

# Understanding Program Structure and Design

## Lecture 2.1

Unit 2: Understanding Program Structure and Design

CPE 1102L | Programming Logic and Design

© 2018

# Unit 2: Understanding Program Structure and Design

- ✓ Features of good program design
- ✓ The disadvantages of unstructured spaghetti code
- ✓ The three basic structures—sequence, selection, and loop
- ✓ Using a priming input to structure a program
- ✓ The need for and recognizing structure

# Features of Good Programming Design

UNIT 2: UNDERSTANDING PROGRAM STRUCTURE AND DESIGN

# Programming style

- Programming style is a term used to describe the effort a programmer should take to make his or her code easy to read and easy to understand.
- Good organization of the code and meaningful variable names help readability, and liberal use of comments can help the reader understand what the program does and why.
- Style Principle

***“Structure and document your program the way you wish other programmers would.”***

# Good Programming Practices



Indentation

Meaningful Variable Names

Internal Documentation

External Documentation

Miscellaneous Comments

# Structured vs Unstructured

UNIT 2: UNDERSTANDING PROGRAM STRUCTURE AND DESIGN

# The Disadvantages of Unstructured Spaghetti Code

- Spaghetti code is snarled, **unstructured program logic**.
- Unstructured programs are programs that **do not follow the rules of structured logic**.
- Structured programs are programs that **do follow the rules of structured logic**.

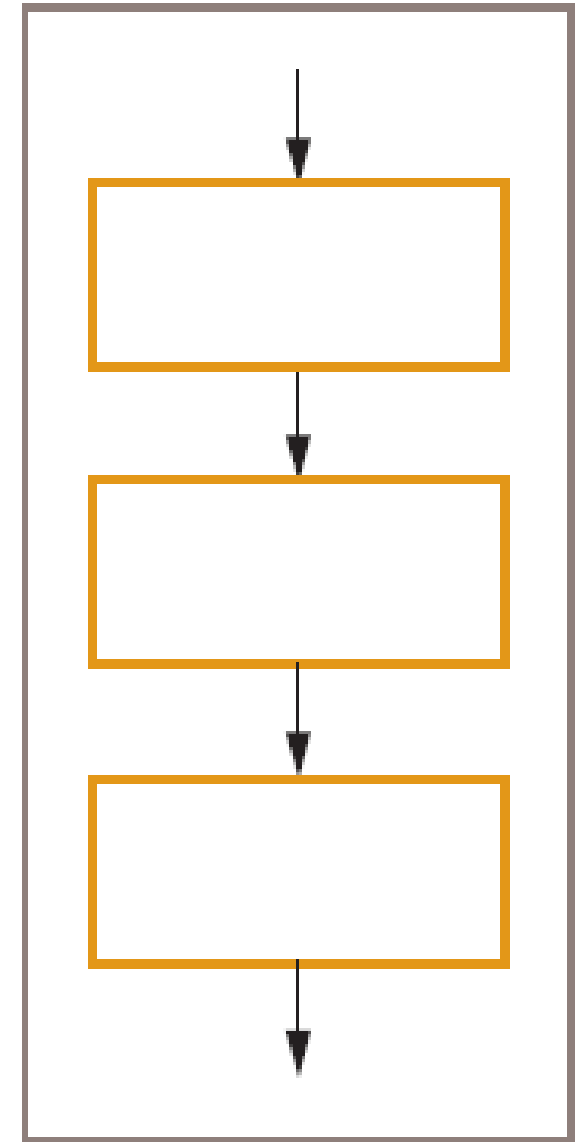
# The Three Basic Structures—Sequence, Selection, and Loop

- A **structure** is a basic unit of programming logic
- Each structure is a sequence, selection, or loop.



# Sequence Structure

- A **sequence structure** contains series of **steps executed in order**.
- A sequence can contain any number of tasks, but there is no option to branch off, skipping any of the tasks.



# Selection Structure

- A **selection structure** or decision structure contains a question, and, depending on the answer, **takes one of two courses of action before continuing with the next task.**
- An end-structure statement designates the end of a pseudocode structure.
- An if-then-else is another name for a dual-alternative selection structure.

