CS118 – Programming Fundamentals

Lecture # 03
Wednesday, August 21, 2019
FALL 2019
FAST – NUCES, Faisalabad Campus

Zain Iqbal

Algorithms and Flowcharts

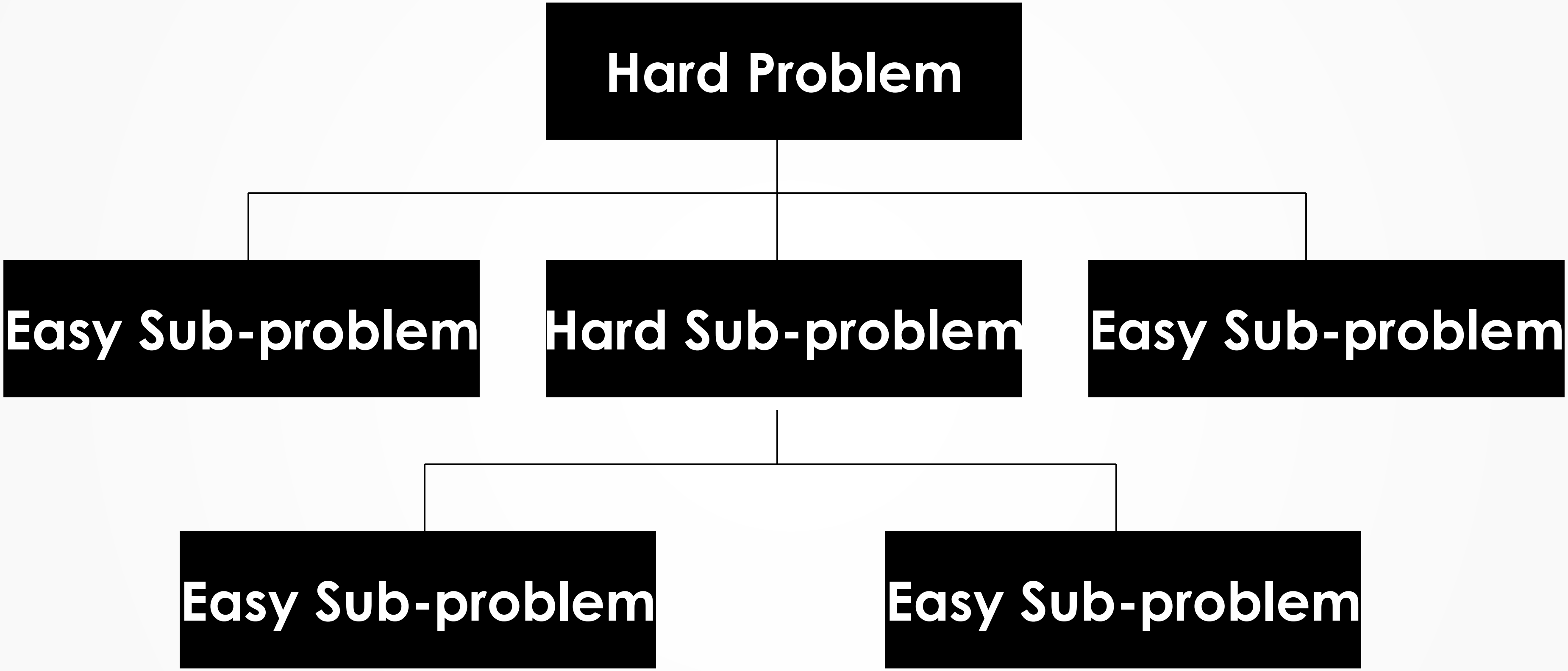
Algorithms

- A concept that pervades all areas of computer science
- Algorithm is a process that a computer could carry out to complete a **well defined task within finite time and resources**
- The **objective of computer science** is to solve problems by developing, analyzing, and implementing algorithmic solutions

Al-Khwarizimi Principle

- All complex problems can be broken into simpler sub-problems
- Solve a complex problem by breaking it down into smaller sub-problems and then solve them (in a specified order), one at a time
- When all the steps are solved, the original problem itself has also been solved
- This process is called ***Algorithm***

Divide and Conquer



Steps in Problem Solving

- First produce a general algorithm (one can use **pseudocode**)
- Refine the algorithm successively to get step by step detailed **algorithm** that is very close to a computer language
- **Pseudocode** is an artificial and informal language that helps programmers develop algorithms
  - Pseudocode is very similar to everyday English

Algorithms & Pseudocode

- A typical programming task can be divided into two phases
- **Problem Solving phase:**
  - Produce an ordered sequence of steps that describe solution of problem
  - This sequence of steps is called an **algorithm**
- **Implementation phase:**
  - Implement the program in some programming language

Sample problem

- Compare two numbers given by the user and tell which one is greater?

