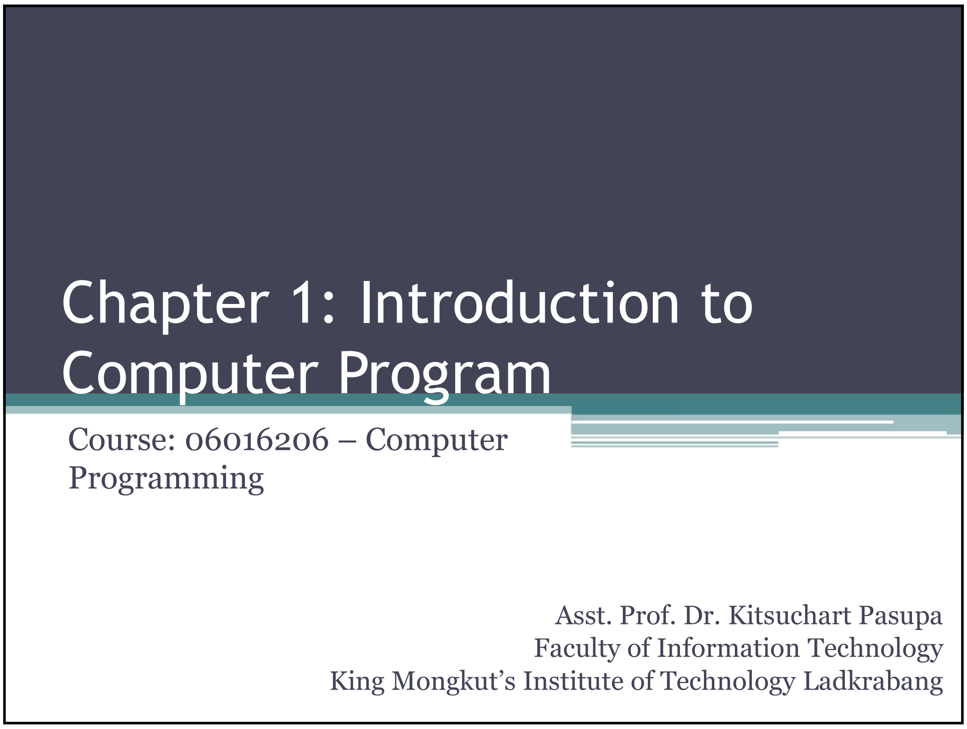
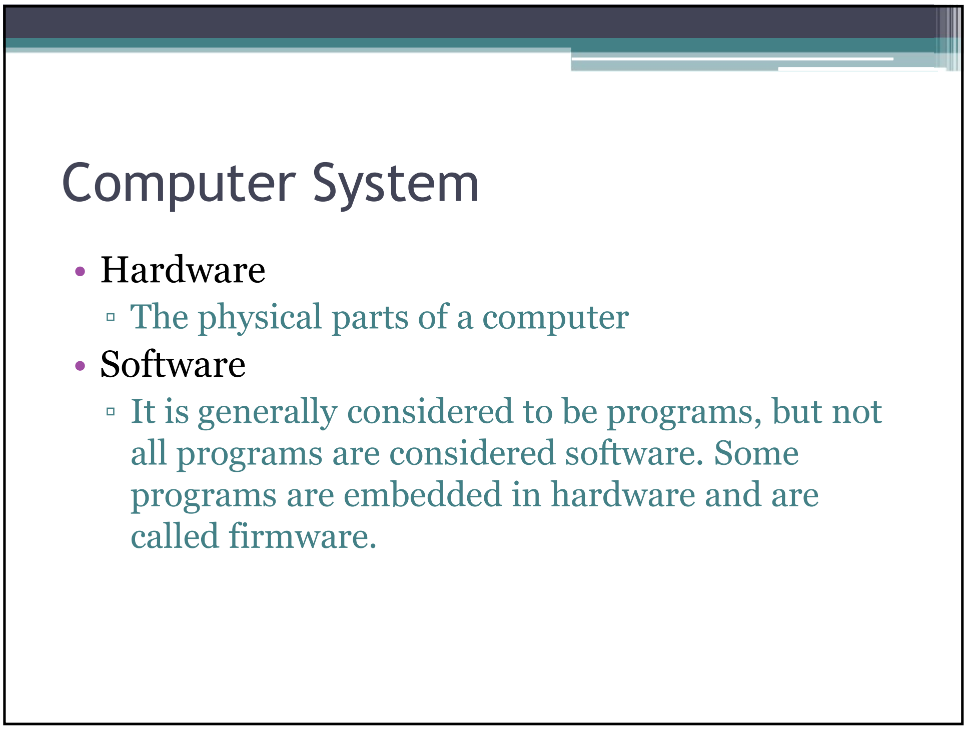


Chapter 1: Introduction to Computer Program

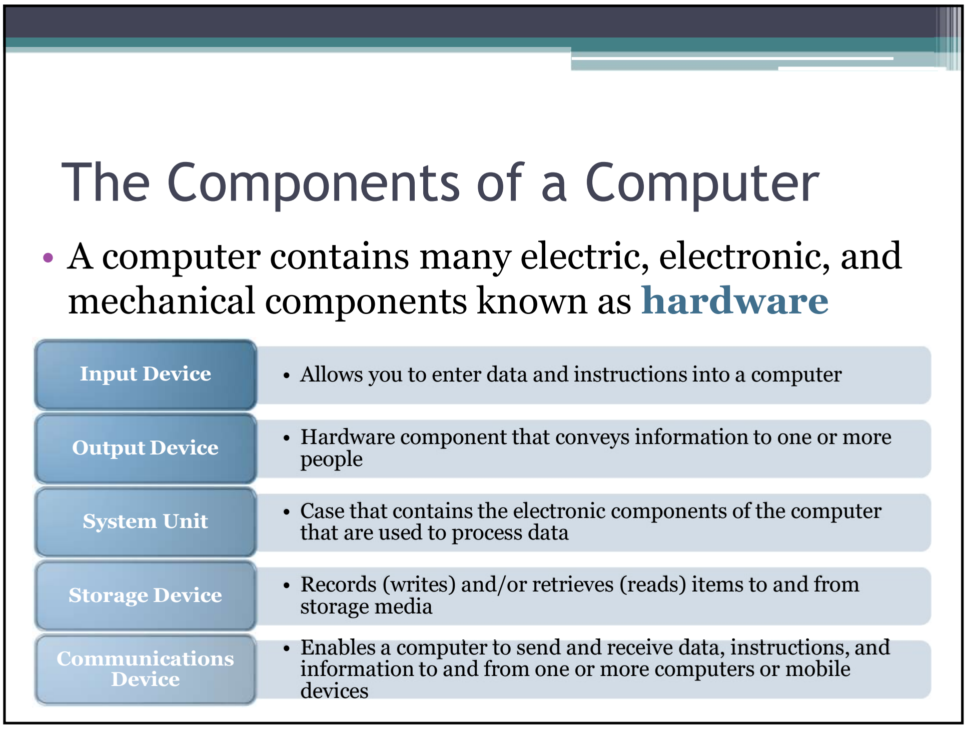
Course: 06016206 – Computer Programming

Asst. Prof. Dr. Kitsuchart Pasupa
Faculty of Information Technology
King Mongkut's Institute of Technology Ladkrabang