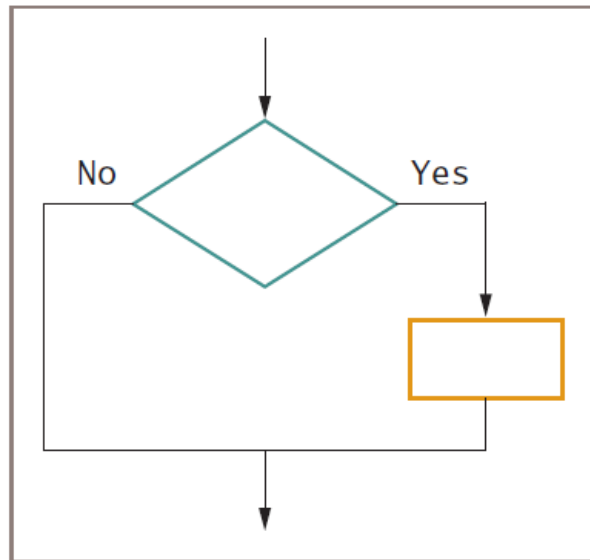
# Selection Structure

- **Dual-alternative ifs** (or dual-alternative selections) define one action to be taken when the tested condition is true and another action to be taken when it is false.
- **Single-alternative ifs** (or single-alternative selections) take action on just one branch of the decision.
- The **null case** or null branch is the branch of a decision in which no action is taken.

# Selection Structure

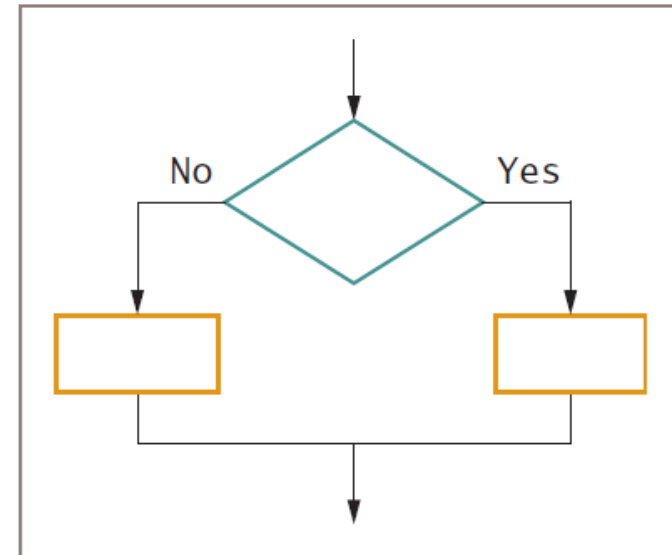
## Single-alternative ifs

```
if someCondition is true then
    do oneProcess
endif
```



## Dual-alternative ifs

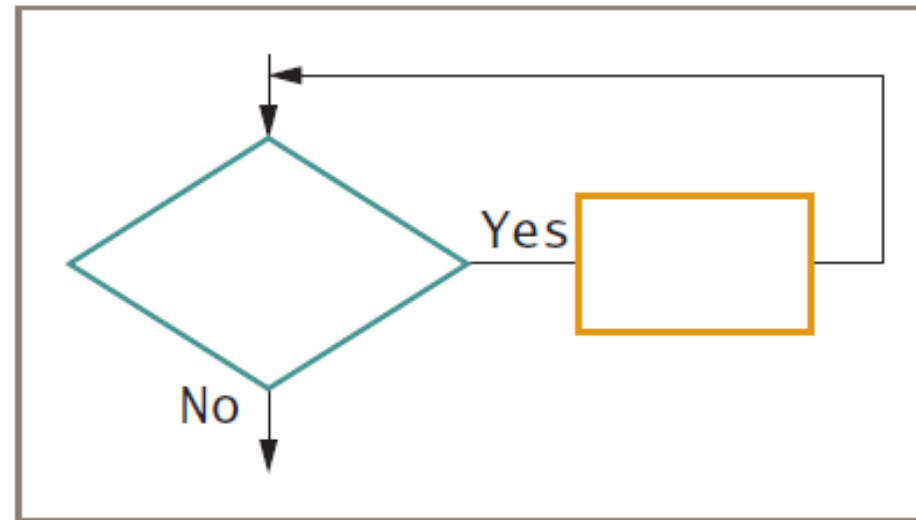
```
if someCondition is true then
    do oneProcess
else
    do theOtherProcess
endif
```



# Loop Structure

- A loop structure continues to repeat actions while a test condition remains true.