Pseudocode – Format

```
1. start
2. declare n1, n2
3. input n1,n2
4. if (n1 > n2)
        4.1 print "n1 is greater"
5. else if (n2 > n1)
        5.1 print "n2 is greater"
6. else
        6.1 print "they are equal"
7. end
```

Algorithms and Pseudocode

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

Algorithms and Pseudocode

Algorithm:

- Input a set of four marks
- Calculate their average by summing and dividing the sum by 4
- If average is above 50
    - Print : “Pass”

  else
    - Print “FAIL”

Algorithms and Pseudocode

Pseudocode:

```
Step 1:  Input M1, M2, M3, M4
Step 2: Grade = (M1+ M2+ M3+ M4) / 4
Step 3: if (Grade > 50) then
                Print "PASS"
            else
                Print "FAIL"
            endif
```

Pseudo-Code: Decision Making

- **If-then**
- **General form:**

```
if (condition is met) then
    statement(s)
```

Example:

```
if temperature < 0 then
    wear a jacket
```

Pseudo-Code: Decision Making

If-then-else

General form:

```
if (condition is met) then
    statement(s)
else
    statements(s)
```

Example:

```
if (at work) then
    Dress formally
else
    Dress casually
```

Pseudo-Code: Fast Food Example

- Use pseudo-code to specify the algorithm for a person who is ordering food at a fast food restaurant

- At the food counter, the person can either order or not order the following items:
  - a burger
  - fries and a drink
  - After placing her order the person then goes to the cashier

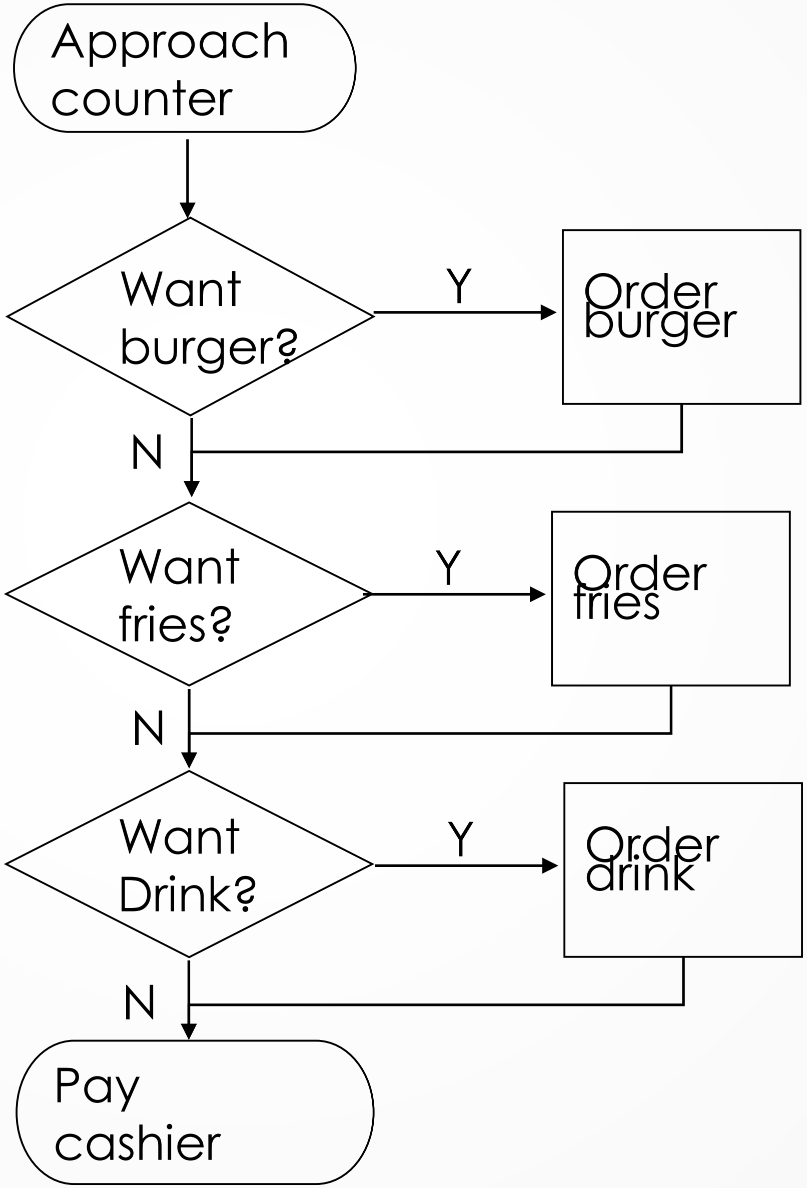
Pseudo-Code: Fast Food Example

```
1. Approach counter
2. if want burger then
      2.1. order burger
3. if want fries then
      3.1. order fries
4. if want drink then
      4.1 order drink
5. Pay cashier
```

