

# CP020001 Computer Programming

Lecture: Flowchart and Pseudocode (Part I)

[teerapong.pa@chula.ac.th](mailto:teerapong.pa@chula.ac.th)

## Reference:

1. <https://colab.research.google.com/notebooks/>
2. <https://problemsolvingwithpython.com/>
3. <https://www.cp.eng.chula.ac.th/books/python101/>
4. <https://www.eng.chula.ac.th/th/20535>
5. <https://towardsdatascience.com/graph-visualisation-basics-with-python-part-i-f-lowcharts-6298c4f412e0>
6. <https://www.programiz.com/article/flowchart-programming>
7. <https://www.toppr.com/guides/python-guide/tutorials/python-flow-control/if-elif-else/python-if-if-else-if-elif-else-and-nested-if-statement/>

# About Me



Kao  
Panboonyuen

kao-panboonyuen 

AI Research Scientist

**Name:** Teerapong Panboonyuen (P'Kao)

**Contact:** [teerapong.pa@chula.ac.th](mailto:teerapong.pa@chula.ac.th)

[panboonyuen.kao@gmail.com](mailto:panboonyuen.kao@gmail.com)

**Education:** Ph.D. (AI) Chula

**Position:** AI Team Lead, MARS

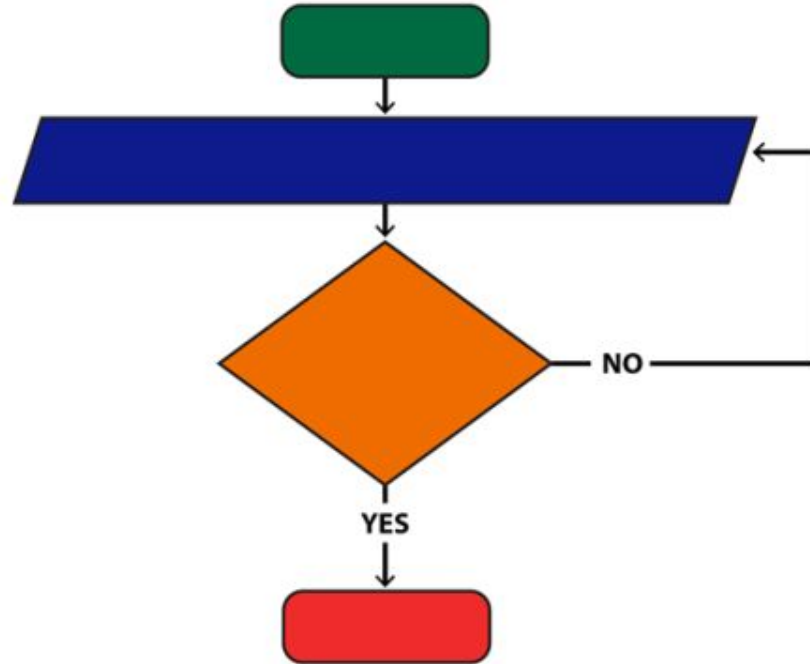
PostDoc, Chula

**Interests:** Computer Vision, Deep Learning

Machine Learning, Remote Sensing

# Flowcharts and Programming

## learning strategies and tips



# Introduce Flowchart Shapes

Use an oval to mark the beginning and end of the program.



# Introduce Flowchart Shapes

Use a parallelogram to show input or output. Input could be the user entering information. Output could be a message on the screen.



# Introduce Flowchart Shapes

Use a rectangle to process an action. It could be a formula to calculate a value, or it might be a step that modifies text.



# Introduce Flowchart Shapes

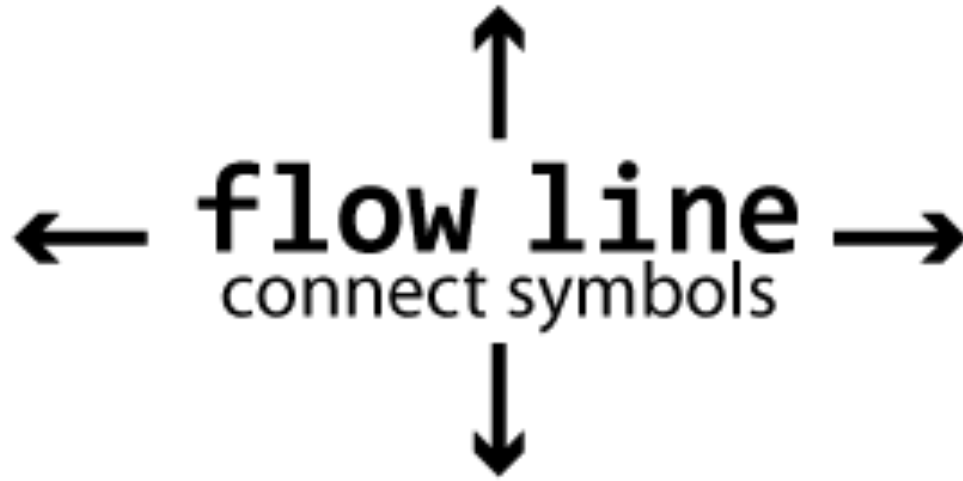
Use a diamond to make decisions. This shape will have two or more lines that come from it – one for each outcome. This step might ask a question or provide options. The result could be true or false, yes or no, or choices (red, blue, or green).