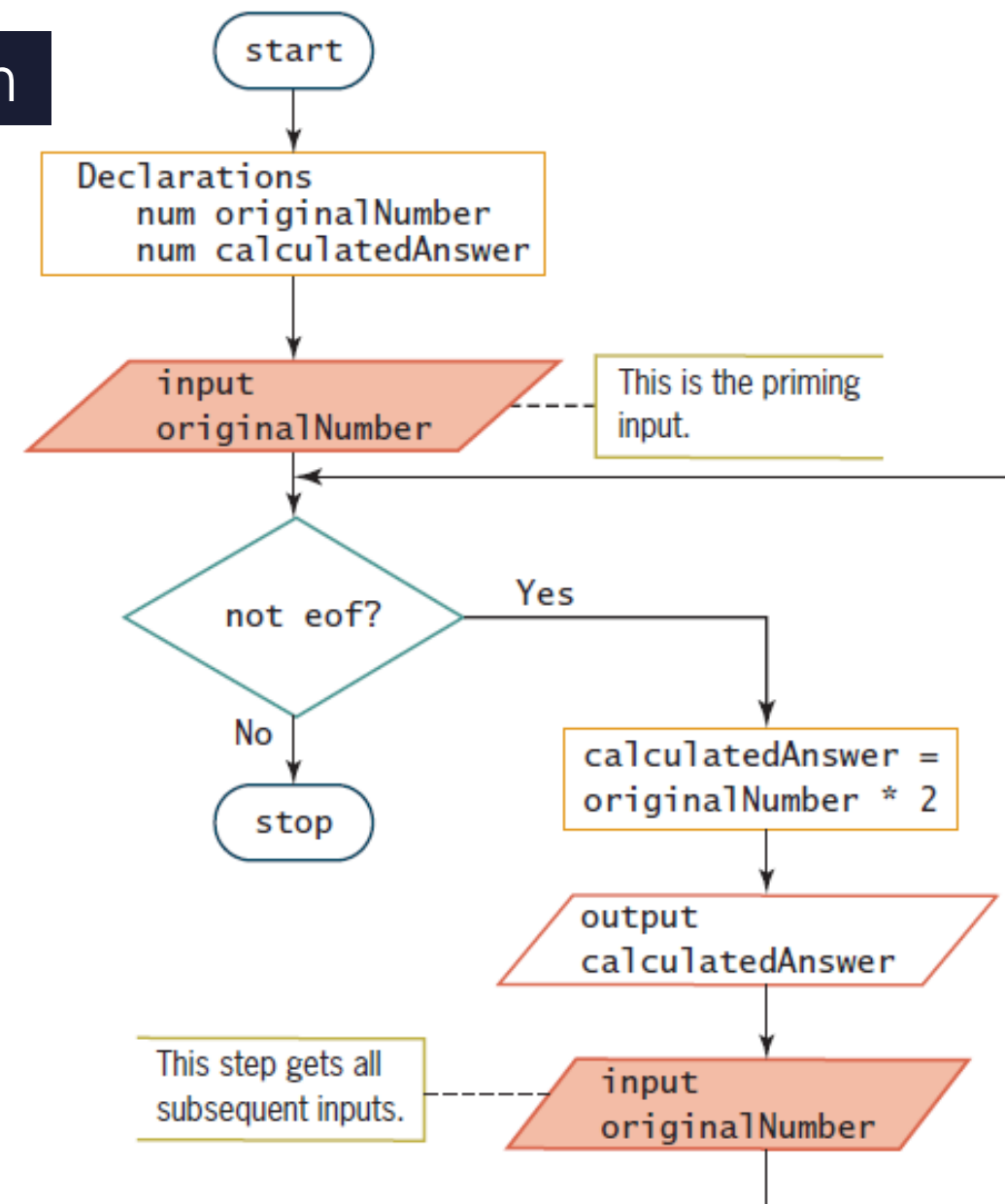
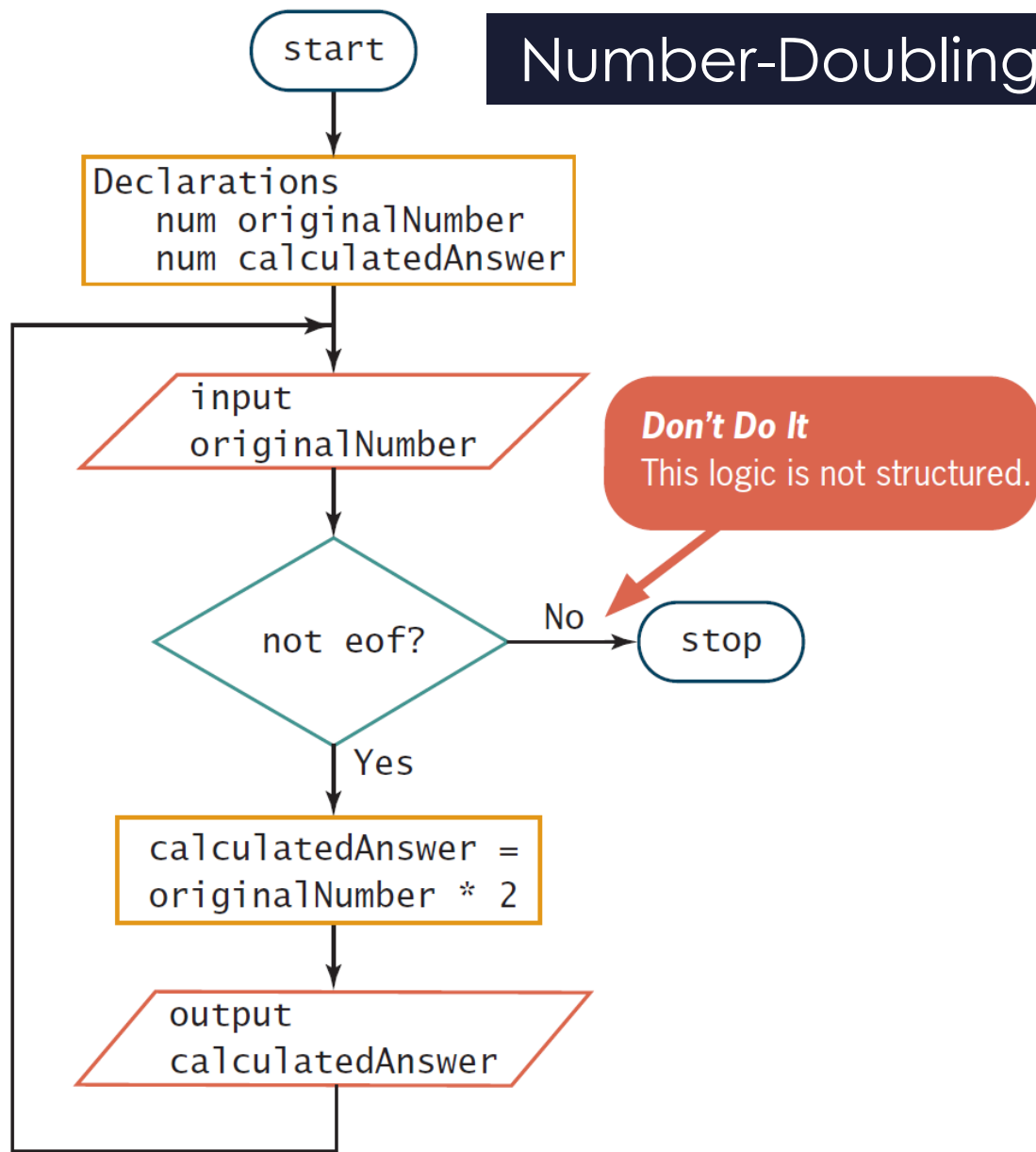
```
while testCondition continues to be true do  
    someProcess  
endwhile
```