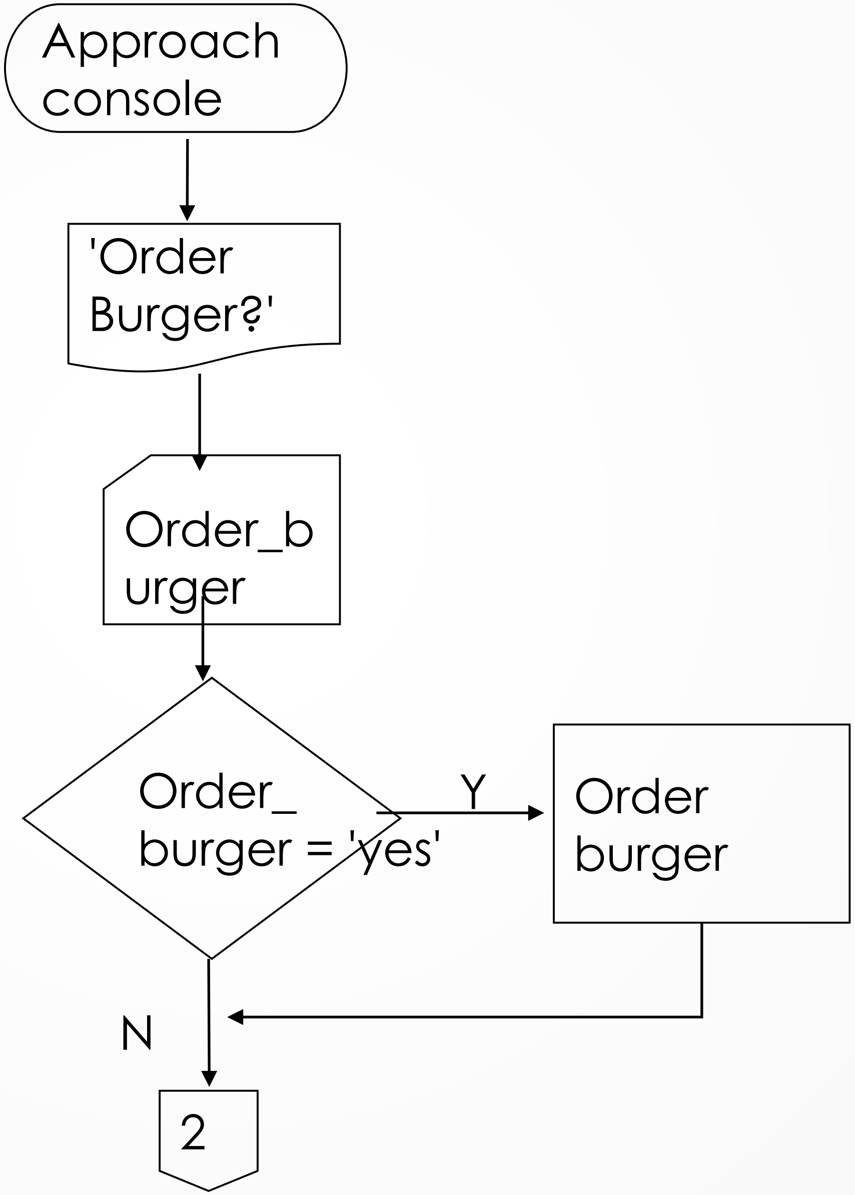
Pseudo-Code: Fast Food Example (Computer)

```
1. Approach counter
2. Output ‘Do you want to order burger?’
3. Input order_burger
4. if order_burger = yes then
    4.1. order_burger
5. Output ‘Do you want to order fries?’
6. Input order_fries
7. if order_fries = yes then
    7.1. order_fries
8. Output 'Do you want to order drink?’
9. Input order_drink
10. If order_drink = yes then
  10.1. order drink
11. Pay cashier
```

