Program Analysis and Design

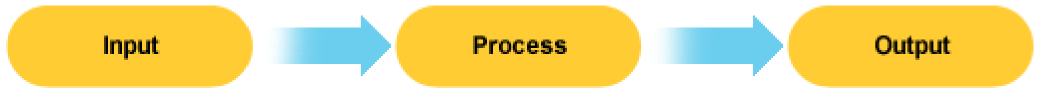
College of Arts, Media and Technology
Chiang Mai University

Application

- A computer software designed to help the user to perform specific tasks
- A Task is defined as a problem
- To solve the problem
  - The solution must be designed

IPO pattern

- The most fundamental design pattern
- Separate program into 3 parts
- Each program consists of
  - Input
  - Process
  - Output



Input

- Provide the input information
- Can be
  - Set from the program
  - Received from the user
  - Archived from storage
- Use for preparing a set of data for process

Process

- A solution part
- Receive the input to calculating the result
- Other variables may be introduced for solution
- Programming structure
  - Sequential programming
  - Selection programming
  - Iterative programming
- Design document required for ease the programming

Output

- Receive the result from the process
- The output may be one output or more
- Formatting the output for user to read

Why IPO ?

- Concern separation
- Focus only the part which it responds
- Make code easier to understand

The Design Document

- Searching for the solution
- Write the solution for other can understand
  - Use to validate your solution
    - By other person
- Technique
  - Natural language
  - Formatted description
  - Diagram

Natural Language

- Simple text explanation
- Free hand writing
- Describe as what authors want
- Programmers required skills to transform from description to program

Formatted Text

- Templates for document are prepared
  - Blank to be filled
  - Conventional writing
  - Formal language
    - Mathematic expression
- Easier for programmers to understand
  - The text should be in the proper format
- The flow of the program could not be seen clearly
- Example
  - Psudo code
  - CafeOBJ

Diagram

- One picture is equal to thousands word
- Use the picture for explain the program
- Easy to see the whole system
- Example
  - UML diagram
  - Flow chart

What we focus ?

- Pseudo code
  - The formatted text to explain the program
  - Implements structured concept
  - Some complex algorithms explained by pseudo code
- Flow chart
  - Visualize the concept
  - Easy to understand (in the complex system)
  - The fundamental knowledge for UML diagram

College of Arts, Media and Technology
Chiang Mai University

Pseudocode

- A mixture of English and formatting to make the step in an algorithm
- A way of expressing algorithms that uses a mixture of *English phrases* and *indention* to make the steps in the solution explicit
- No grammar rules in pseudocode
- Not case sensitive

Rules for Pseudocode

- Write only one statement per line
- Capitalize initial keyword
- Indent to show hierarchy
- End multiline structures
- Keep statements language independent

One statement Per Line

- Each statement expresses one action for computer
- Each task will correspond to one line of pseudo code

Task List

Read name, hours worked, rate of pay

Perform calculations

    gross = hours worked * rate of pay

Write name, hours worked, gross

Pseudocode

```
READ name, hoursWorked, payRate

gross = hoursWorked * payRate

WRITE name, hoursWorked, gross
```

Capitalize Initial Keyword

- Keyword such as
  - READ, WRITE (I/O process)
  - IF, ELSE, ENDIF (Selection process)
  - WHILE, ENDWHILE (Repetition process)

Pseudocode

```
READ name, hoursWorked, payRate
gross = hoursWorked * payRate
WRITE name, hoursWorked, gross
```

Indent to show Hierarchy

- Each design structure uses a particular indentation pattern
- Sequence:
  - Keep statements in sequence all starting in the same column
- Selection:
  - Indent statements that fall inside selection structure, but not the keywords that form the selection

```
READ name, grossPay, taxes
IF taxes > 0
      net = grossPay – taxes
ELSE
      net = grossPay
ENDIF
WRITE name, net
```

- Loop:
  - Indent statements that fall inside the loop but not keywords that form the loop

End Multiline Structures

```
READ name, grossPay, taxes
IF taxes > 0
    net = grossPay – taxes
ELSE
    net = grossPay
ENDIF
WRITE name, net
```

- `ENDIF` used for end multiline of `IF`
- The same applies for `WHILE/ENDWHILE`

Language Independence

- Describe a logic plan to develop a program
- Not programming
- The grammar rules for program are not applied

Rules for Variable Names

- Begin with lowercase letter
- Contain no spaces
- Unique names within code
- Consistent use of names

Working with Fields

Calculations

+	**add**
-	**subtract**
*****	**multiply**
/	**divide**
**** or ^**	**exponentiation**
( )	**grouping**

Selection

>	greater than
<	less than
=	equal to
>=	greater than or equal to
<=	less than or equal to
<>	not equal to

CAMT
College of Arts, Media and Technology
Chiang Mai University

Pseudo code Trick

- Separate the part of pseudo code as input (READ), process (the statements), and output(WRITE)
- No variable declaration required
    - Leave it for the programmer to select the proper type
    - Can be defined if you required the specific data type
- The decoration of output
    - Programmers have to decorate the output themselves
    - Do anything to make the output meaningful to the users

Computer Basic Operations

- Receive information
  - PROMPT instruction
    - For waiting for user input
  - GET instruction to read input from user to the variable

Example pseudocode

```
PROMPT FOR studentMark
GET studentMark
```

CAMT
College of Arts, Media and Technology
Chiang Mai University

Computer Basic Operations

- Put out the information
  - PRINT
    - Send output to printer
  - WRITE
    - Send output to file
  - PUT, OUTPUT, DISPLAY
    - Send to screen

Example pseudocode

```
PRINT 'Program Completed'
WRITE customer record TO master file
OUTPUT total tax
DISPLAY 'End of data'
```

CAMT
College of Arts, Media and Technology
Chiang Mai University

Computer Basic Operations

- Perform arithmetic
  - To be consistent with high-level programming language

    + for Add   - for Subtract

    * for Multiply   / for Divide   ( ) for Parentheses
  - The order of operations are the same as in normal programming language

Computer Basic Operations

- Assign a value to a variable or memory location
  - Initialization
    - Use INITIALIZE or SET
  - Assign a value
    - Use '=' or '←'
  - To keep a variable for later use
    - Use SAVE or STORE

Example pseudocode

```
INITIALIZE total_price TO zero
SET student_count TO zero
Total_price = cost_price + sales_tax
Total_price ← cost_price + sales_tax
STORE customer_num IN last_customer_num
```

CAMT
College of Arts, Media and Technology
Chiang Mai University

Activity

- Group of 3 people
- The given pseudo code is a module for calculating the land price

```
READ width, height, pricePerSqureMeter
area = width*height
price = area*pricePerSqureMeter
DISPLAY area, price
```

- What are the results if given the following dataset

width	height	pricePerSquareMeter
5	5	10
2	0.5	2
3	1	1

CAMT
College of Arts, Media and Technology
Chiang Mai University