Computer System

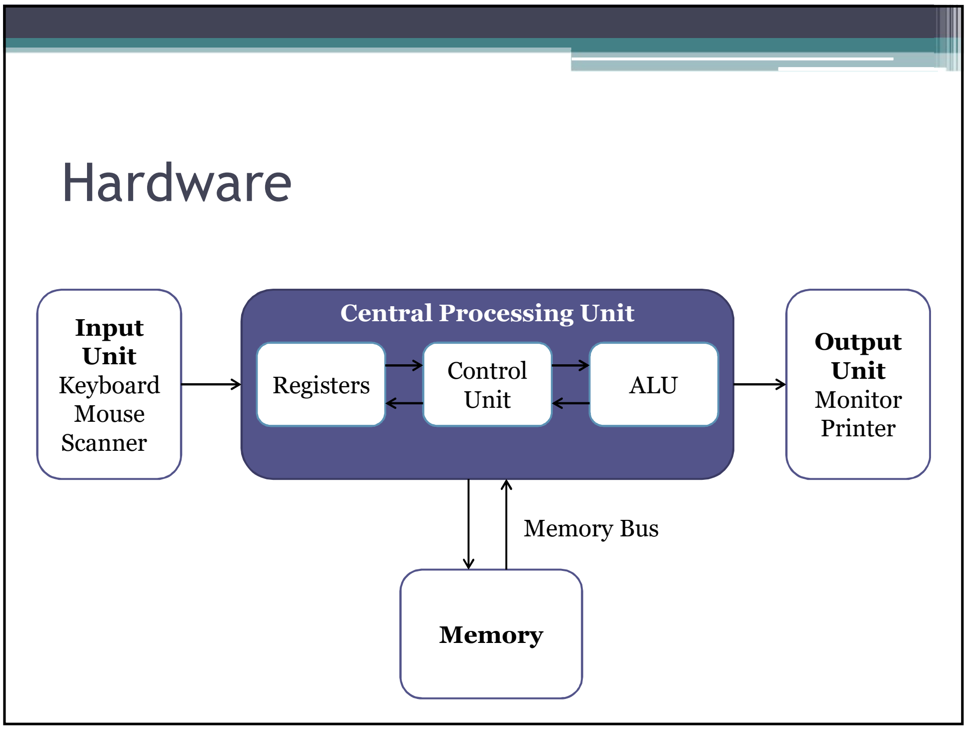
- Hardware
  - The physical parts of a computer
- Software
  - It is generally considered to be programs, but not all programs are considered software. Some programs are embedded in hardware and are called firmware.



The Components of a Computer

- A computer contains many electric, electronic, and mechanical components known as **hardware**

Input Device	• Allows you to enter data and instructions into a computer
Output Device	• Hardware component that conveys information to one or more people
System Unit	• Case that contains the electronic components of the computer that are used to process data
Storage Device	• Records (writes) and/or retrieves (reads) items to and from storage media
Communications Device	• Enables a computer to send and receive data, instructions, and information to and from one or more computers or mobile devices