# Using a priming input to structure a program

- A **priming input** is the statement that gets the **first input value prior to starting a structured loop**. Usually, the last step within the loop body gets the next and all subsequent input values.

# Number-Doubling Problem

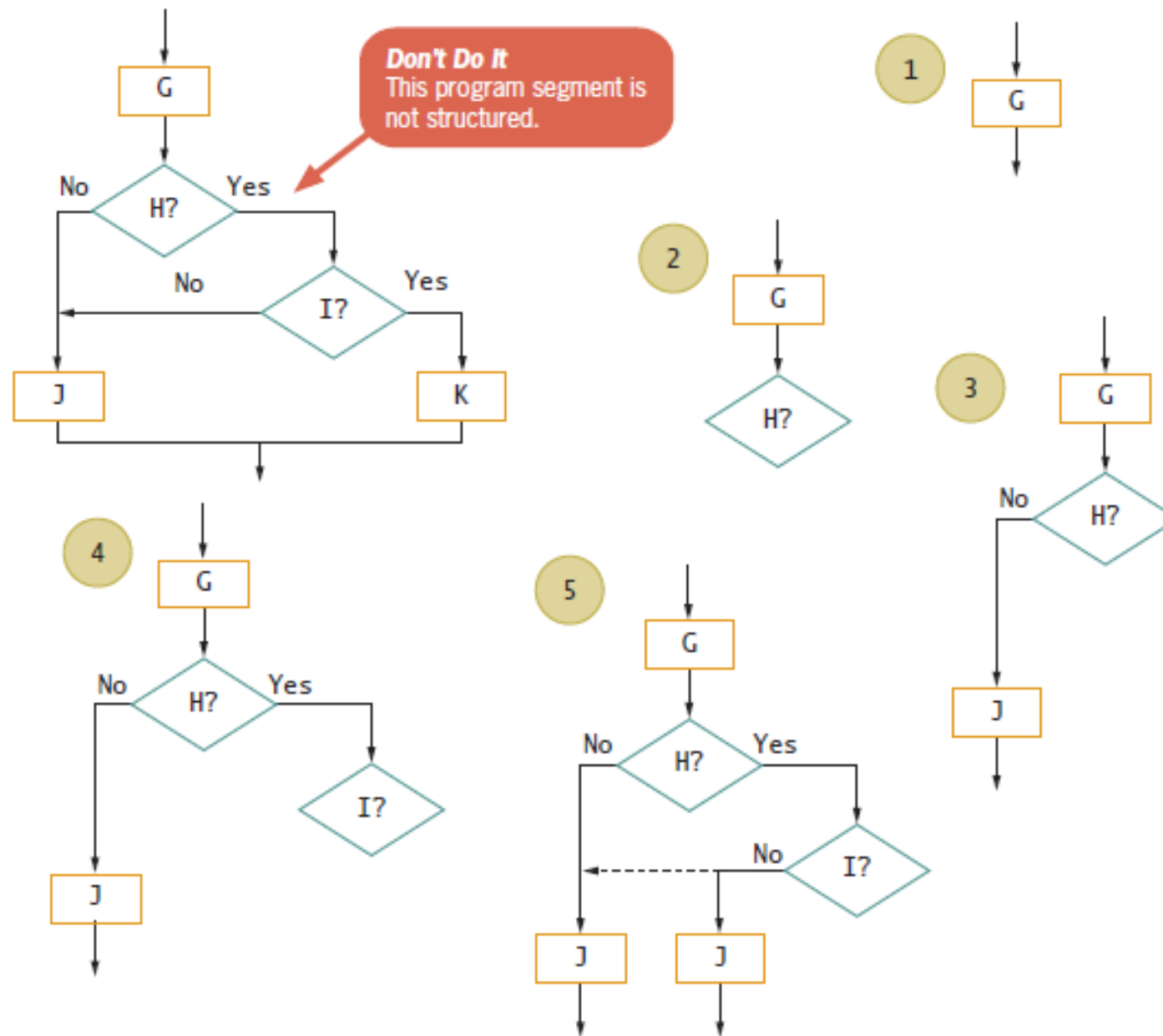