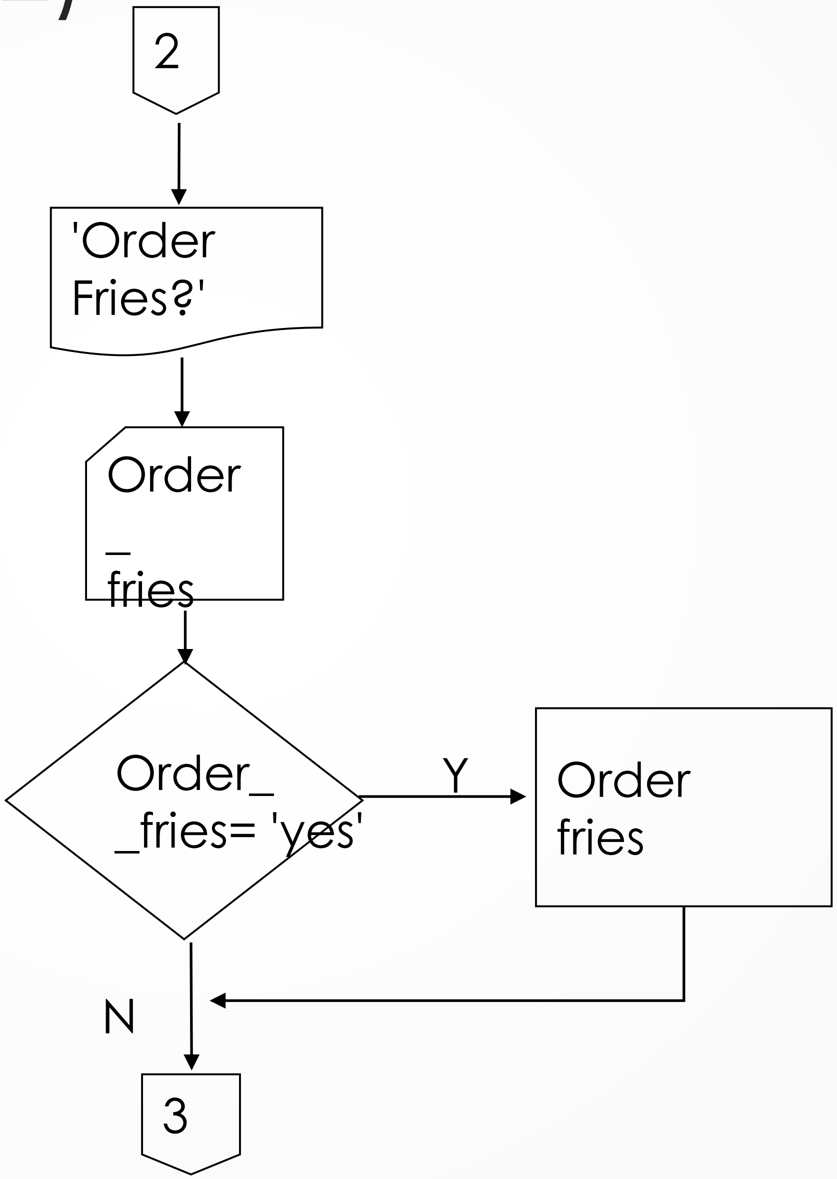
Flowchart: Fast Food Example



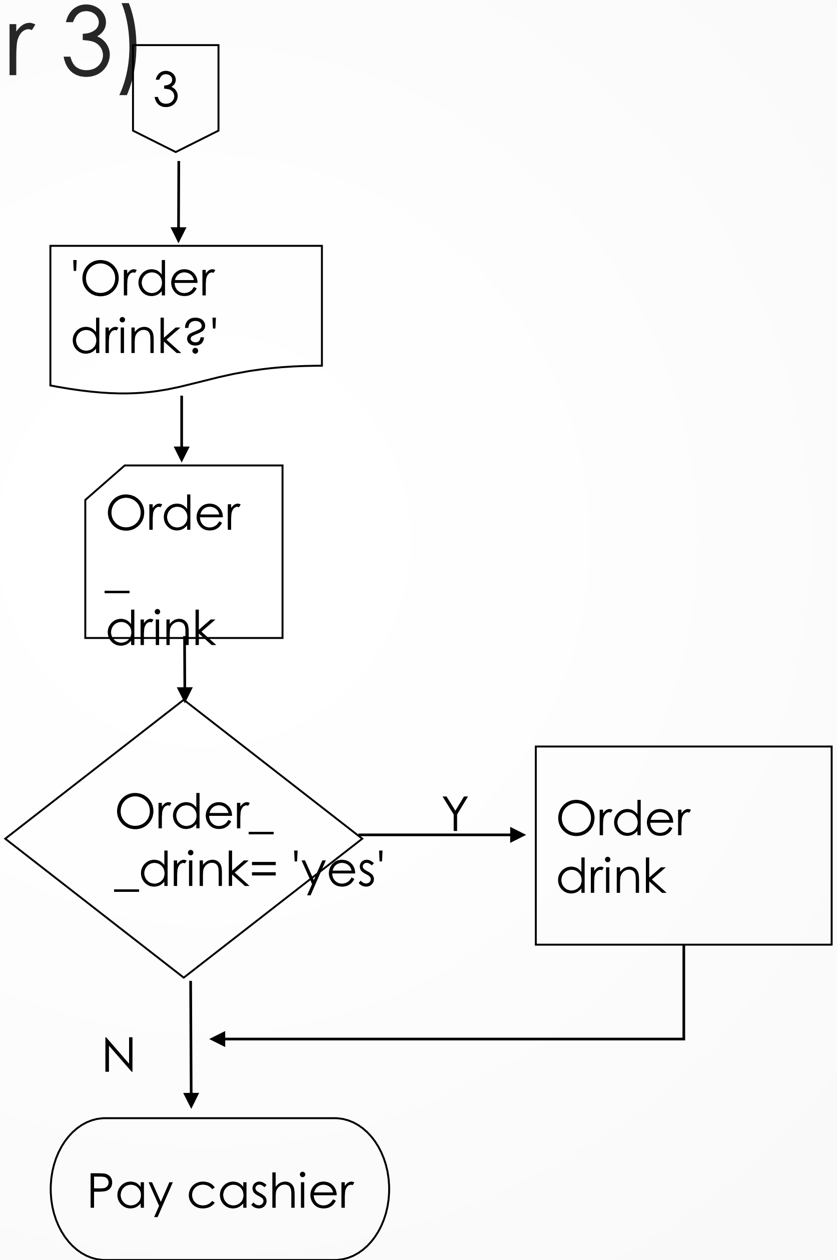
Flowchart: Fast Food Example (Computer)



Flowchart: Fast Food Example (Computer 2)



Flowchart: Fast Food Example (Computer 3)



Pseudo-Code: ATM Example

Use pseudo-code to specify the algorithm for an ATM bank machine. The bank machine has four options:

1) Show current balance

2) Deposit money

3) Withdraw money

4) Quit

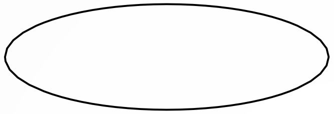
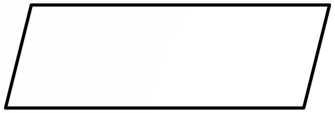
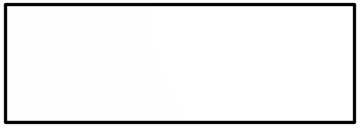
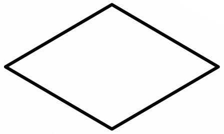
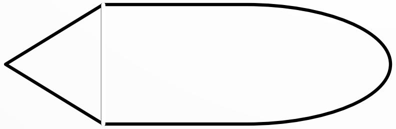
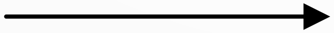
After an option has been selected, the ATM will continue displaying the four options to the person until he selects the option to quit the ATM

Pseudo-Code: ATM Example

```
Approach ATM
Repeat
   Output 'Select option'
   Output '1) Make withdrawal'
   Output '2) Make deposit'
   Output '3) Show balance'
   Output '4) Quit'
   Input option
   If option = deposit then
      Output 'Enter amount to deposit'
      Input amount
      balance ← balance + amount
   If option = withdrawal then
      Output 'Enter amount to withdraw'
      Input amount
      balance ← balance – amount
   If option = ‘Show Balance’
      Output 'Balance is ' balance
Until option = quit
Stop
```

Flow Charts

Flow Chart Symbols

Name	**Symbol**	**Use in flowchart**
Oval		Denotes the beginning or end of the program
Parallelogram		Denotes an input
Rectangle		Denotes a process to be carried out (e.g. addition, subtraction etc.)
Diamond		Denotes a decision (or branch) to be made. The program should continue along one of two routes. (e.g. IF/THEN/ELSE)
Hybrid		Denotes and output operation
Flow Line		Denotes the direction of logic flow in the program

The Flowchart

- **Dictionary Definition:** A schematic representation of a sequence of operation, as in a manufacturing process or computer program
- **Technical Definition:** A graphical representation of the sequence of operations in an information system or program
  - **Information system flowcharts:** show how data flows from source documents through the computer to final distribution to the users
  - **Program flowcharts:** show the sequence of instructions in a single program or subroutine
- Different symbols are used to draw each type of flowchart

The Flowchart

- A flowchart
  - Shows logic of an algorithm
  - Emphasizes individual steps and their interconnections

  - E.g. control flow from one action to another

if Selection Structure

Selection structure

- Choose among alternative courses of action

Pseudocode example:

```
If student's grade is greater than or equal to 50
    Print "Passed"
```

- If the condition is ***true***
  - Print statement executed, program continues to next statement
- If the condition is ***false***
  - Print statement ignored, program continues
- Indenting makes programs easier to read

if Selection Structure

Translation into Algorithm

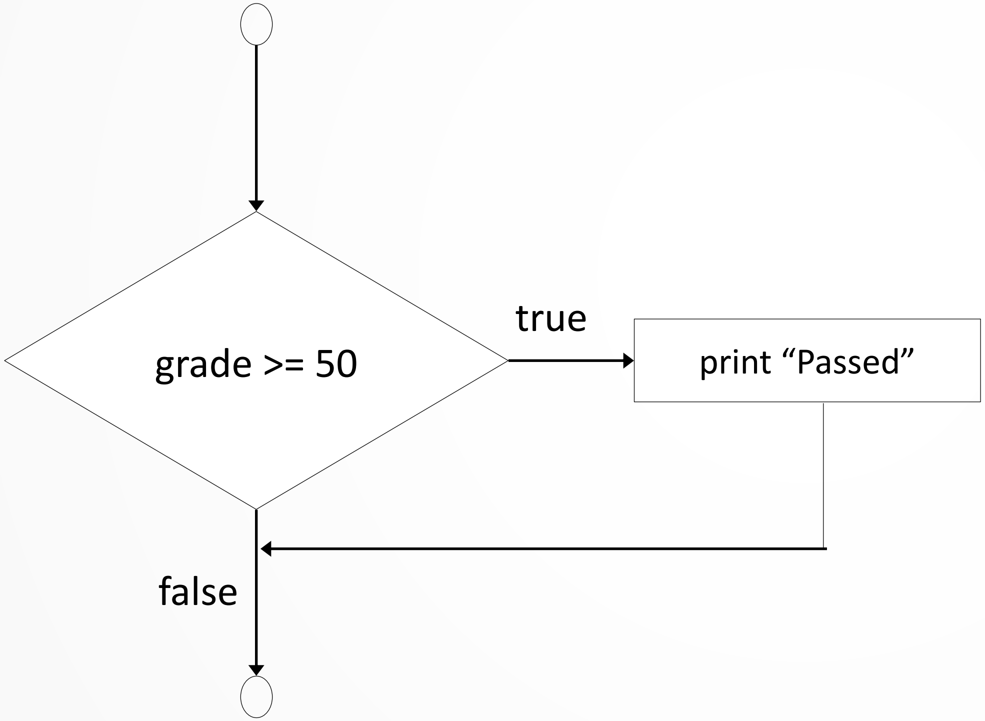
```
If student's grade is greater than or equal to 60
    Print "Passed"
```

```
if ( grade >= 50 )
    print "Passed";
```