Hardware

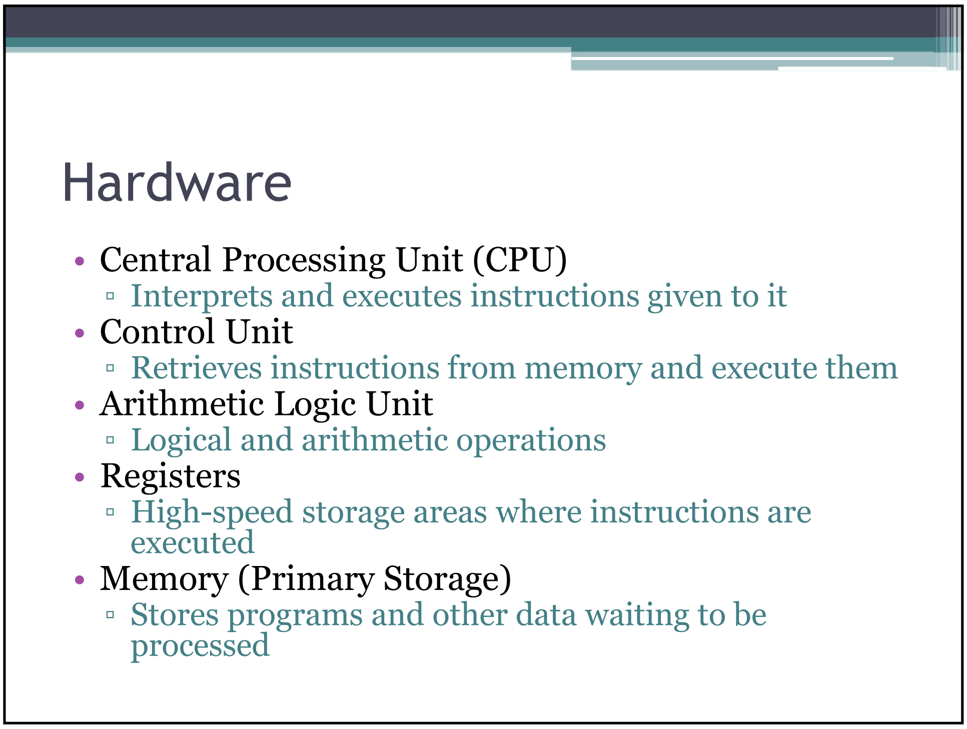
Input Unit
Keyboard
Mouse
Scanner

Central Processing Unit
Registers
Control Unit
ALU

Output Unit
Monitor
Printer

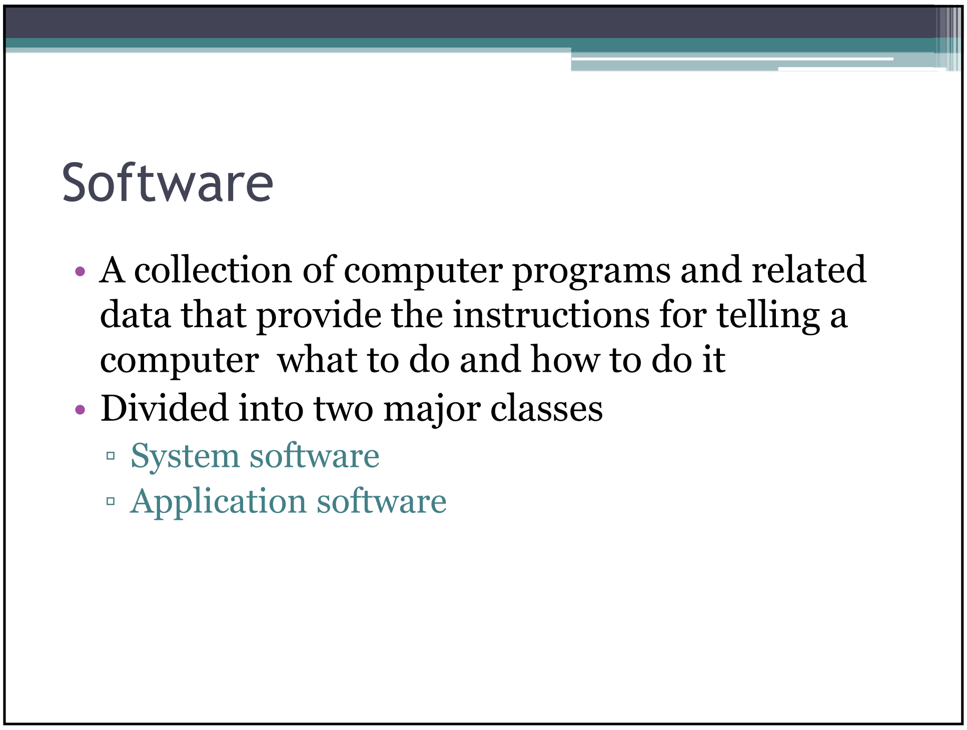
Memory Bus

Memory



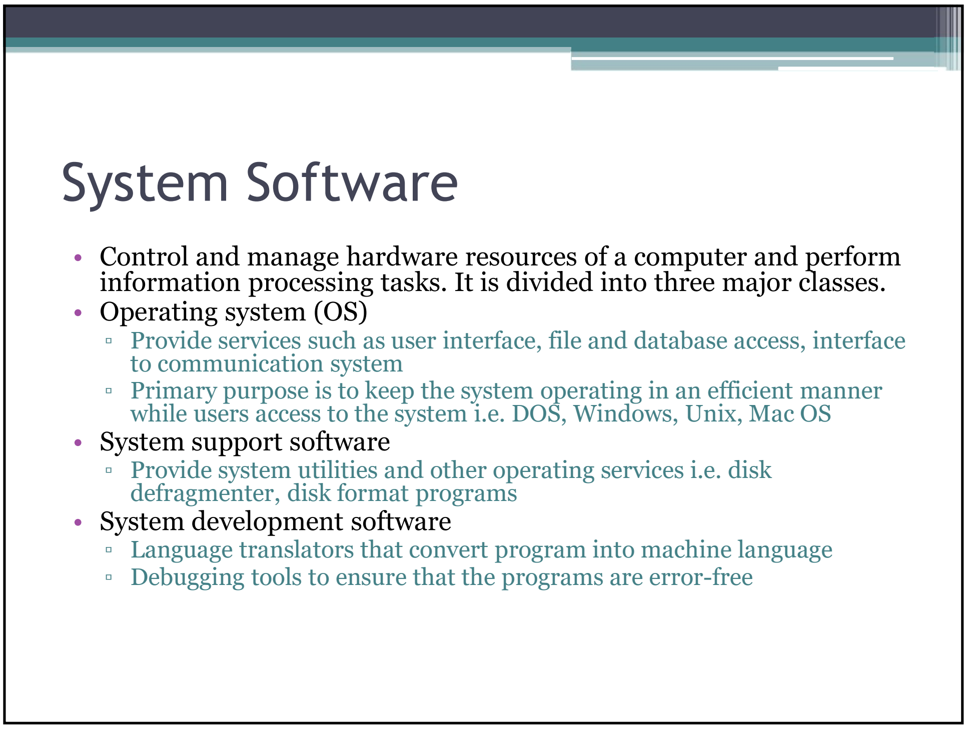
Hardware

- Central Processing Unit (CPU)
  - Interprets and executes instructions given to it
- Control Unit
  - Retrieves instructions from memory and execute them
- Arithmetic Logic Unit
  - Logical and arithmetic operations
- Registers
  - High-speed storage areas where instructions are executed
- Memory (Primary Storage)
  - Stores programs and other data waiting to be processed



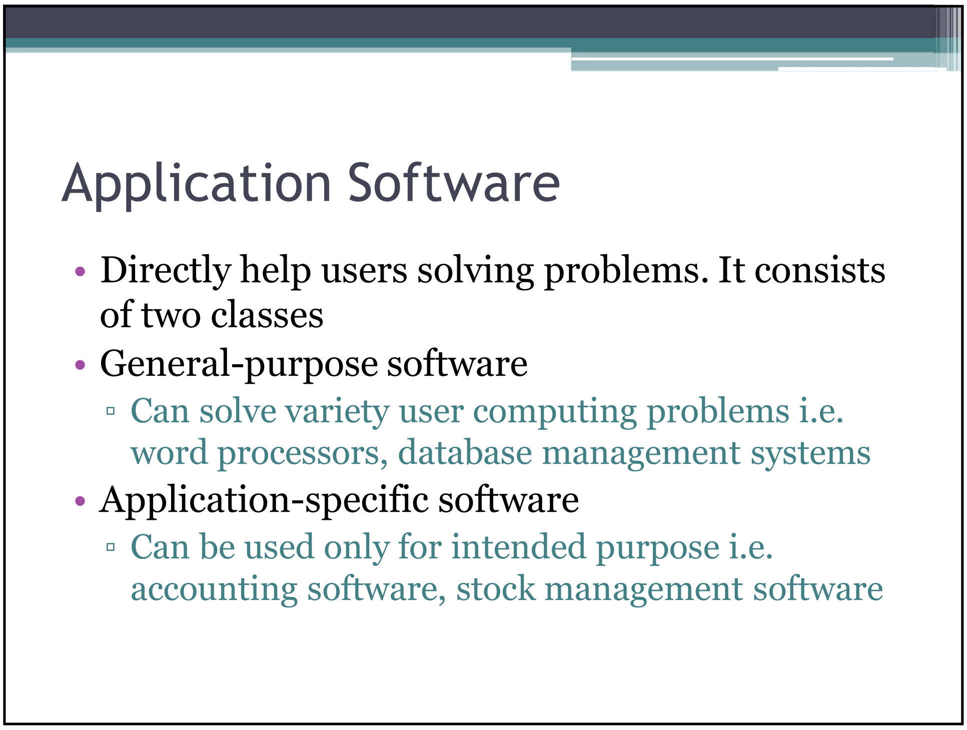
Software

- A collection of computer programs and related data that provide the instructions for telling a computer what to do and how to do it
- Divided into two major classes
  - System software
  - Application software



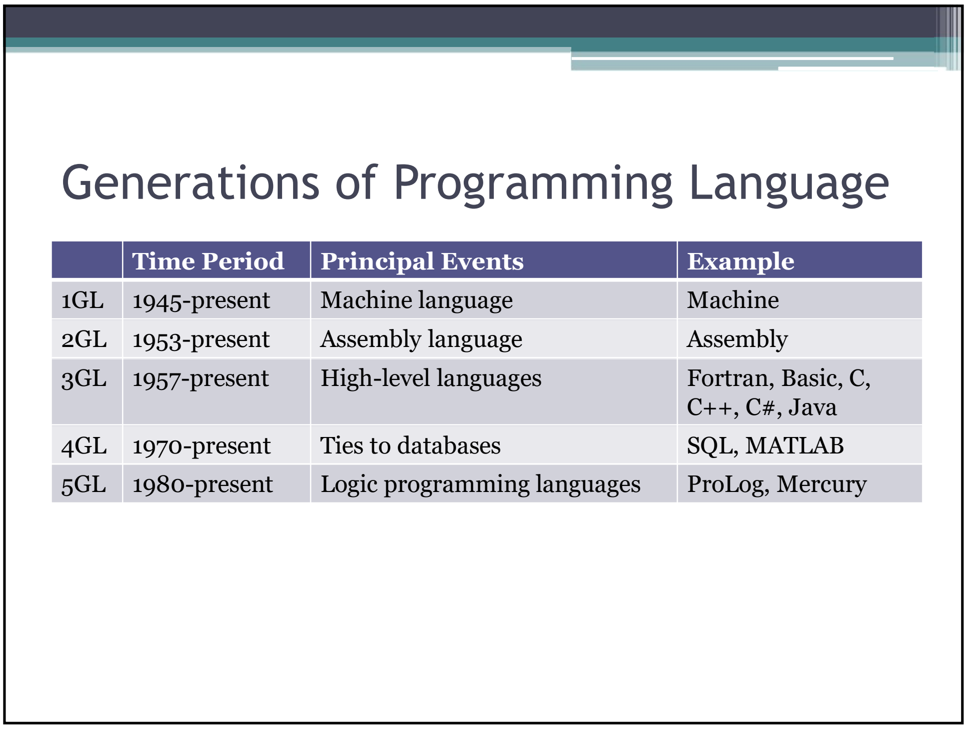
System Software

- Control and manage hardware resources of a computer and perform information processing tasks. It is divided into three major classes.
- Operating system (OS)
  - Provide services such as user interface, file and database access, interface to communication system
  - Primary purpose is to keep the system operating in an efficient manner while users access to the system i.e. DOS, Windows, Unix, Mac OS
- System support software
  - Provide system utilities and other operating services i.e. disk defragmenter, disk format programs
- System development software
  - Language translators that convert program into machine language
  - Debugging tools to ensure that the programs are error-free