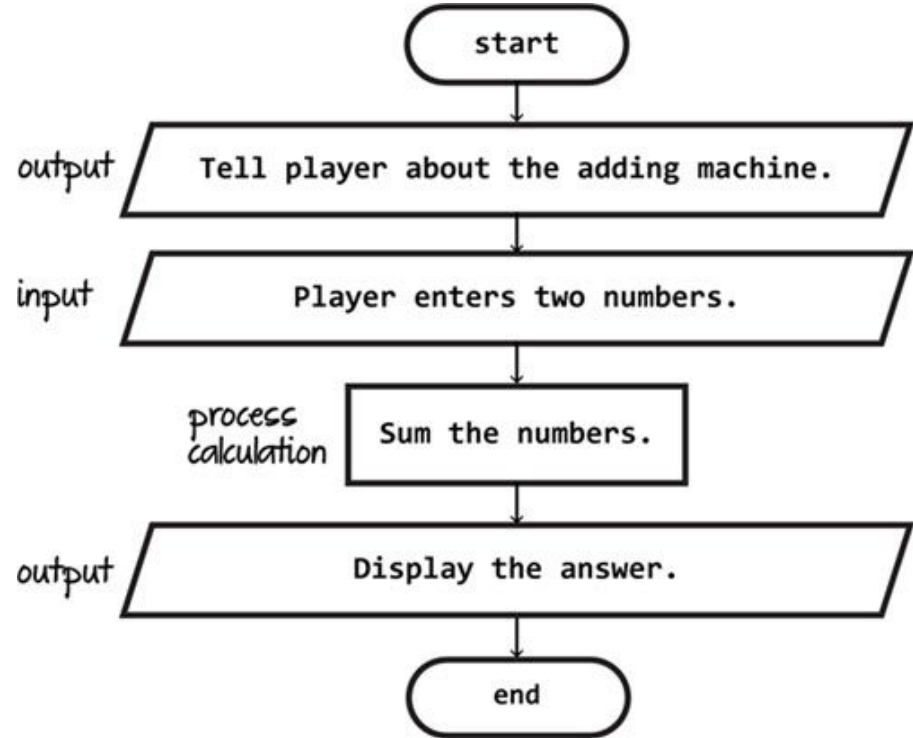
# Introduce Flowchart Shapes

Use lines to connect the shapes. The arrows show the direction of the steps.  
Some lines should include labels, such as yes or no, to explain what is happening.



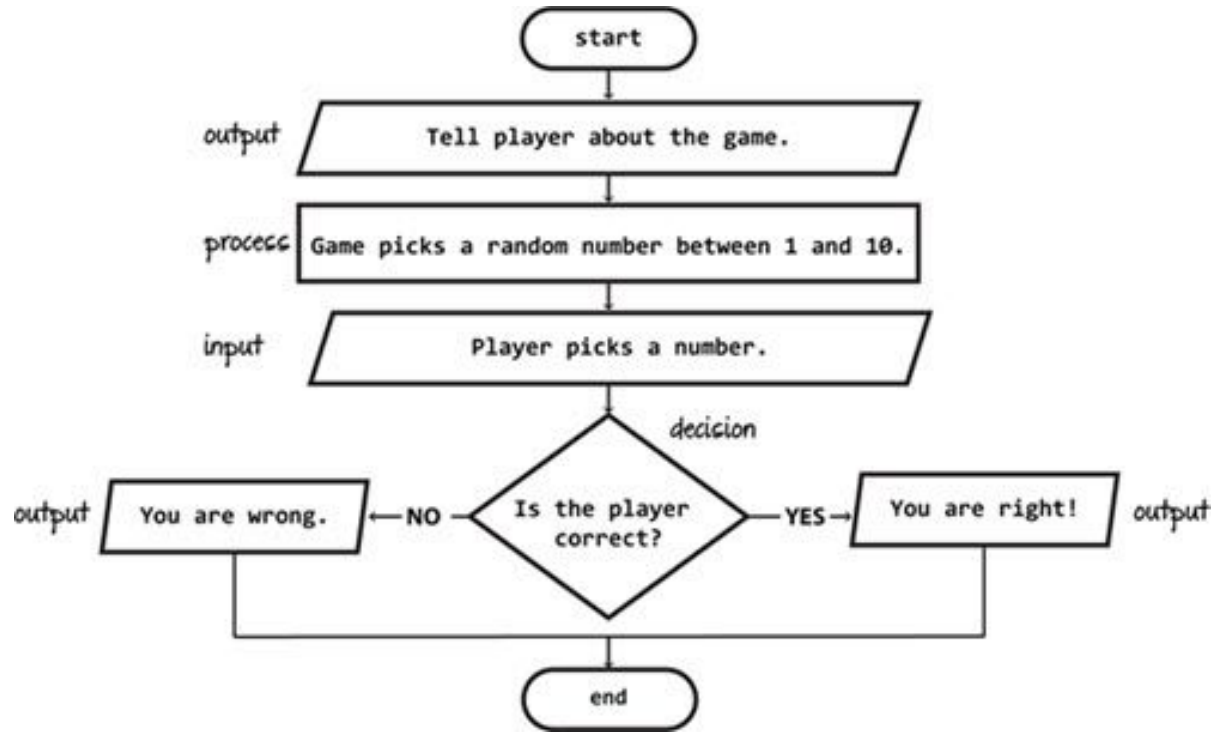
# Start Simple with a Familiar Task

**A calculator** is something familiar. It is on every phone and computer. Since it is recognizable, student can transfer their existing knowledge to the task. This enables them to focus their attention towards organizing the program components into a flowchart.



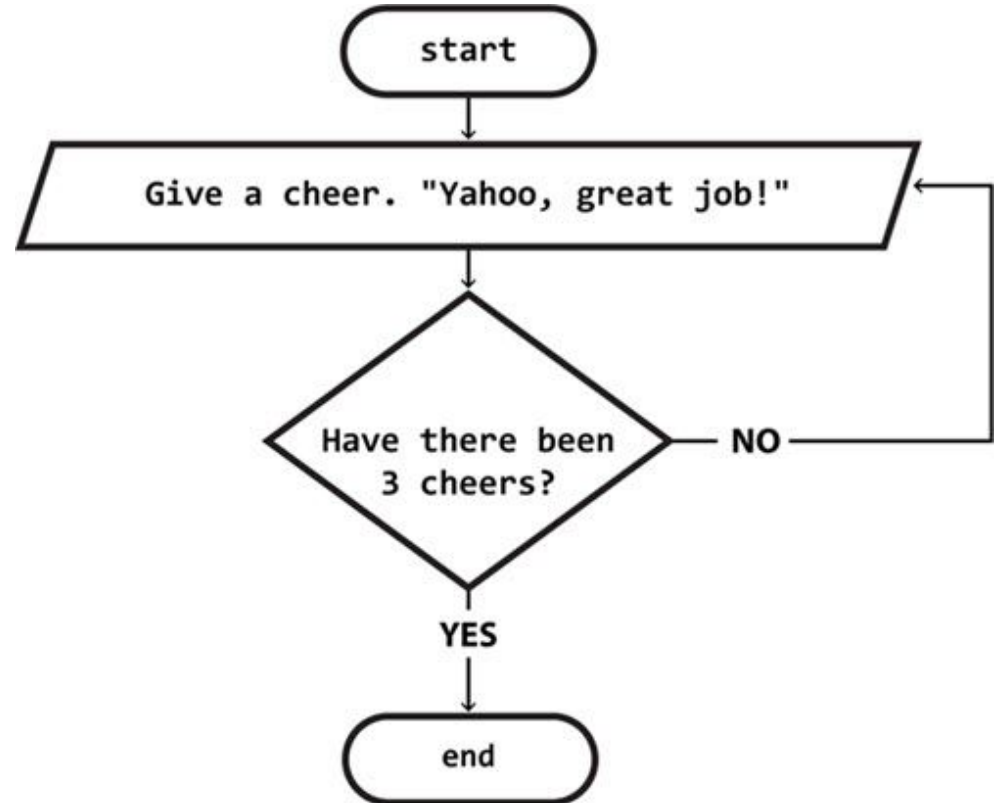
# Decision Making Sample: A guessing game.

