

# Problem Solving

## PSEUDOCODE & ALGORITHM

by  
Noor Azida Binti Sahabudin  
Faculty of Computer Systems & Software Engineering  
[azida@ump.edu.my](mailto:azida@ump.edu.my)



OER Problem Solving by Noor Azida Binti Sahabudin work is under licensed  
[Creative Commons Attribution-NonCommercial-NoDerivatives 4.0  
International License](https://creativecommons.org/licenses/by-nc-nd/4.0/).

# Chapter Description

- **Expected Outcomes**

- Students able to use **pseudocode** and **algorithm** to develop the instruction for each module in the solution of a problem

- **References**

- Sprankle, M., and Hubbard, J., (2012). Problem Solving and Programming Concepts : 9th Edition. Prentice Hall, 2012. ISBN : 0132492644
- Retrieve from:  
<http://userpages.wittenberg.edu/bshelburne/Comp150/Algorithms.htm>

# What is Pseudocode?

**List of instruction to solve tasks**

**Informal way to design of a computer program /  
algorithm**



# 3 Standards Rules to Produce Pseudocode

↵ Each instruction have number to enforce ordered sequence of operations (with subordinate. Example 3, 3.1, 3.2 ... 4)

**1**

≈ Each instruction should be clear

**2**

∞ Completeness to ensure nothing is left out

**3**

# Example 1: Pseudocode

**Computing Sales Tax** : Pseudo-code the task of computing the final price of an item after figuring in sales tax.

Note the three types of instructions: input (**read**), process/calculate (=) and output (**print**)

1. Start
2. **read** price of item
3. **read** sales tax rate
4. sales tax = price of item times sales tax rate
5. total price = price of item plus sales tax
6. **print** final price
7. End

Variables: price of item, sales tax rate, sales tax, total price

# Example 2: Pseudocode

**Computing Salary:** Salary pay depends on the pay rate and the number of hours worked per week. However, if you work more than 30 hours, you get paid time-and-a-half for all hours worked over 30.

Pseudo-code the task of computing gross pay given pay rate and hours worked.

1. Start
2. **read** hours worked
3. **read** pay rate
4. **if** hours worked  $\leq 30$  then
  - 4.1 salary = pay rate times hours worked
5. **else**
  - 5.1 salary = pay rate times 40 plus 1.5 times pay rate times (hours worked minus 40)
6. **print** salary
7. End

Variables: hours worked, pay rate, salary

Retrieve from: <http://userpages.wittenberg.edu/bshelburne/Comp150/Algorithms.htm>