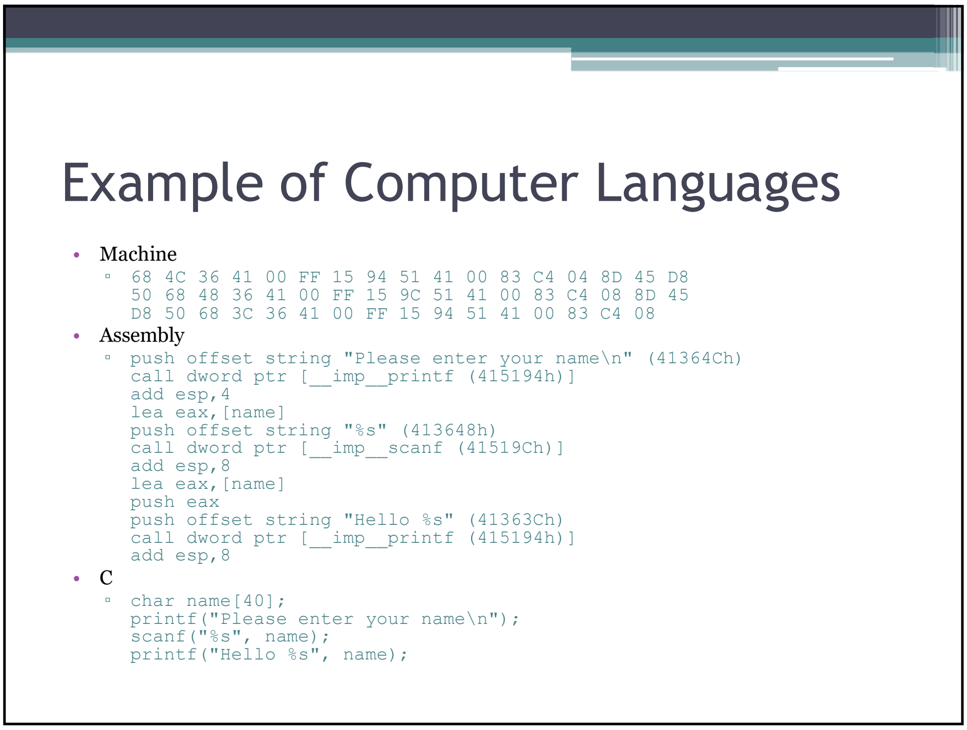
Application Software

- Directly help users solving problems. It consists of two classes
- General-purpose software
  - Can solve variety user computing problems i.e. word processors, database management systems
- Application-specific software
  - Can be used only for intended purpose i.e. accounting software, stock management software



Generations of Programming Language

	Time Period	Principal Events	Example
1GL	1945-present	Machine language	Machine
2GL	1953-present	Assembly language	Assembly
3GL	1957-present	High-level languages	Fortran, Basic, C, C++, C#, Java
4GL	1970-present	Ties to databases	SQL, MATLAB
5GL	1980-present	Logic programming languages	ProLog, Mercury