Players must correctly pick a number between 1 and 10 to win. Although this program only has a few steps, it uses all the standard shapes in a flowchart and has branches.



# Apply Pattern Recognition to Loop Instructions

A simple cheer is a great place to start making loops in a flowchart. Now that students understand the meaning of the standard shapes and branches, they can focus upon the direction of the lines. Since only the output is repeated, there is a simplicity to the task that is ideal for beginners.



# Intermediate & Advanced Flowchart Symbols

## Document Symbols

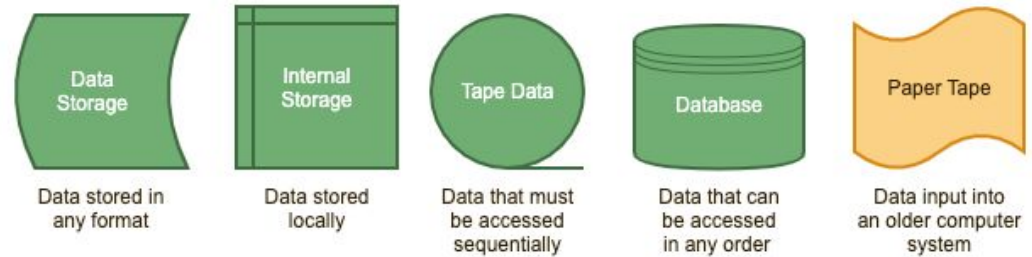
Single and multiple document icons show that there are additional points of reference involved in your flowchart. You might use these to indicate items like “create an invoice” or “review testing paperwork.”



# Intermediate & Advanced Flowchart Symbols

## Data Symbols

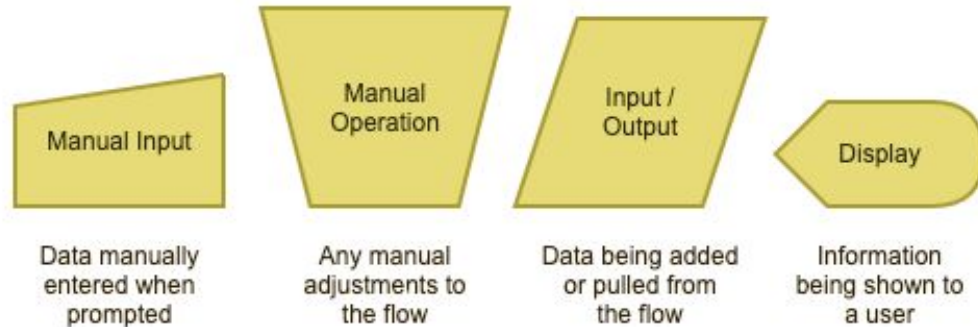
Data symbols clarify where the data your flowchart references is being stored. (You probably won't use the paper tape symbol, but it definitely came in handy back in the day.)



# Intermediate & Advanced Flowchart Symbols

## Input & Output Symbols

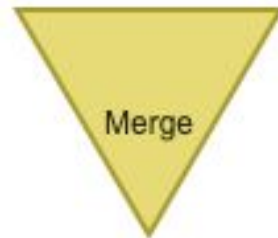
Input and output symbols show where and how data is coming in and out throughout your process.



# Intermediate & Advanced Flowchart Symbols

## Merging & Connecting Symbols

Agreed-upon merging and connector symbols make it easier to connect flowcharts that span multiple pages.



Point where  
separate processes  
join together



Indicates that  
flow continues on  
a new page



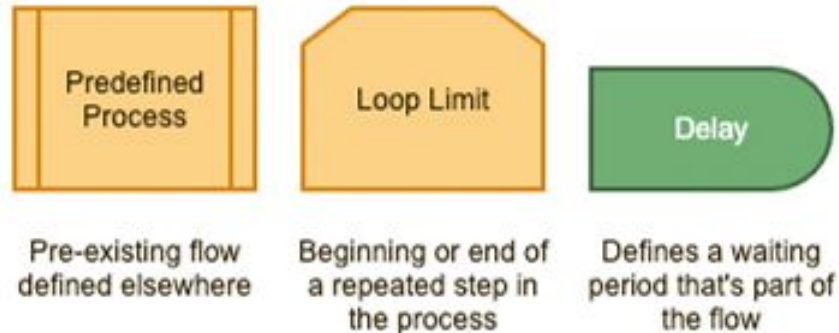
Used to show flow  
across multiple  
charts or pages



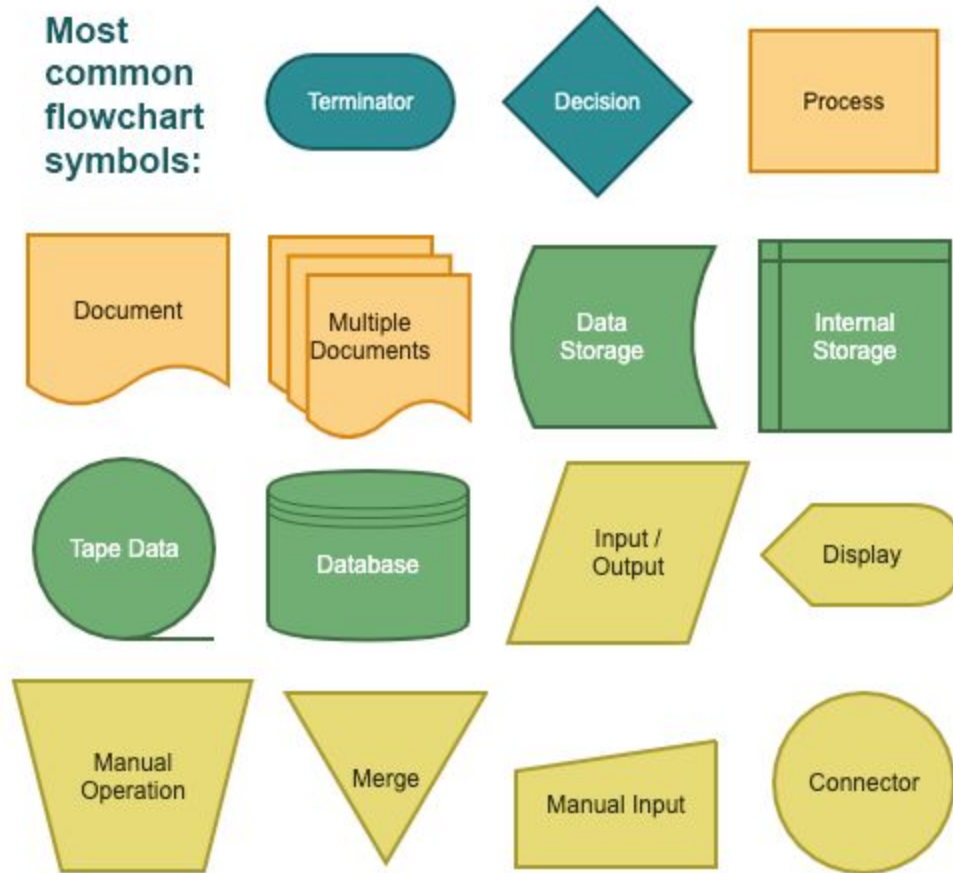
# Intermediate & Advanced Flowchart Symbols

## Additional Useful Flowchart Symbols

The above are a few additional symbols that prove your flowcharting prowess when put to good use.