# The need for structure and recognizing structure

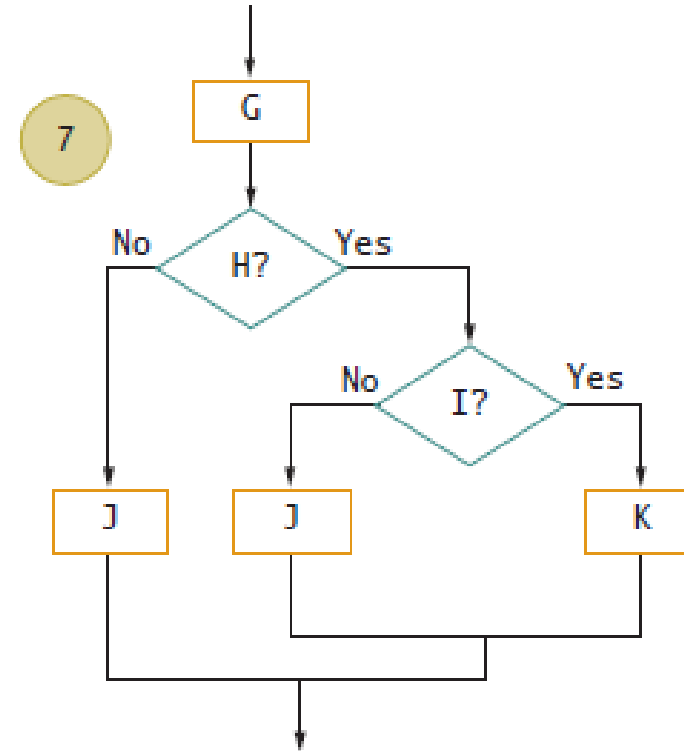
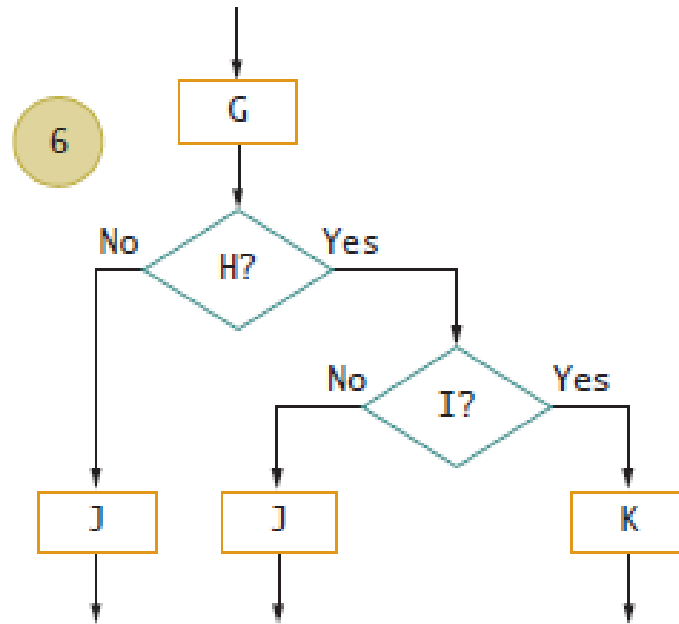
- Programmers use structured techniques to promote clarity, professionalism, efficiency, and modularity.
- One way to order an unstructured flowchart segment is to imagine it as a bowl of spaghetti that you must untangle.