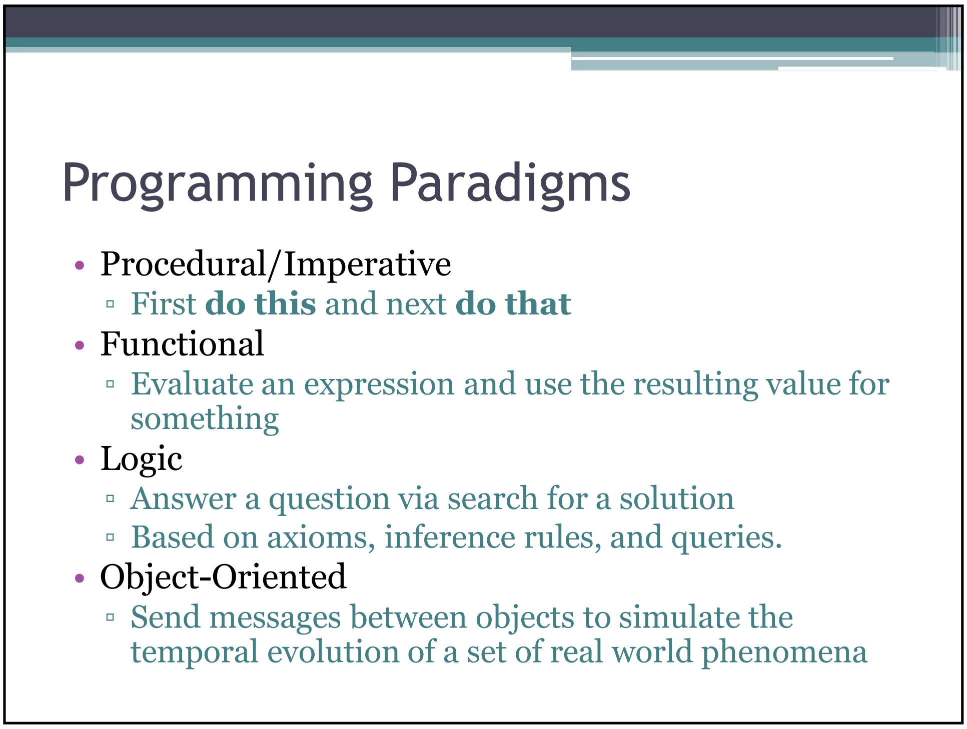
Example of Computer Languages

- Machine

  -
    ```
    68 4C 36 41 00 FF 15 94 51 41 00 83 C4 04 8D 45 D8
    50 68 48 36 41 00 FF 15 9C 51 41 00 83 C4 08 8D 45
    D8 50 68 3C 36 41 00 FF 15 94 51 41 00 83 C4 08
    ```
- Assembly

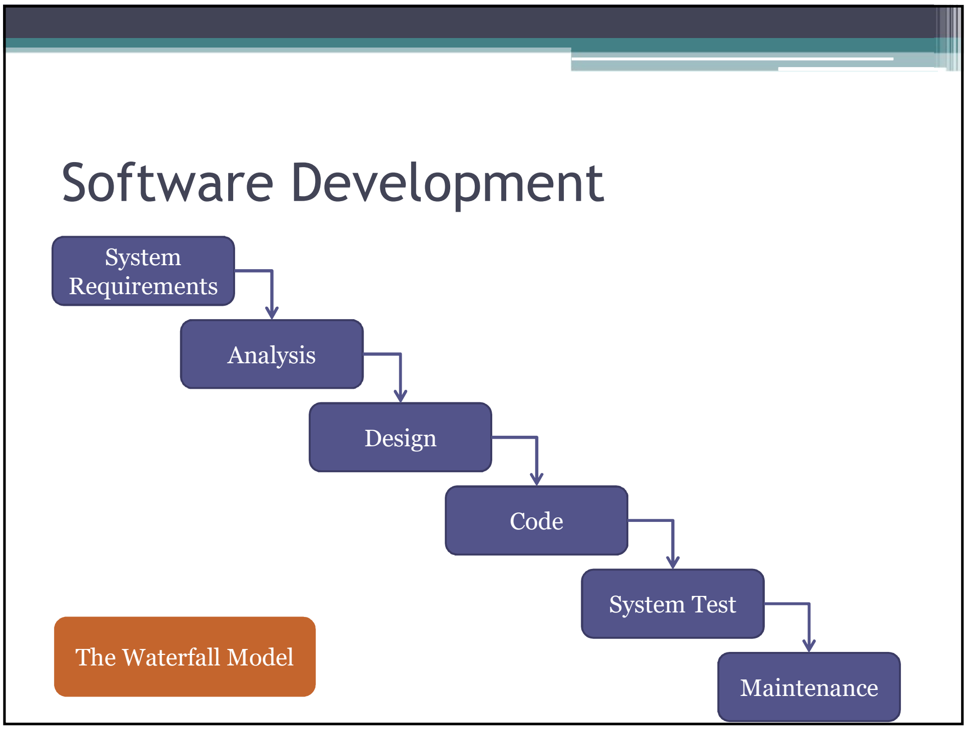
  -
    ```
    push offset string "Please enter your name\n" (41364Ch)
    call dword ptr [__imp__printf (415194h)]
    add esp,4
    lea eax,[name]
    push offset string "%s" (413648h)
    call dword ptr [__imp__scanf (41519Ch)]
    add esp,8
    lea eax,[name]
    push eax
    push offset string "Hello %s" (41363Ch)
    call dword ptr [__imp__printf (415194h)]
    add esp,8
    ```
- C

  -
    ```
    char name[40];
    printf("Please enter your name\n");
    scanf("%s", name);
    printf("Hello %s", name);
    ```



Programming Paradigms

- Procedural/Imperative
  - First **do this** and next **do that**
- Functional
  - Evaluate an expression and use the resulting value for something
- Logic
  - Answer a question via search for a solution
  - Based on axioms, inference rules, and queries.
- Object-Oriented