**Most  
common  
flowchart  
symbols:**



start

stop

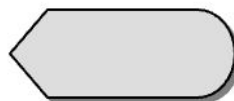
เริ่มต้น/สิ้นสุด



process



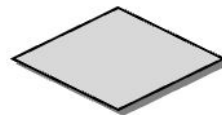
รับข้อมูลจากแป้นพิมพ์



แสดงผลออกทาง  
จอภาพ



เส้นทาง



decision

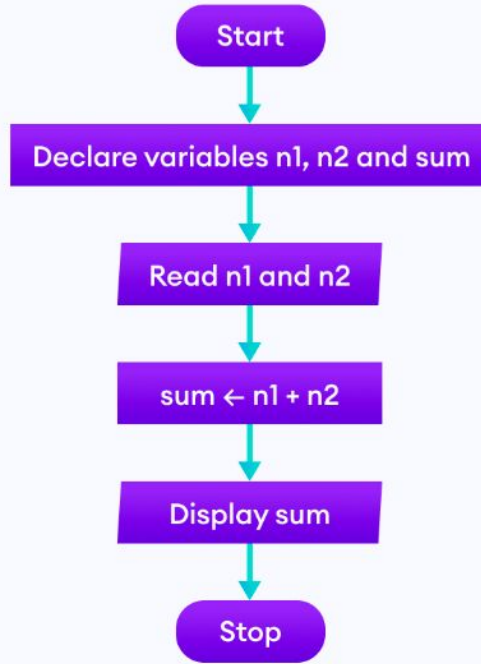


จุดเชื่อม

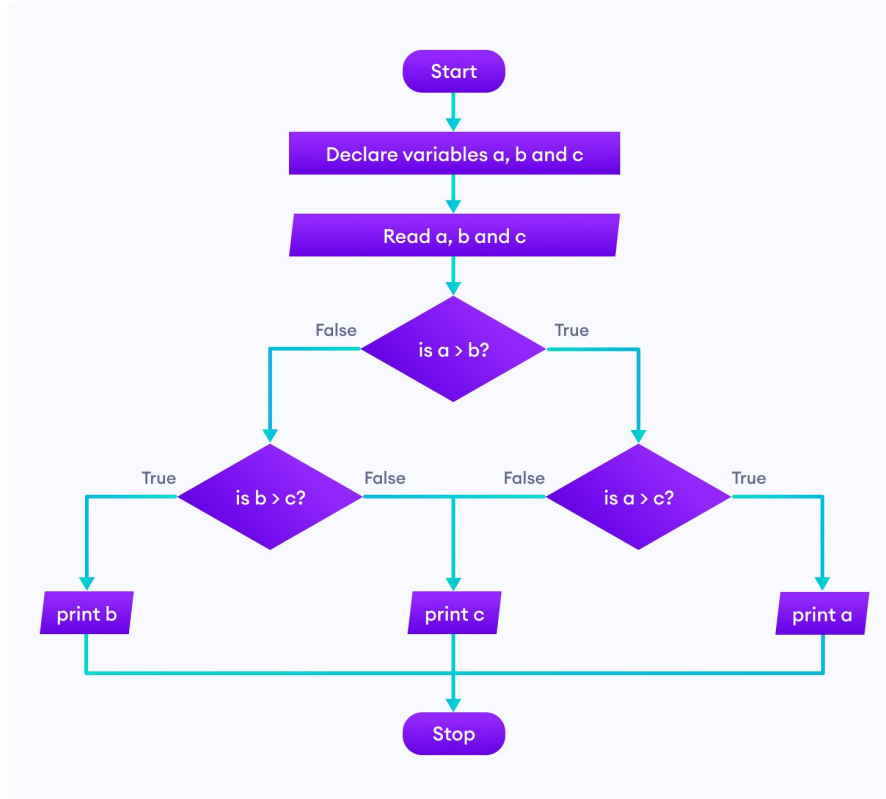


input/output

# Examples of flowcharts in programming



Find the largest among three different numbers entered by the user.



