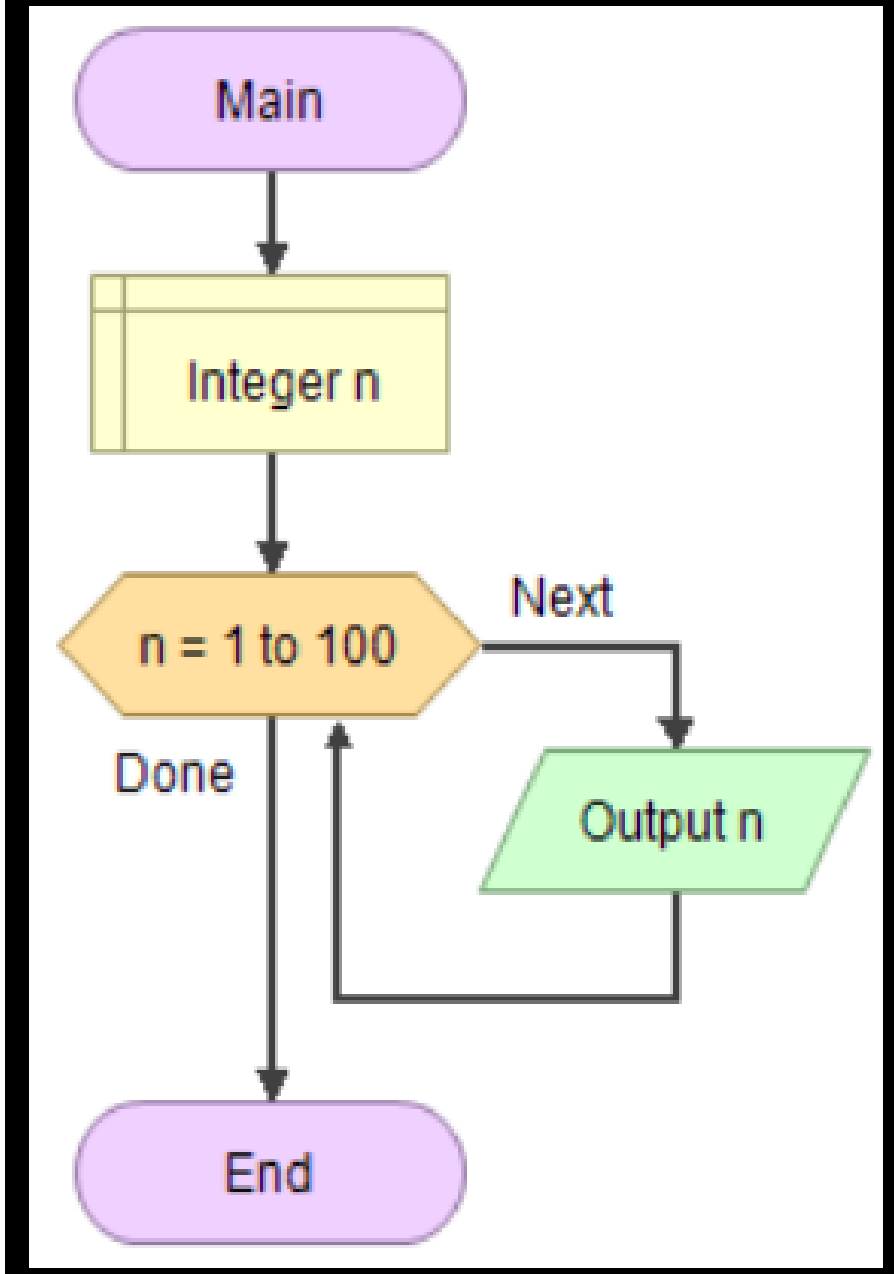

https://www.slideshare.net/dinakan1/06-1-repetition




# Repetitive Structure in Flowgorithm

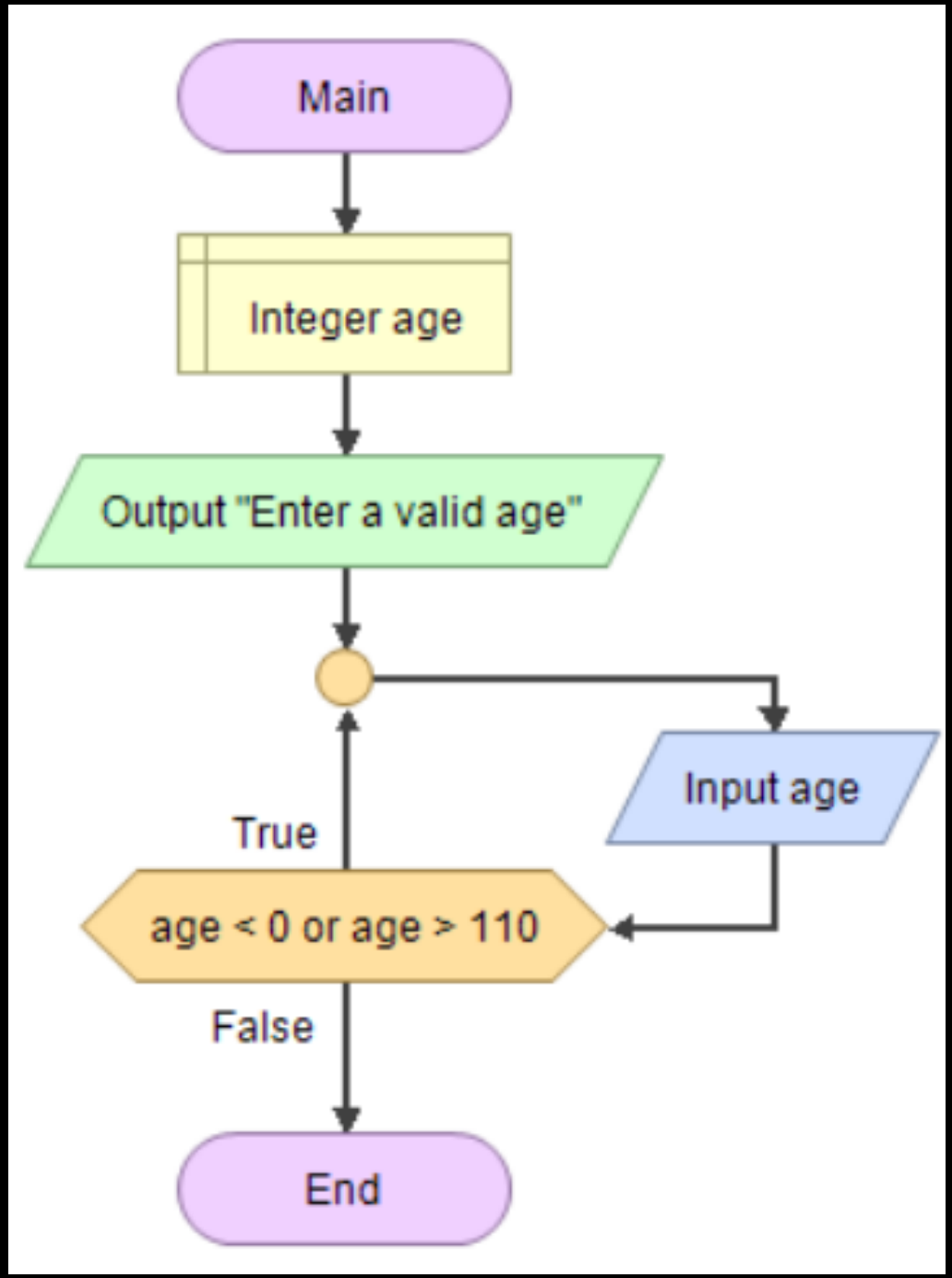
http://flowgorithm.org/documentation/while.htm




# Repetitive Structures in Flowgorithm

http://flowgorithm.org/documentation/for.htm




# Repetitive Structure in Flowgorithm

http://flowgorithm.org/documentation/do.htm

# Example – Sequential

Develop an algorithm that will calculate an hourly employee's weekly pay

Step 1 : Understand the problem
Input : pay rate and number of hours
Process: pay = rate * hours
Output: pay