  - Send messages between objects to simulate the temporal evolution of a set of real world phenomena



Software Development

System Requirements

Analysis

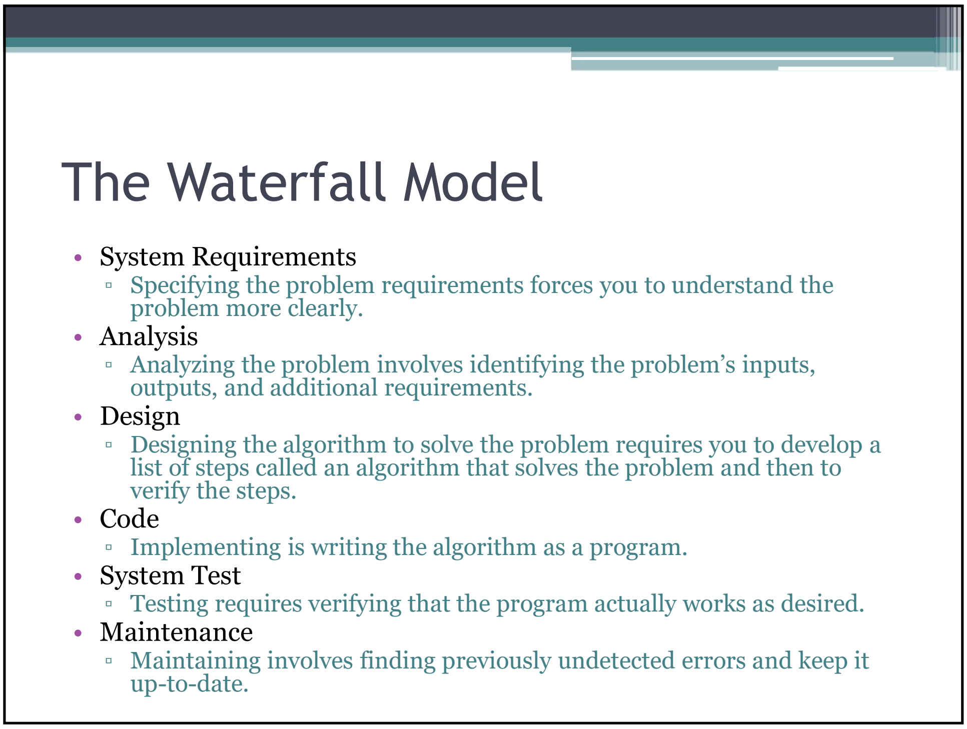
Design

Code

System Test

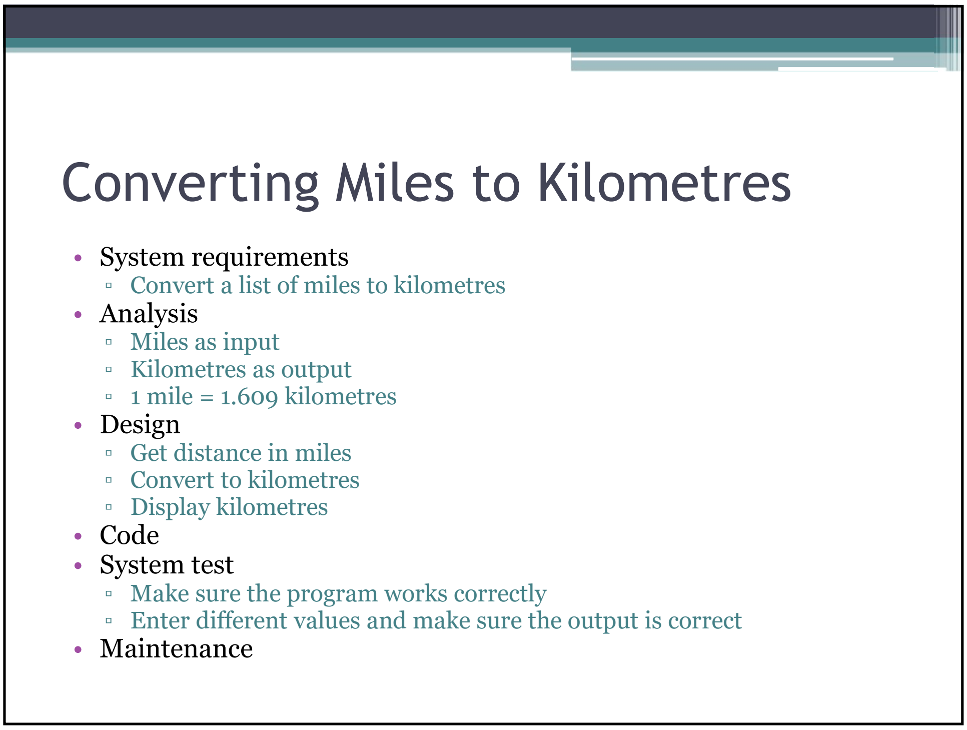
Maintenance

The Waterfall Model



The Waterfall Model

- System Requirements
  - Specifying the problem requirements forces you to understand the problem more clearly.
- Analysis
  - Analyzing the problem involves identifying the problem’s inputs, outputs, and additional requirements.
- Design
  - Designing the algorithm to solve the problem requires you to develop a list of steps called an algorithm that solves the problem and then to verify the steps.
- Code
  - Implementing is writing the algorithm as a program.
- System Test
  - Testing requires verifying that the program actually works as desired.
- Maintenance
  - Maintaining involves finding previously undetected errors and keep it up-to-date.



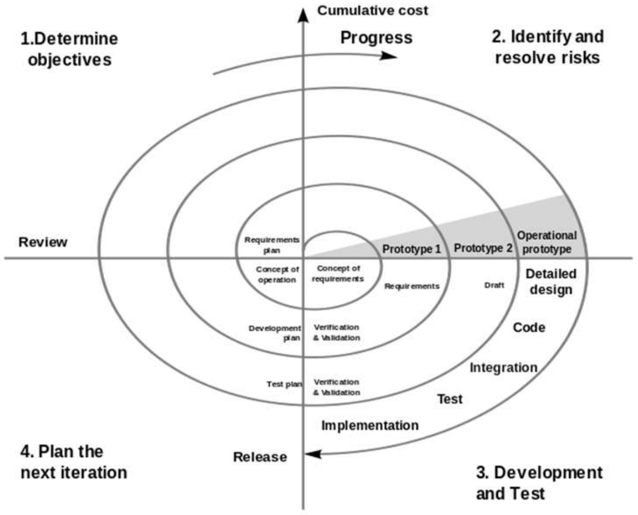
Converting Miles to Kilometres

- System requirements
  - Convert a list of miles to kilometres
- Analysis
  - Miles as input
  - Kilometres as output
