เริ่มต้นโปรแกรมภาษาซี

Introduction to C/C++ Programming Language

เอกสารประกอบการสอนวิชา CP111

มหาวิทยาลัยศรีนครินทรวิโรฒ

2 November 2010

อ. เอก ตังคนานนท์

เนื้อหา

Agenda

- แนะนำวิชา CP111
- แนะนำภาษาซี
- การออกแบบการเขียนโปรแกรม
- เครื่องมือในการอธิบายโปรแกรม

แนะนำวิชา CP111

Introduction to CP111

ทำไมต้องเรียน CP111 ?

Why study CP111 ?

- เพื่อฝึกแนวคิดให้เป็นระบบ
- เพื่อให้เข้าใจถึงศิลปะการเขียนโปรแกรม
- เพื่อเป็นพื้นฐานสำหรับการเขียนโปรแกรมภาษาอื่นๆ
- เพื่อนำการเขียนโปรแกรมภาษาซีไปประยุกต์ใช้งาน

ผู้สอน
Lecturer

- อ.เอก ตังคนานนท์
  - ห้องพัก 19-1818
  - ake@swu.ac.th
- เวลาออฟฟิสสำหรับ CP111
  - วันอังคาร 13.00 – 16.00

เค้าโครงการเรียนการสอน

Course Syllabus

- บทที่ 1 - เริ่มต้นโปรแกรมภาษาซี
- บทที่ 2 - ตัวแปร ประเภทตัวแปร ตัวดำเนินการ
- บทที่ 3 - การรับและการแสดงข้อมูล
- บทที่ 4 - โครงสร้างของโปรแกรมและการควบคุม
- บทที่ 5 - ฟังก์ชั่น
- บทที่ 6 - ประเภทตัวแปรขั้นสูง
- บทที่ 7 – ไลบรารี่พื้นฐาน
- บทที่ 8 – โครงสร้างข้อมูลพื้นฐาน

ข้อตกลงในการเรียน

Rules

- ไม่คุยกันในชั้นเรียน เพราะจะได้มีสมาธิในการเรียนและไม่ไปรบกวนสมาธิของเพื่อน
- หากอยู่หน้าเครื่องคอมพิวเตอร์ให้ทำกิจกรรมที่เกี่ยวกับการเรียนการสอนเท่านั้น
- ห้ามมิให้มีการคัดลอกงานจากคนอื่นโดยปราศจากการอ้างอิง หากตรวจพบ plagiarism จะได้เกรด E ทันที

แนะนำภาษาซี

Introduction to C Programming Language

ภาษาคอมพิวเตอร์ระดับต่ำ

Low-Level Computer Programming Language

- **ภาษาระดับต่ำ**

```
0100 0101 1010 1111
1101 1010 1011 1100
1010 1111 1011 1000
1100 0010 0110 1011
1011 0101 1101 1001
```

  - **ภาษาเครื่อง**

    มีลักษณะเป็นเลขฐานสอง เป็นภาษาที่คอมพิวเตอร์เข้าใจได้ดีแต่คนเข้าใจได้ยาก

```
push dx ah
mov ds, ax
mov ah, 9
int 21h
```

  - **ภาษาแอสเซมบลี**

    เริ่มมีการใช้ตัวอักษรแทนชุดคำสั่งที่เป็นเลขฐานสอง แต่ก็ยังทำให้การพัฒนาโปรแกรมเป็นไปได้ค่อนลำบาก

  - **ภาษาอื่นๆ ที่มีความเกี่ยวข้องกับสถาปัตยกรรมของคอมพิวเตอร์โดยตรง**

ภาษาคอมพิวเตอร์ระดับสูง

High-Level Computer Programming Language

- **ภาษาระดับสูง**
  - **ภาษา C/C++**

```
int a;
for (a = 1; a < 10; a++) {
   printf('a = %d' , a);
   printf('2 x a = %d', 2 * a);
}
```

เป็นภาษาที่ถูกออกแบบมาให้มนุษย์สามารถเข้าใจได้ง่ายและสามารถใช้สั่งงานคอมพิวเตอร์ได้อย่างมีประสิทธิภาพ