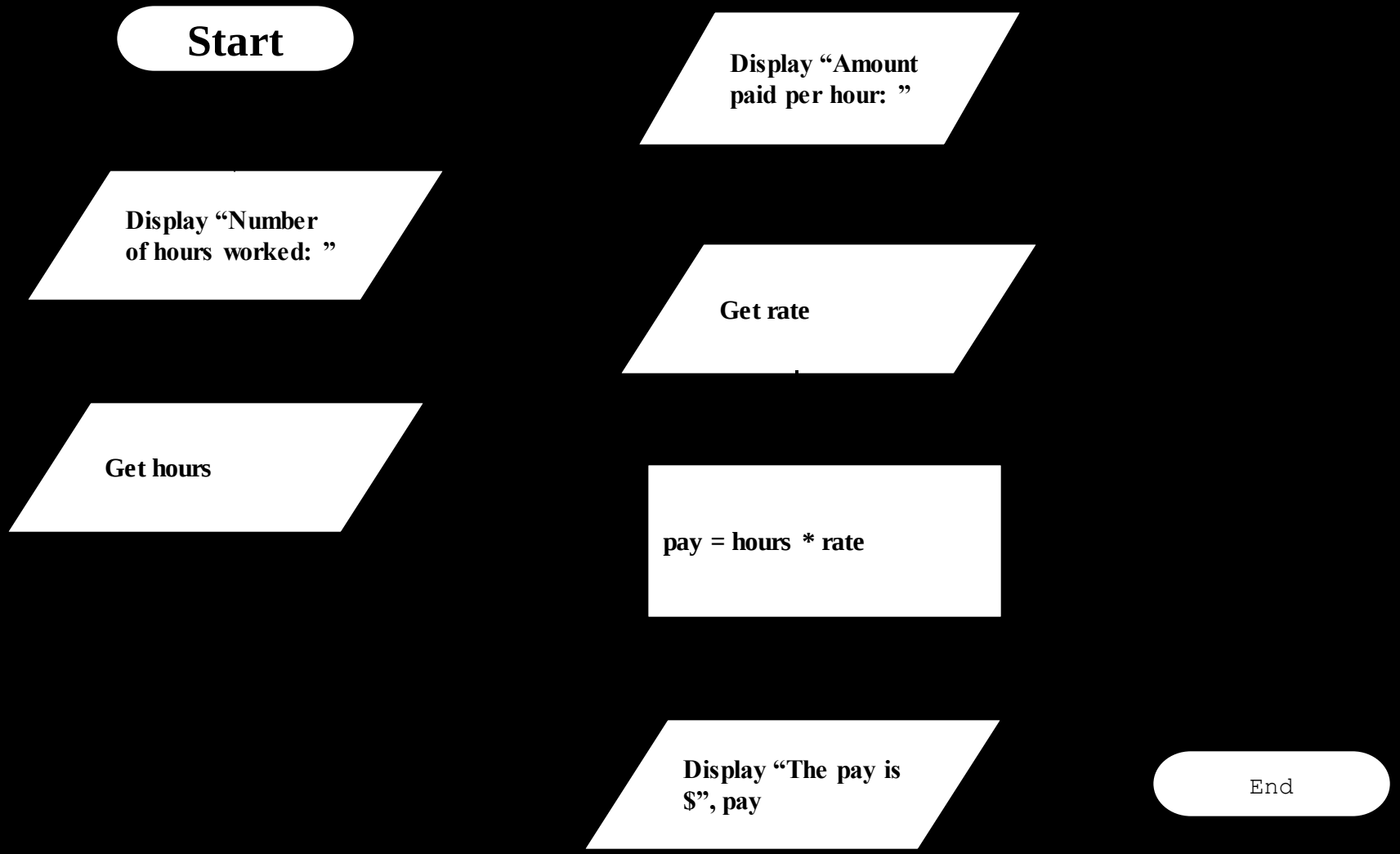
# Algorithm – Calculate pay

Plain English
Ask for the pay rate and the number of hours worked. Multiply the pay rate by the number of hours. The result is the pay
Pseudocode - looks like code, but not a real language