  - 1 mile = 1.609 kilometres
- Design
  - Get distance in miles
  - Convert to kilometres
  - Display kilometres
- Code
- System test
  - Make sure the program works correctly
  - Enter different values and make sure the output is correct
- Maintenance

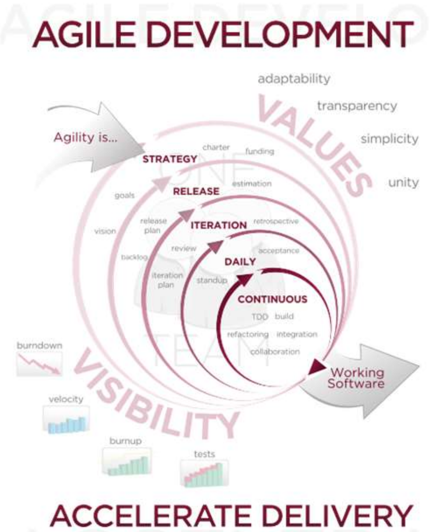
Other methods

- Spiral
- Agile
- Etc.

Spiral

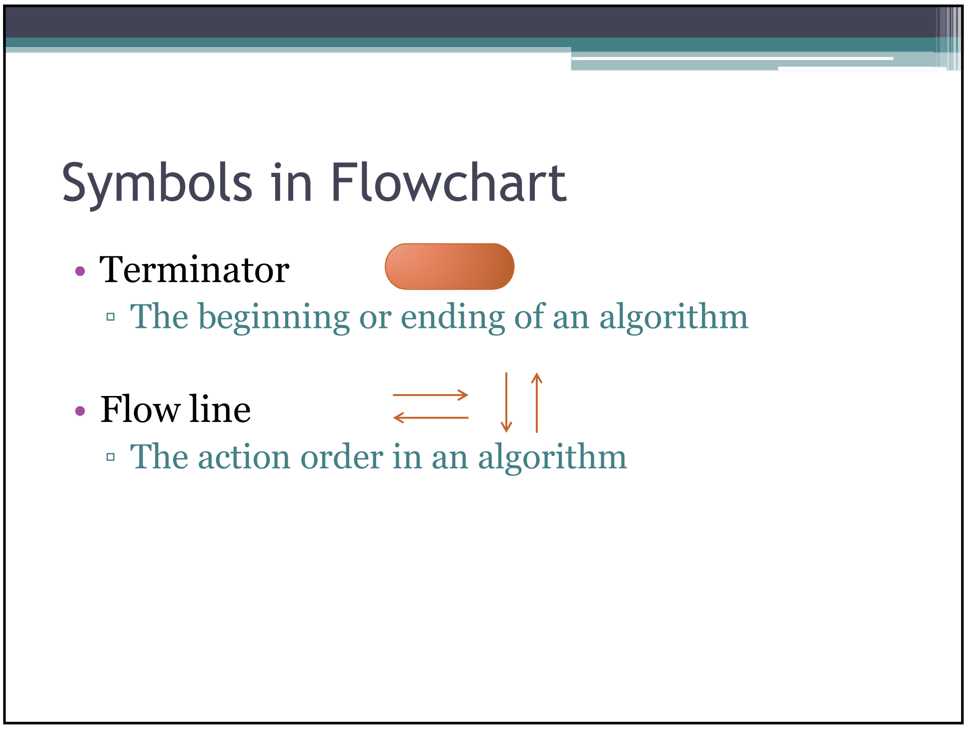


Agile



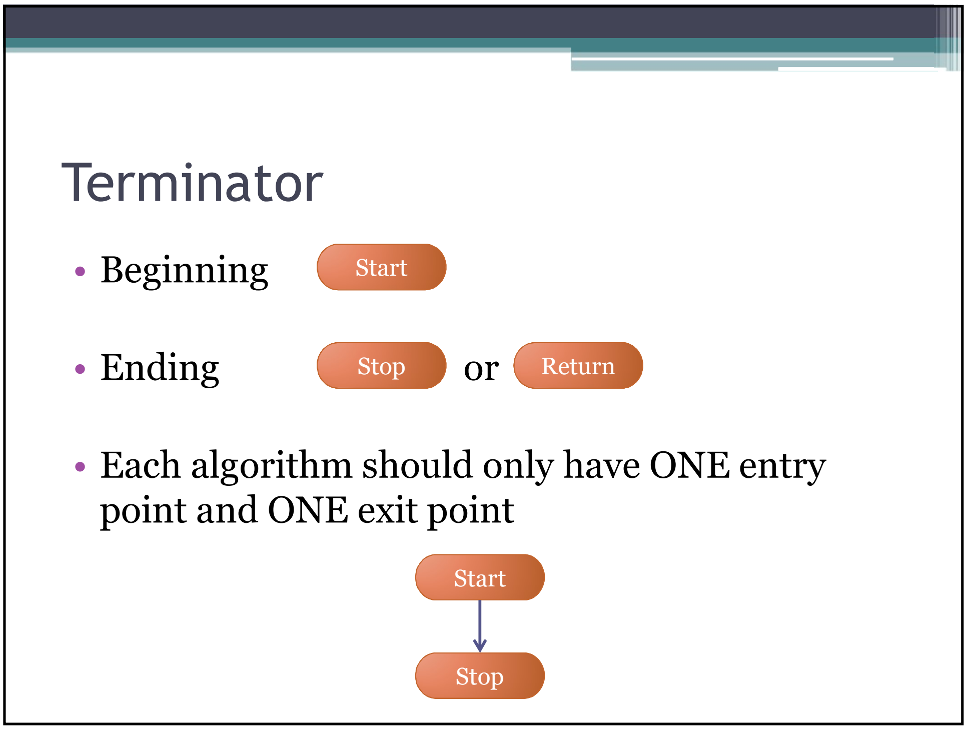
Flowchart & Pseudo code

- Flowchart
  - A diagram that shows the step-by-step execution of a program
- Pseudo code
  - A combination of English phrases and language constructs to describe algorithm steps. Algorithm is a list of steps for solving a problem.



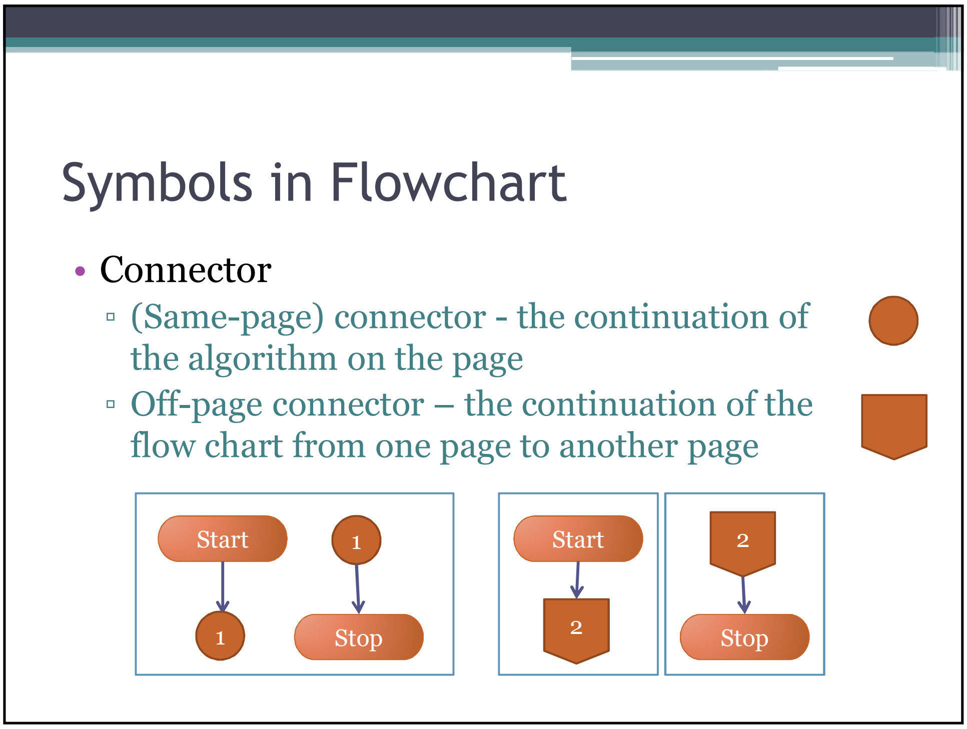
Symbols in Flowchart

- Terminator
  - The beginning or ending of an algorithm
- Flow line
  - The action order in an algorithm



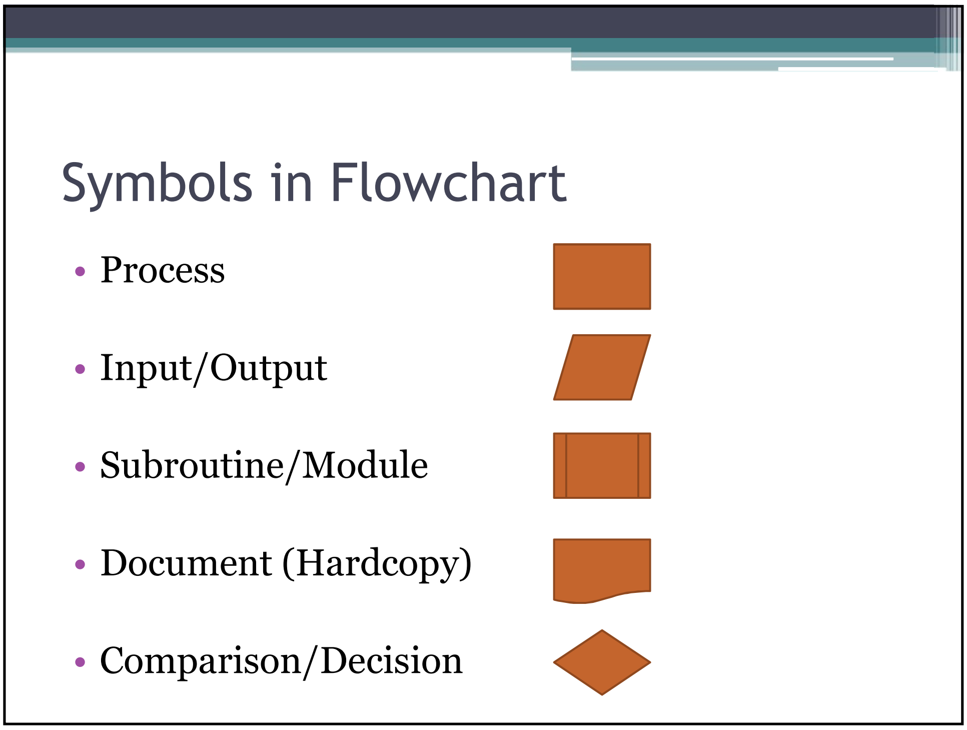
Terminator

- Beginning Start
- Ending Stop or Return
- Each algorithm should only have ONE entry point and ONE exit point



Symbols in Flowchart

- Connector
  - (Same-page) connector - the continuation of the algorithm on the page