# What are swimlane diagrams?

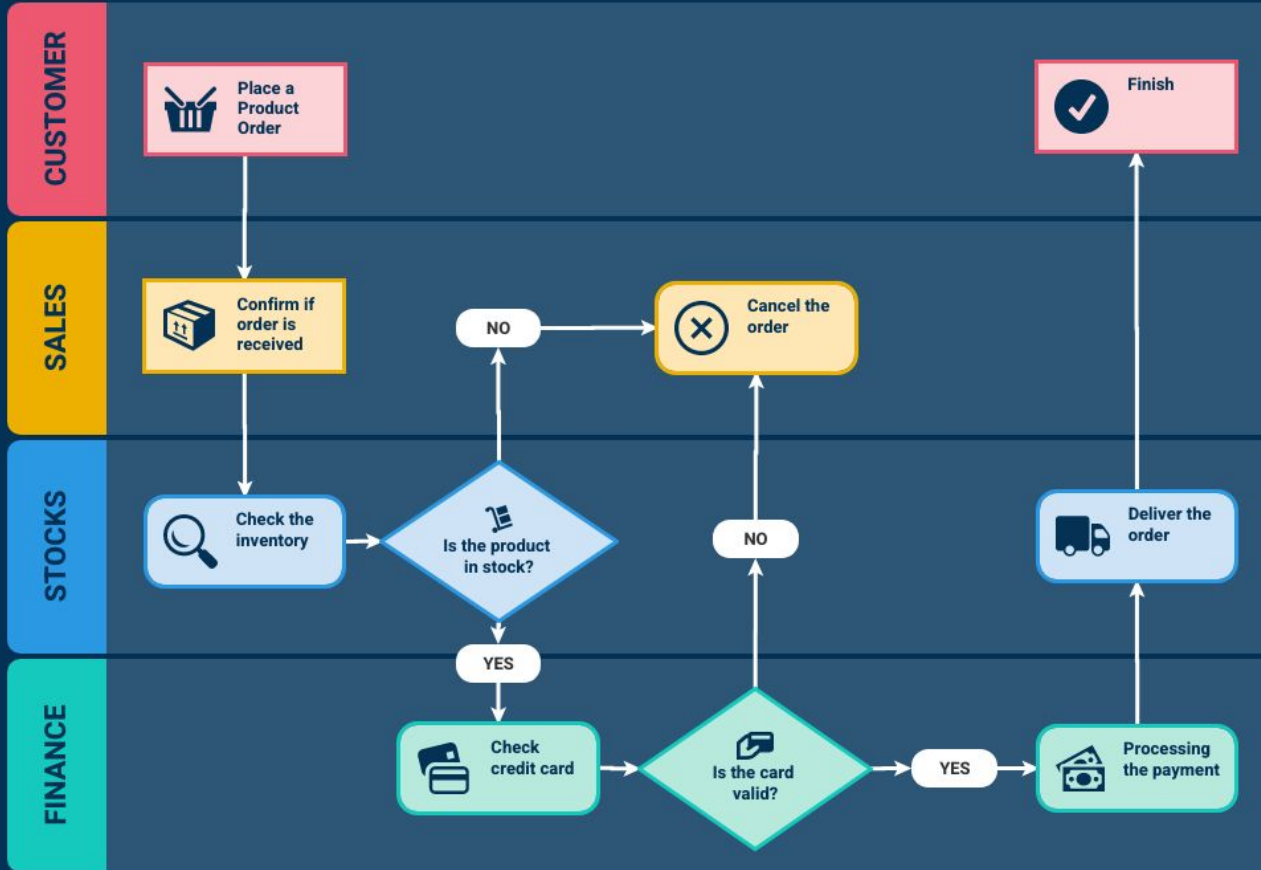
A swimlane diagram is a type of flowchart that **delineates who does what in a process**. Using the metaphor of lanes in a pool, a swimlane diagram provides clarity and accountability by placing process steps within the **horizontal** or **vertical** “swimlanes” of a particular employee, work group or department.

It shows connections, communication and handoffs between these lanes, and it can serve to highlight waste, redundancy and inefficiency in a process.