# What is Algorithm?

An ordered  
sequence

Each instruction  
is clear, do-able,  
can be done  
without difficulty

Performs some  
task

Can be executed  
by a computing  
agent which is  
not necessarily a  
computer

# Properties of an Algorithm

## Input

- Algorithm need to has zero or more inputs from a set of objects

## Output

- Algorithm must has one or more outputs that related to the input

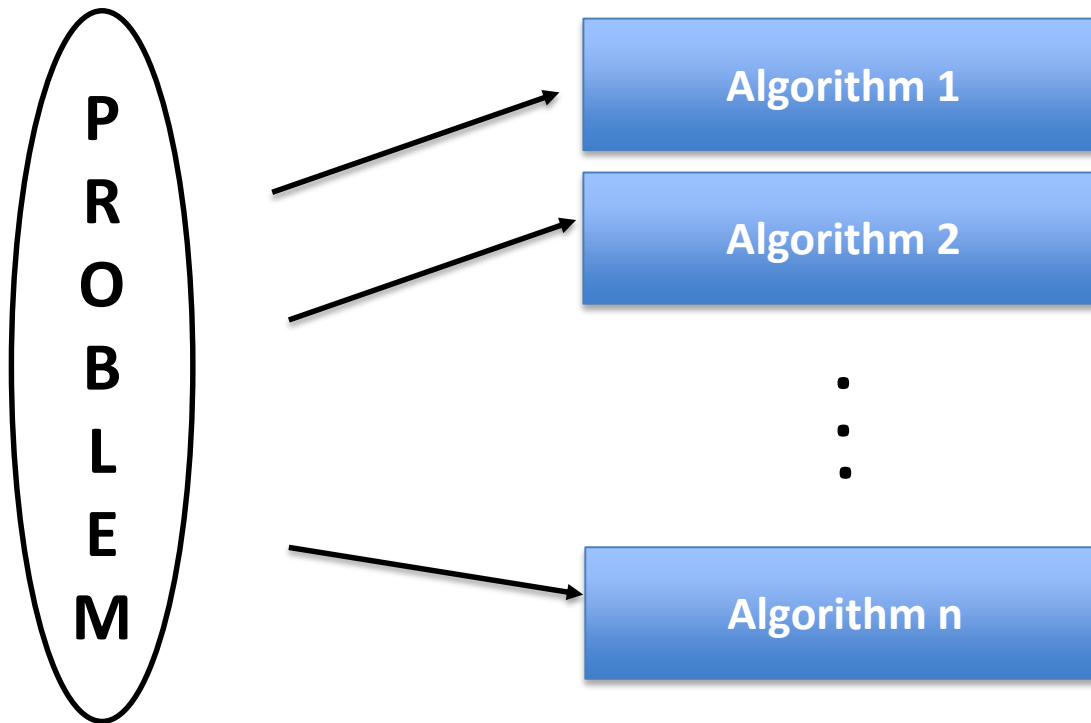
## Effectiveness

- All operations must sufficiently can be done



# Problem VS Algorithm

Each problem can have many algorithm



# Common Elements of Algorithms

## Need data/input

- Values from external source
- Algorithm require data to define problem

## Involve calculation

- Arithmetic computation
- Comparison
- Testing logical condition

## Result / output

- Reported results to the user

# Category of Algorithms Operations

## Sequential

- Execute instructions in order

## Selection

- Choosing among two or more action based on input data
- asks a true/false question

## Iteration / looping

- Execute repeated a set of instruction until logical condition holds

# Example 1: Algorithm in Daily Life

## Problem: Make a cake

Put five eggs in a bowl

Add a cup of sugar

Stir

Add a cup of buttermilk

Add a teaspoon of vanilla extract

Add two cups of flour

Add two spoon of coco powder

Mixed the ingredient

Put into a cake container

Put into oven

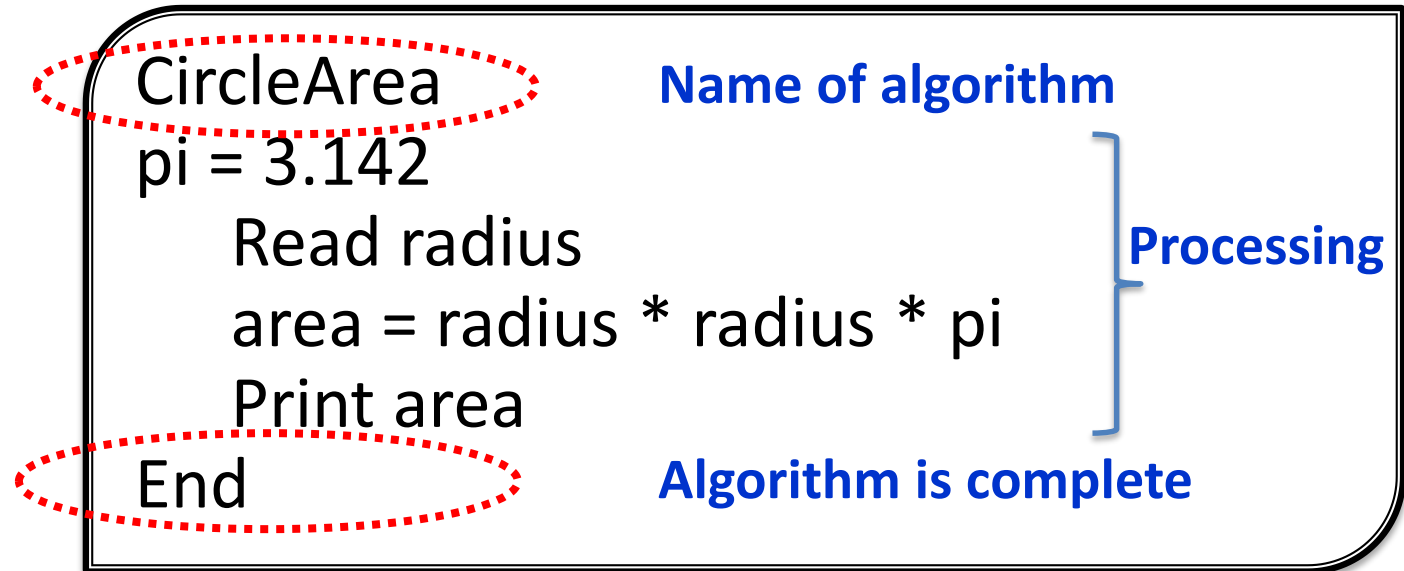
Set the temperature

Set the timer



# Example 1: Algorithm

Problem: Find an area of a circle.