  - Off-page connector – the continuation of the flow chart from one page to another page



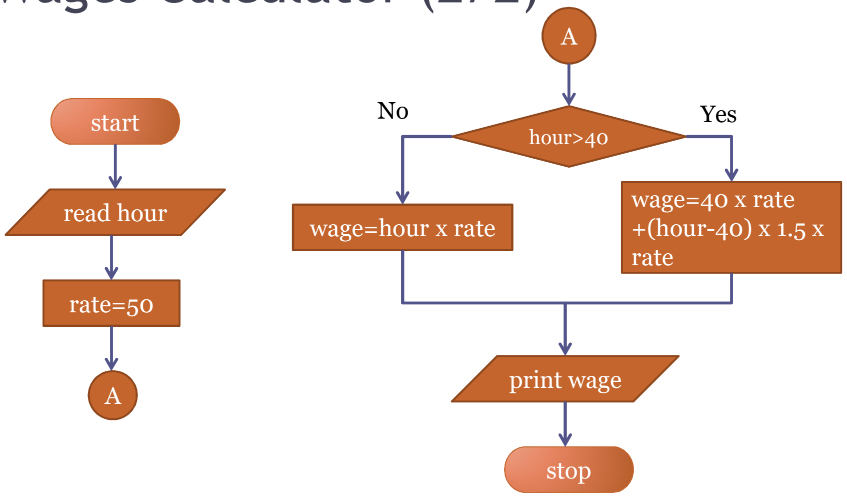
Symbols in Flowchart

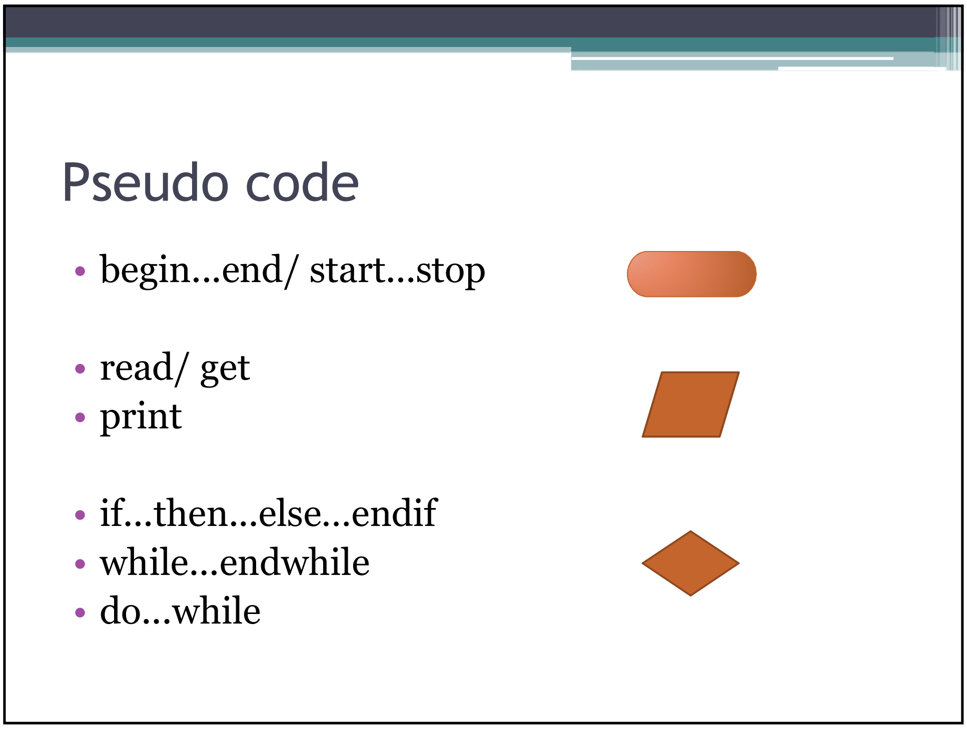
- Process
- Input/Output
- Subroutine/Module
- Document (Hardcopy)
- Comparison/Decision

Wages Calculator (1/2)

- Problem: wages calculator which is applicable for both regular and overtime rate.
- If an employee works more than 40 hours a week, he/she is entitled to overtime pay equal to 1.5 times this regular rate of pay.
- Regular rate of pay: 50 Baht/hour

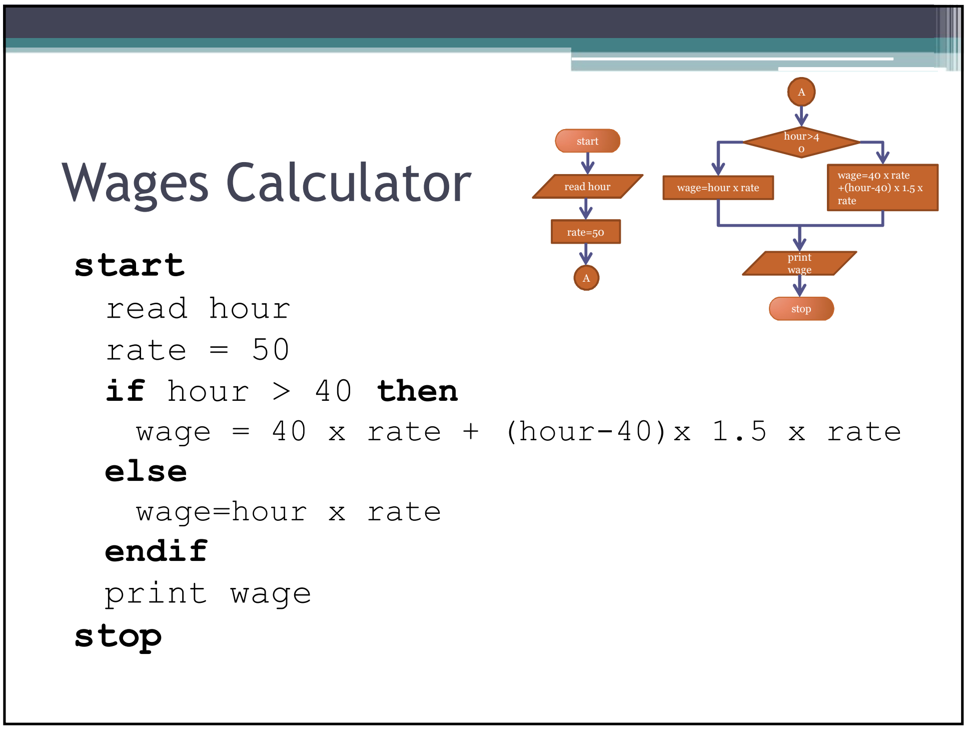
Wages Calculator (2/2)





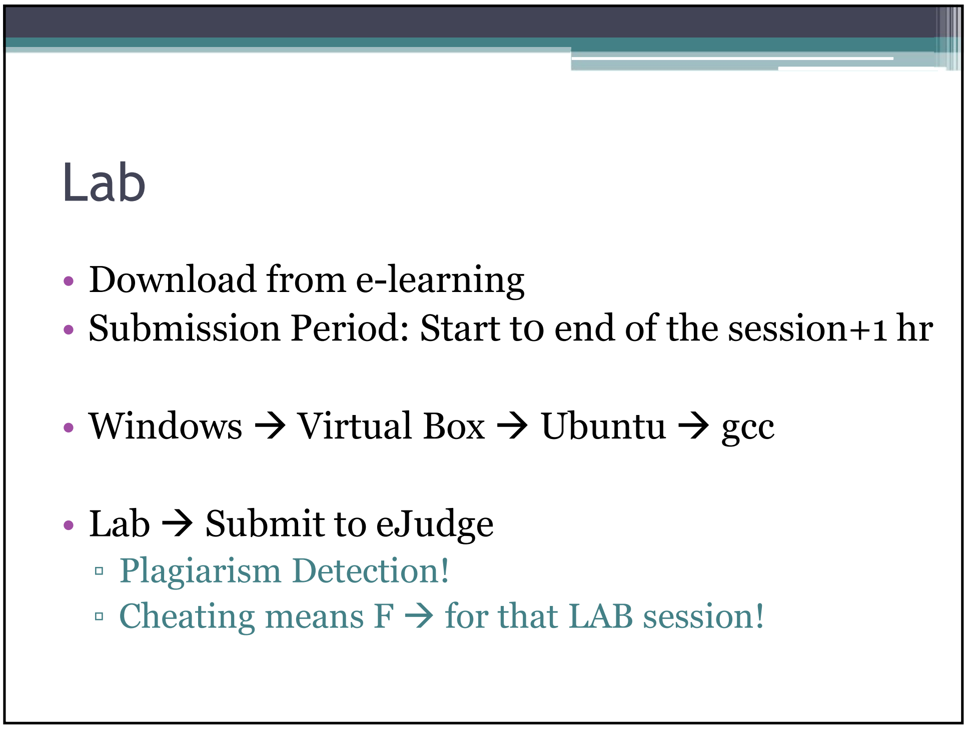
Pseudo code

- begin…end/ start…stop
- read/ get
- print
- if…then…else…endif
- while…endwhile
- do…while



Wages Calculator

```
start
  read hour
  rate = 50
  if hour > 40 then
    wage = 40 x rate + (hour-40)x 1.5 x rate
  else
    wage=hour x rate
  endif
  print wage
stop
```



Lab

- Download from e-learning
- Submission Period: Start to end of the session+1 hr

- Windows → Virtual Box → Ubuntu → gcc

- Lab → Submit to eJudge
  - Plagiarism Detection!
  - Cheating means F → for that LAB session!