# Sales Process Swimlane Flowchart



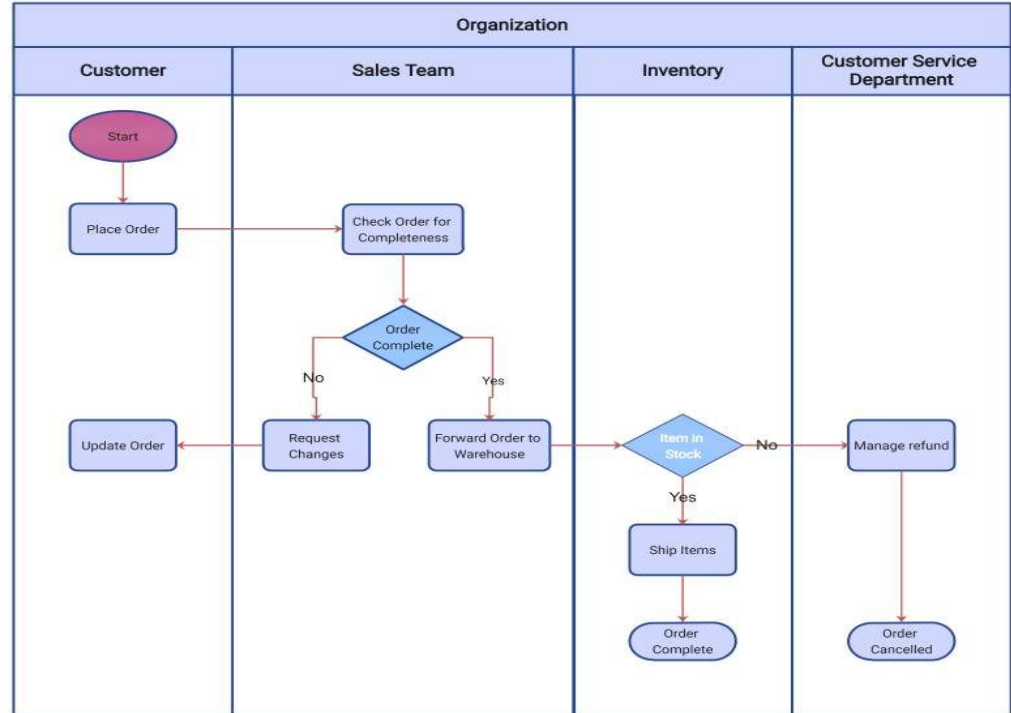
ZOOM RABBIT



# Sample Swimlane Diagram Example

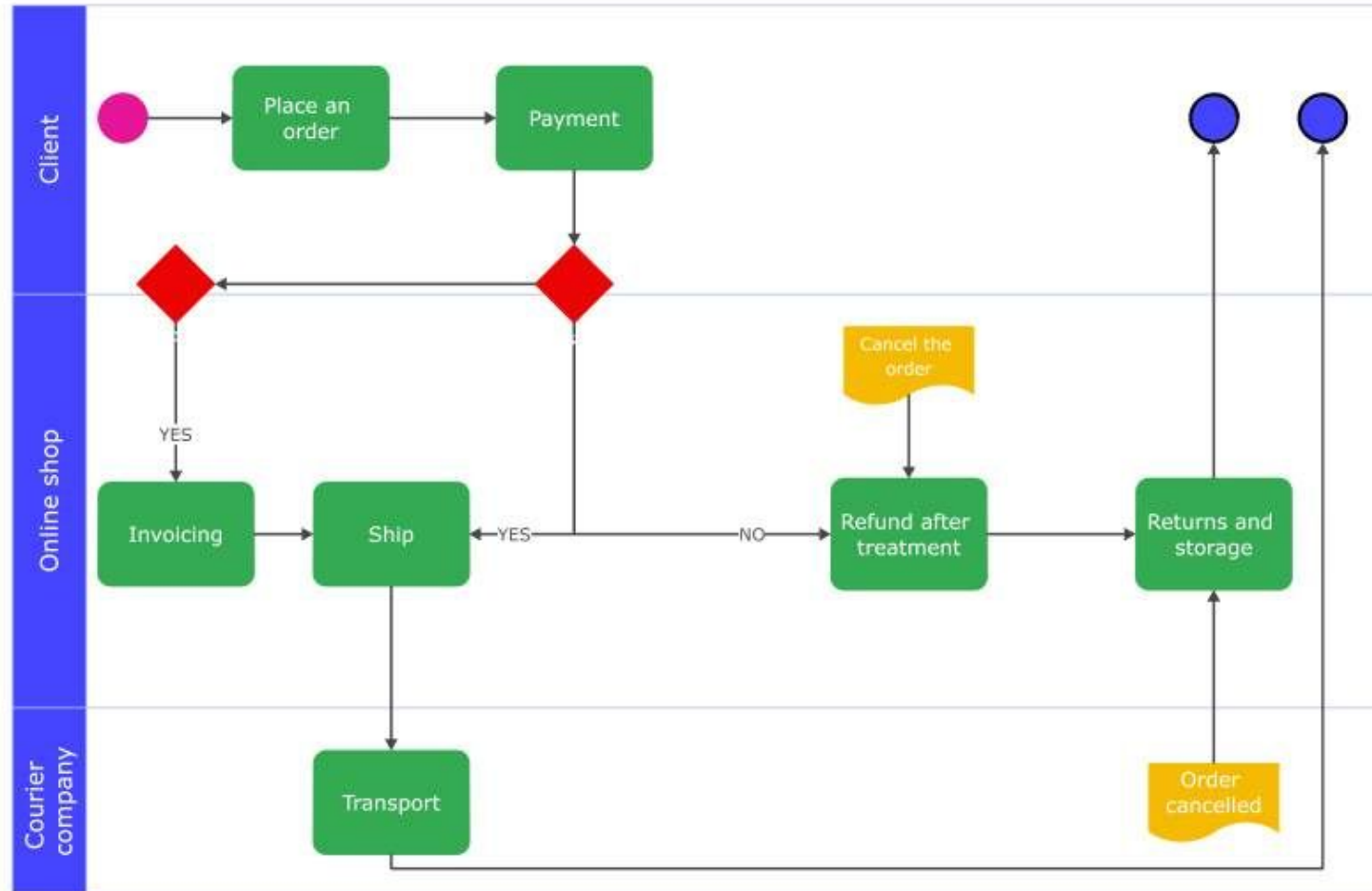
This example is a sample swimlane diagram. It visually represents the sales deployment system. There are four swimlanes in the diagram, and each lane represents a different category. The process flow starts with the customers, moves through various activities, and gets accepted in the third lane or declined in the fourth lane. The criteria for a successful order are also easy to determine with the help of the diagram. If the ordered product is in stock, the system approves; if not, the system declines.