1. Variables: hours, rate, pay
2. Display “Number of hours worked: ”
3. Get hours
4. Display “Amount paid per hour: ”
5. Get rate
6. pay = hours * rate
7. Display “The pay is $” , pay

Notice the importance of order and lack of ambiguity

# Example - Flowchart



# Example – Decision Making

Develop an algorithm that will figure out if a number is positive

Step 1 : Understand the problem
Input : The number
Process: Is it greater than zero?
Output: Message that number is positive or non-positive

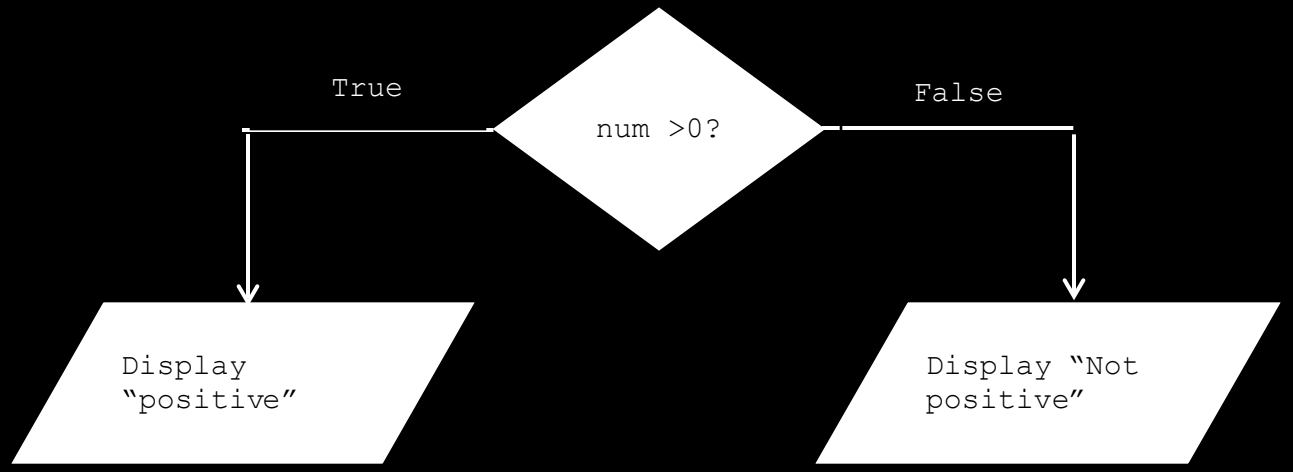
# Algorithm

Plain English
Ask for the number. Check if it is greater than zero. If it is, it is a positive number. If not (i.e. else), it is not positive
Pseudocode -

1. Variable: num
2. Display “Enter the number: ”
3. Get num
4. if num > 0
5. Display “It is positive”
6. Else
7. Display “It is not positive”

# Flowcharts-Decision Making Example

If num > 0 display “Positive”
Else (that means 0 or negative) display “Not Positive”



# Looping

- Develop an algorithm that will add the numbers from 1 to 10. That is 1 + 2 + 3 + …+ 10 = 55

Step 1 : Understand the problem
Input : None needed
Process: Add 0 to 1 to get a new sum. Add 2 to the old sum to get a new sum. Add 3 to the old sum to get a new sum……Add 10 to the old sum to get a new sum
Output: The new sum after 10 iterations

# Algorithm

**Plain English**

Start count at 1.
Add to the sum (originally zero) to get a new sum.
Increment the count.
Repeat the last two steps 10 times

**Pseudocode -**

While_Example

1. Numeric Variables: counter = 1, sum = 0
2. While counter <= 10:
3. sum = sum + counter
4. counter = counter + 1
5. EndWhile
6. Display “ The sum is”, sum

# THANK YOU