- **Diamond symbol (decision symbol)**
  - Indicates decision is to be made
  - Contains an expression that can be true or false
    - Test condition, follow path
- **if structure**
  - Single-entry/single-exit

if Selection Structure

- Flowchart of pseudocode statement



A decision can be made on any expression.

zero - **`false`**

nonzero - **`true`**

Example:

`3 - 4` is **`true`**

if/else Selection Structure

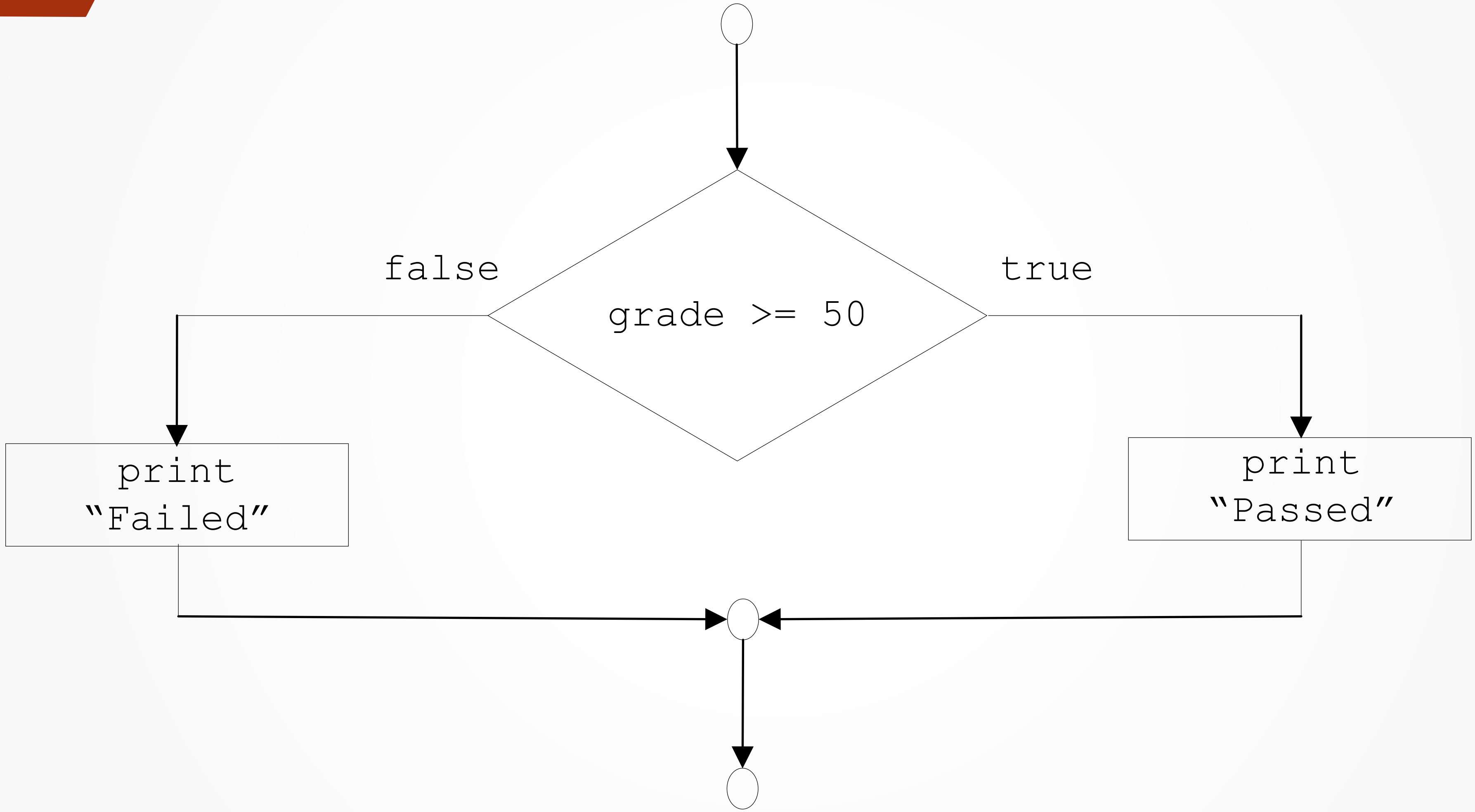
- if
  - Performs action if condition true
- if/else
  - Different actions if conditions true or false