# Example



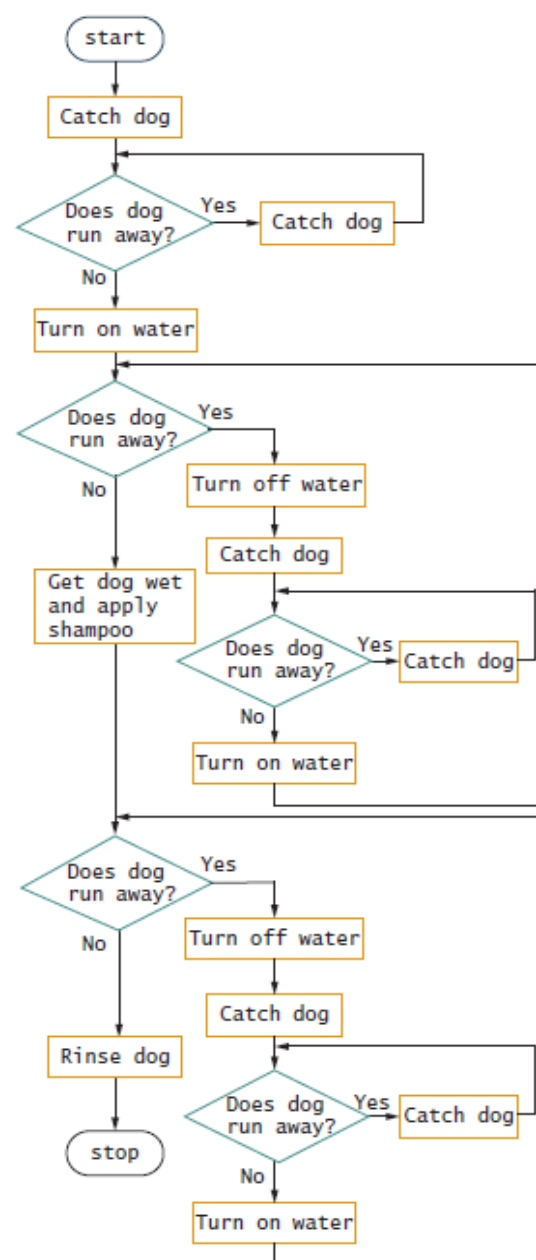
# Example



# Structuring and modularizing unstructured logic

- Any set of logical steps can be rewritten to conform to the three structures: sequence, selection, and loop.

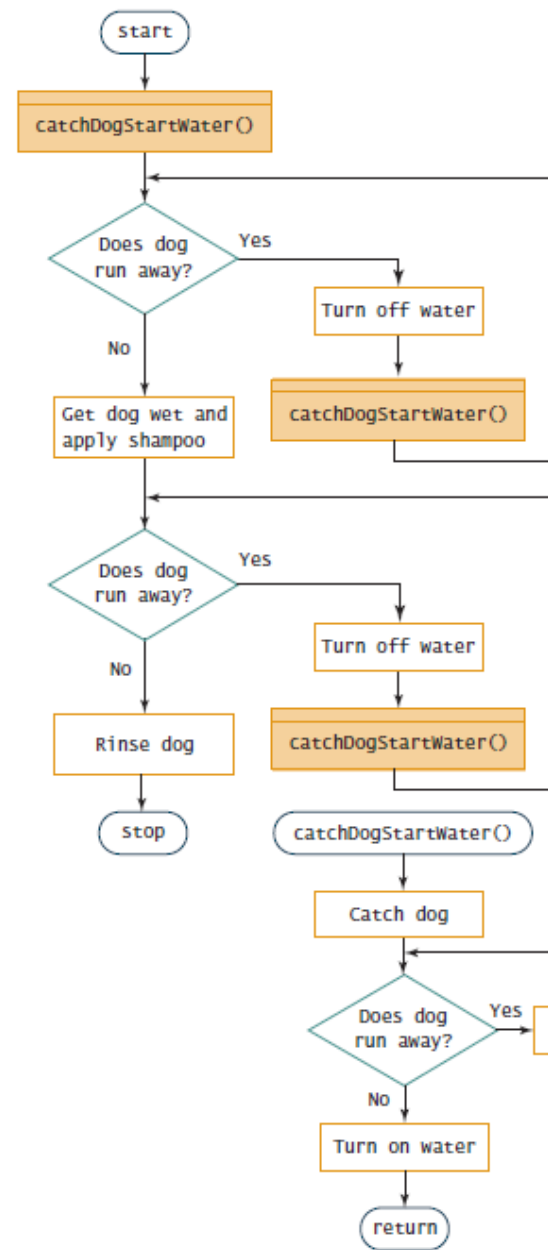
# Example



```

start
Catch dog
while dog runs away
    Catch dog
endwhile
Turn on water
while dog runs away
    Turn off water
    Catch dog
while dog runs away
    Catch dog
endwhile
Turn on water
endwhile
Get dog wet and apply shampoo
while dog runs away
    Turn off water
    Catch dog
while dog runs away
    Catch dog
endwhile
Turn on water
endwhile
Rinse dog
stop
  
```

# Example



```

start
catchDogStartWater()
while dog runs away
    Turn off water
    catchDogStartWater()
endwhile
Get dog wet and apply shampoo
while dog runs away
    Turn off water
    catchDogStartWater()
endwhile
Rinse dog
stop

catchDogStartWater()
Catch dog
while dog runs away
    Catch dog
endwhile
Turn on water
return
    
```



COMPUTER ENGINEERING  
UNIVERSITY *of* SAN CARLOS