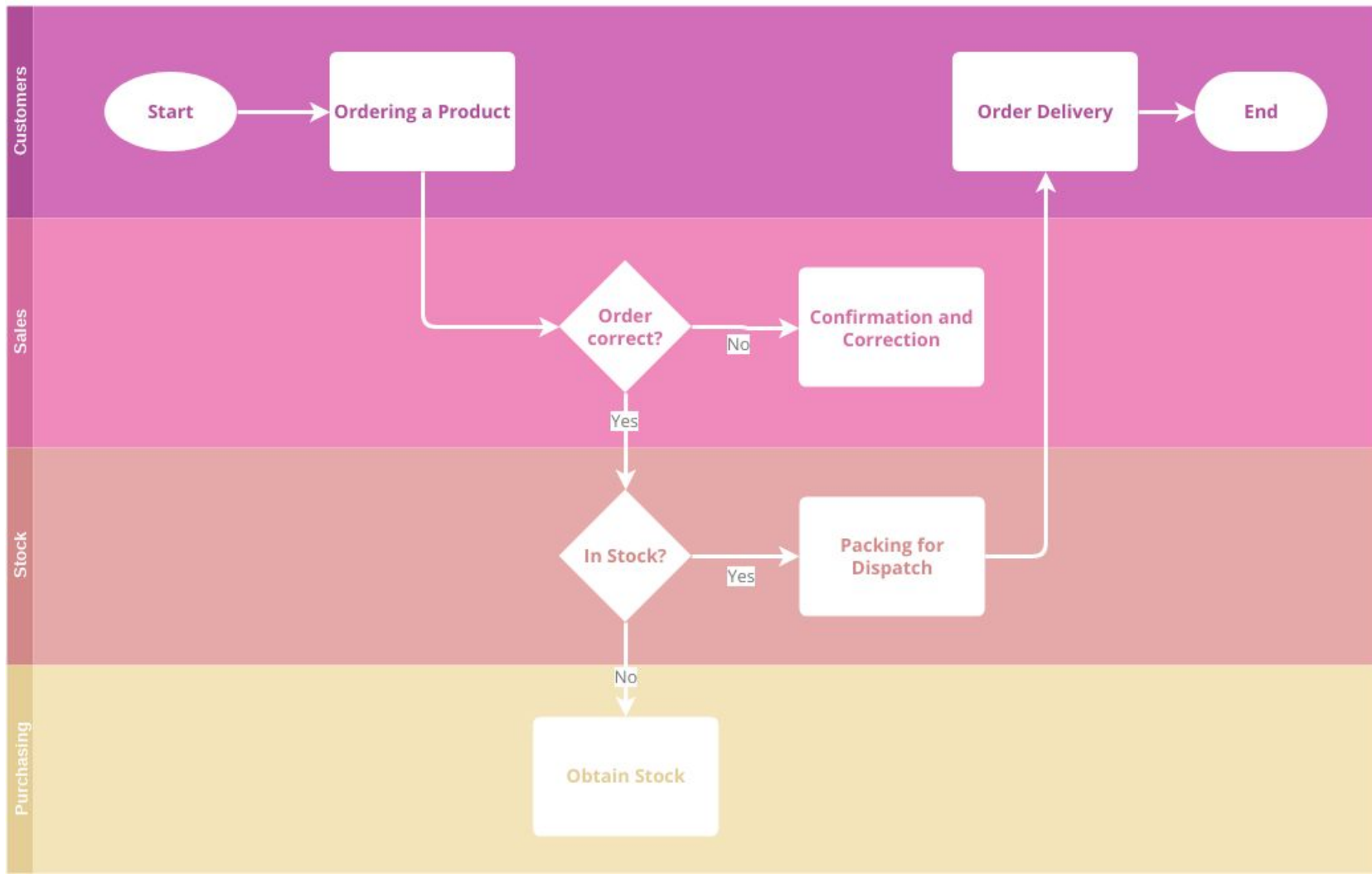
Deployment Flowchart

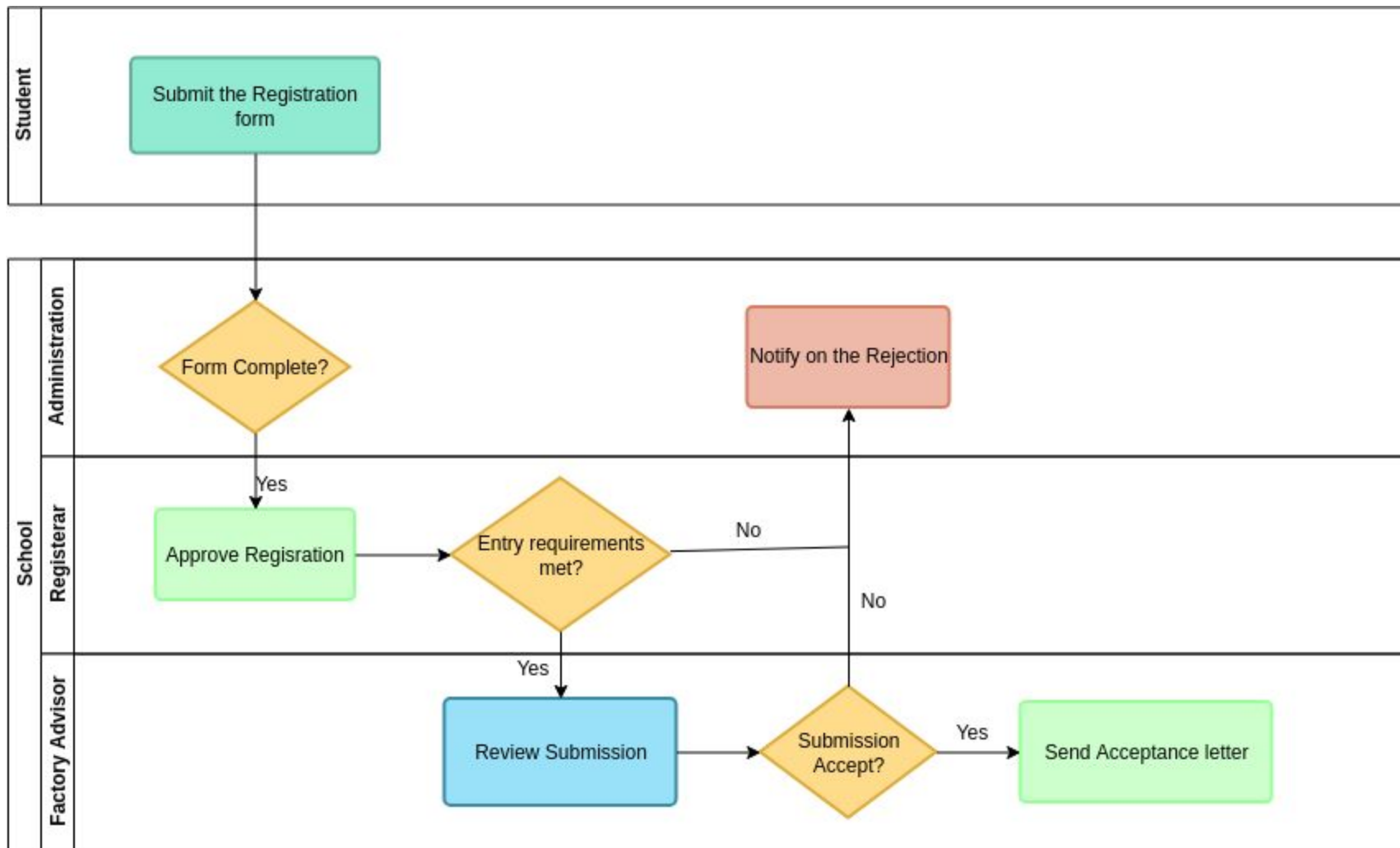




# Online Full Shopping Process







## ขั้นตอนการจัดทำ transcript ปริญญาตรีสำหรับผู้สำเร็จการศึกษา

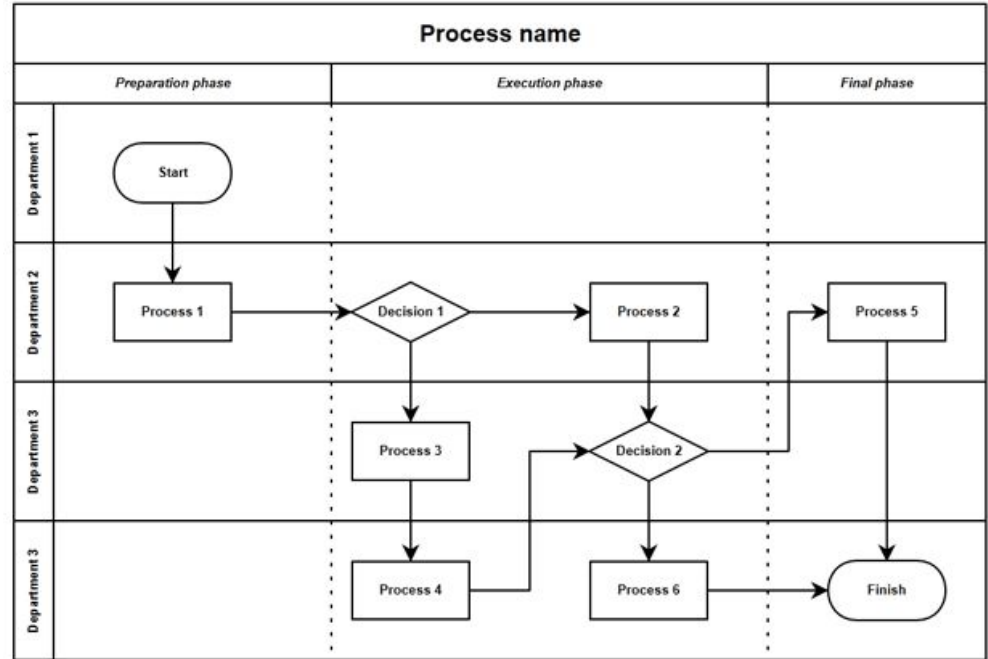
1. ผู้มาติดต่อกรอกข้อมูลคำร้อง และส่งคำร้องให้เจ้าหน้าที่พร้อมรูปถ่าย
2. เจ้าหน้าที่รับคำร้องและตรวจสอบข้อมูล
  - 2.1. เมื่อรับคำร้องแล้ว ตรวจสอบคำร้องว่าดำเนินการกรอกข้อมูลครบถ้วนทุกช่อง
  - 2.2. ตรวจสอบรูปถ่ายสวมชุดครุย ขนาด 1 นิ้ว เท่าจำนวน transcript ที่ขอ
  - 2.3. ตรวจสอบว่าผู้มาติดต่อเขียนชื่อ-นามสกุล รหัสนักศึกษา หลังรูปถ่ายทุกใบ
3. เข้าระบบสารสนเทศเจ้าหน้าที่
4. เปรียบข้อมูลคำร้องกับข้อมูลในระบบสารสนเทศว่าเหมือนกันหรือไม่ โดยตรวจสอบ
  - ชื่อ-นามสกุล ภาษาไทย ชื่อ-นามสกุล ภาษาอังกฤษ
  - วัน/เดือน/ ค.ศ.เกิด
  - วันที่เข้าศึกษา
  - สถานะการเป็นนักศึกษา ถ้าสถานะเป็นสำเร็จการศึกษาแล้ว ตรวจสอบ ภาคเรียนและปีการศึกษาที่สำเร็จการศึกษา การได้รับเกียรตินิยม