Pseudocode

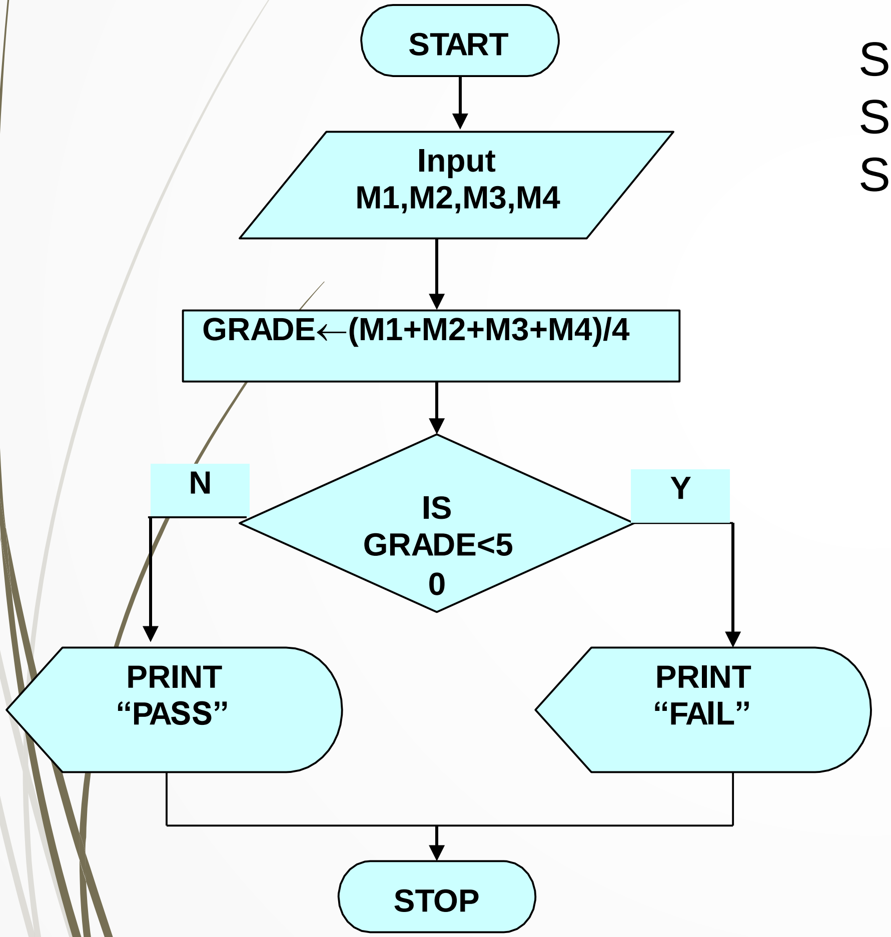
```
if student's grade is greater than or equal to 40
    print "Passed"
else
    print "Failed"
```

```
if ( grade >= 50 )
    Print "Passed";
else
    Print "Failed";
```

if/else Selection Structure



Example



```
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
```

Questions