## Example 2: Algorithm

**Problem:** Determine students “Fail” or “Pass” based on their mark. Mark less than 40 means the student fail, otherwise is passed.

Grade

    Read mark


    If mark < 40

        Print “Fail”

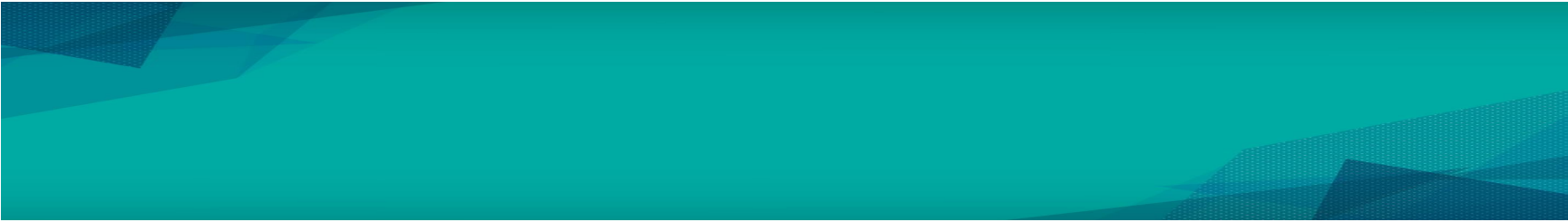
    Else

        Print “Passed”

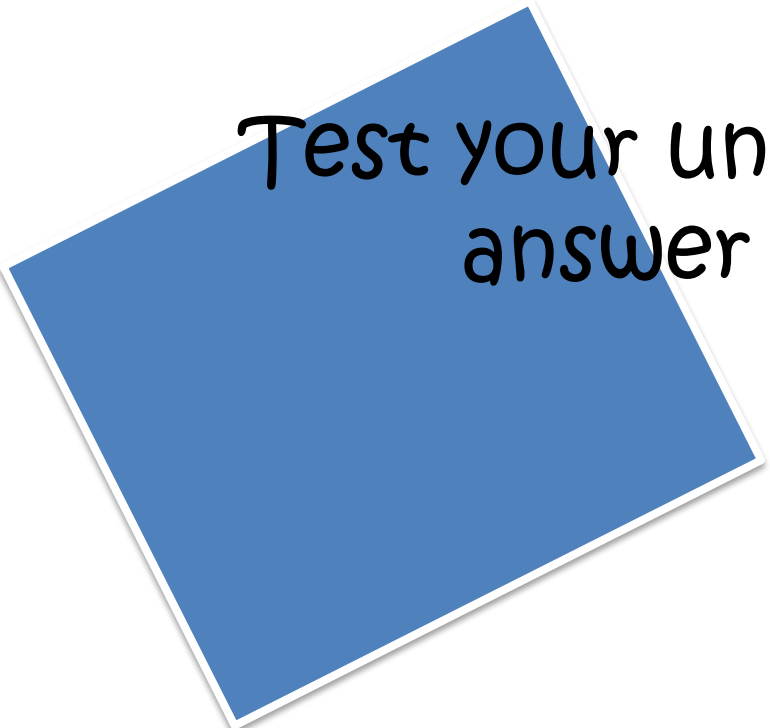
End



Once you have finished  
developing your algorithm, you  
need to check your solution  
algorithm (DESK CHECKING – for  
more info refer Chapter 5)



Test your understanding by  
answer Tutorial 3





# Conclusion / What we have learn today?



What is Pseudocode?



Standard rules of Pseudocode



What is algorithm?



Elements of algorithm



Algorithm operations

# Author Information

NOOR AZIDA BINTI SAHABUDIN

Senior Lecturer

Faculty of Computer Systems & Software Engineering

Universiti Malaysia Pahang

PhD in Educational Technology