หากมีข้อมูลไม่ถูกต้องจะดำเนินการสอบถามกับผู้มาติดต่อเพื่อแก้ไขข้อมูลในส่วนที่ไม่ถูกต้องหากถูกต้องแล้วบันทึก

5. สั่งพิมพ์ TRANSCRIPT
6. ตัดรูป
7. เสนอผู้อำนวยการสำนักทะเบียนฯ ลงลายมือชื่อ
8. ประทับตราสถาบัน
9. จัดเก็บเอกสารรอนำส่ง
10. บันทึกลงในสมุดรับคำร้อง โดยมีรายละเอียดที่ระบุ คือ ชื่อ-นามสกุล คณะ/สาขาวิชา และจำนวนชุดที่ขอ

# Swimlane Techniques

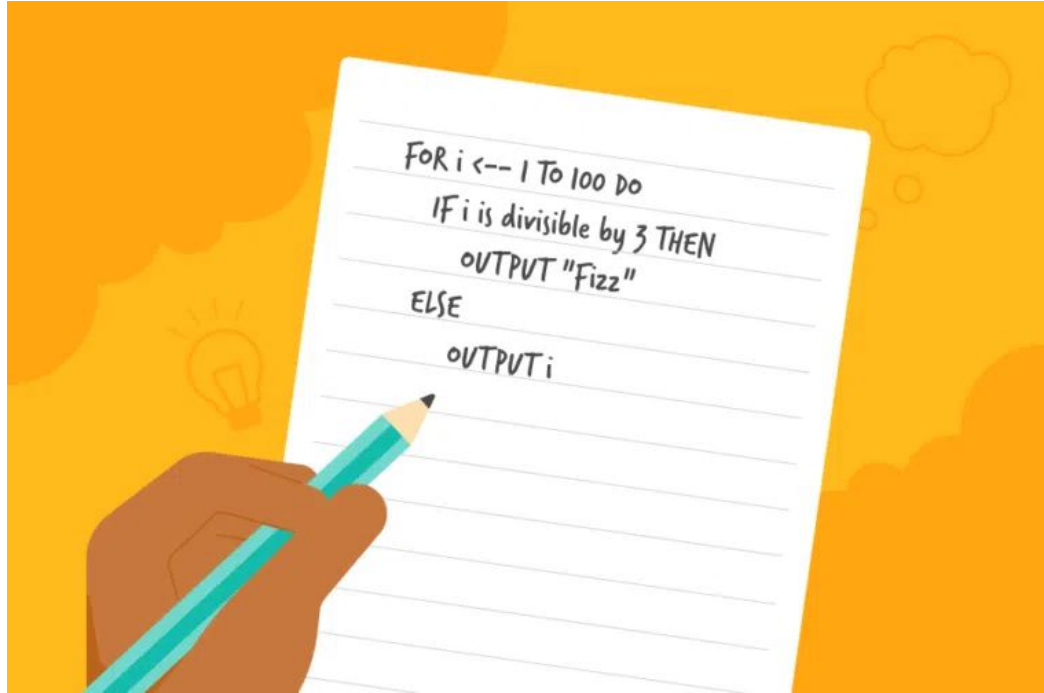
- **Determine the lanes.** Think over how to divide the process into groups to be represented by swim lanes. Draw vertical or horizontal lines, equally spaced on the page, to create the swim lanes;
- **Start your diagram.** Set the start point of the process. Insert a rounded rectangle on the top of a swim lane to visualize start point and name it;
- **Add a connection.** Draw a short arrow if the next action is in the same swim lane. Draw a long arrow if the next action is in the other swim lane;
- **Add a process.** Draw a rectangle and write a brief description inside. Proceed in this manner. Review an article about flowcharts to examine available shapes and their designation;
- **Add connections.** Connect each box with a link to indicate the flow;
- **Verify accuracy.** Consult with the colleagues to verify accuracy.



# Pseudocode

## What is Pseudocode?

- a way of expressing an algorithm **without** conforming to specific syntax rules.
- By learning to read and write pseudocode, you can easily communicate ideas and concepts to other programmers, even though they may be using completely different languages.



## Sample: A prime number

```
number <-- 17
prime <-- TRUE

FOR i <-- 2 TO number
  IF number is divisible by i THEN
    prime <-- FALSE

IF prime = TRUE:
  OUTPUT "prime"
ELSE
  OUTPUT "not prime"
```

Pseudo Code

```
number = 17
prime = True


for i in range(2, number):
    if number % i == 0:
        prime = False
        break

if prime:
    print("prime")
else:
    print("not prime")
```

Python Code

# 1. Pseudocode: Addition of Two Numbers

plaintext

 Copy code


Input two numbers: A, B

Sum = A + B

Display Sum

Python Code:

python

 Copy code

```
A = 5
```

```
B = 7
```


```
Sum = A + B
```

```
print("Sum:", Sum)
```



## 2. Pseudocode: Calculate Area of a Rectangle


plaintext

 Copy code

```
Input length and width of rectangle: Length, Width  
Area = Length * Width  
Display Area
```

Python Code:


python

 Copy code

```
Length = 10  
Width = 5  
Area = Length * Width  
print("Area:", Area)
```

### 3. Pseudocode: Check if a Number is Even or Odd

plaintext

 Copy code

```
Input a number: Number
IF Number modulo 2 equals 0 THEN
    Display "Even"
ELSE
    Display "Odd"
```

Python Code:


python

 Copy code

```
Number = 8
if Number % 2 == 0:
    print("Even")
else:
    print("Odd")
```

## 4. Pseudocode: Calculate Factorial of a Number


plaintext

 Copy code

```
Input a number: N
Factorial = 1
FOR i from 1 to N
    Factorial = Factorial * i
Display Factorial
```

Python Code:

python

 Copy code

```
N = 5
Factorial = 1
for i in range(1, N+1):
    Factorial *= i
print("Factorial:", Factorial)
```