  - **ภาษาอื่นๆ ที่มีการบดบังการติดต่อกับสถาปัตยกรรมของคอมพิวเตอร์**

ลักษณะการทำงานของภาษาระดับสูง

Computer Programming Language

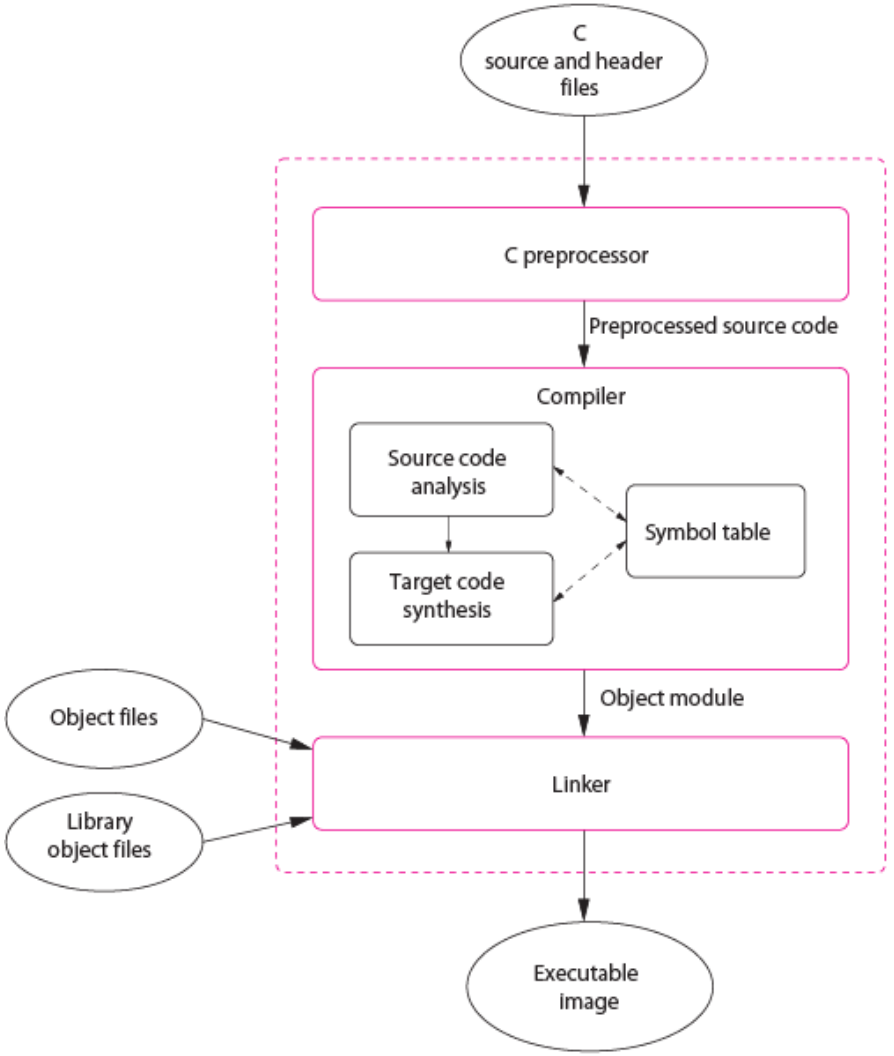
- **Interpreted**
  - เป็นภาษาที่ทำงานผ่าน interpreter
  - Python, MATLAB, PHP, Perl, etc.
- **Compiled**
  - เป็นภาษาที่ถูกแปลงให้เป็นภาษาเครื่องก่อนแล้วจึงทำงาน
  - C, C++, COBOL, Visual Basic, etc.
- บางภาษาอาจทำงานได้ทั้งสองรูปแบบ

อาจแบ่งเพิ่มเติมเป็น Translated หมายถึงภาษาที่มีการแปลงเป็นภาษาอื่นก่อนแล้ว compile โดยภาษาที่เป็นที่นิยมแปลงไปคือ C

คอมไพเลอร์ภาษาซี

The C Complier

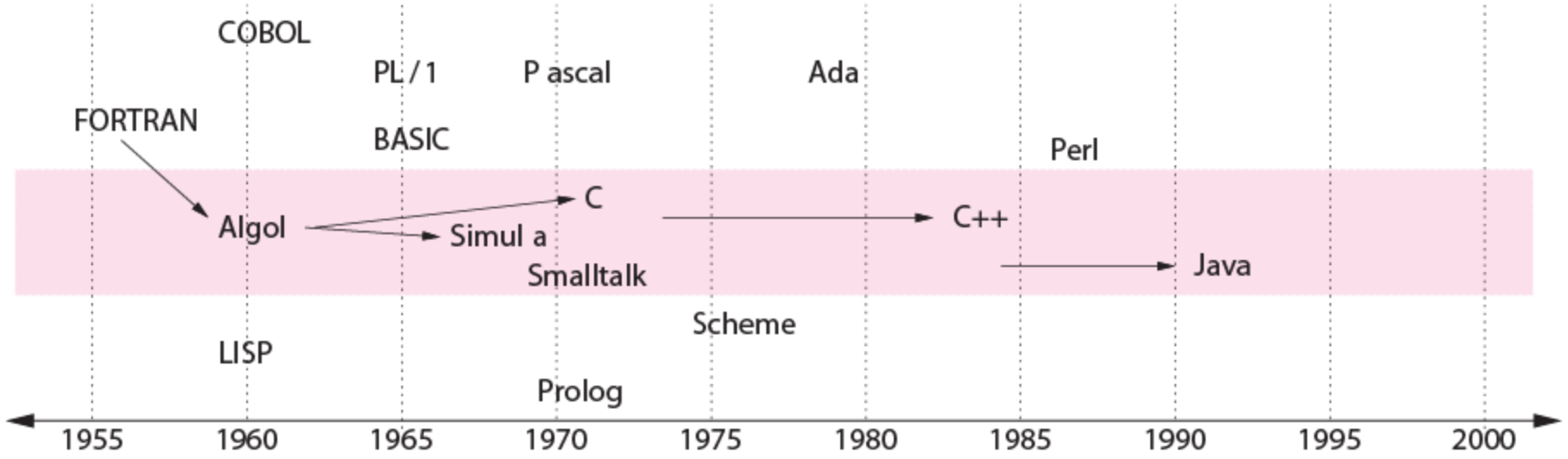
- **Preprocessor**
  - macro substitution
  - conditional compilation
  - “source-level” transformations
  - output is still C
- **Compiler**
  - generates object file
  - machine instructions
- **Linker**
  - combine object files (including libraries) into executable image



Citation: Yale N. Patt, Sanjay J. Patel. Introduction to Computing Systems 2nd Edition Pg 295 - 297.

อ. เอก ตังคนานนท์

พัฒนาการของภาษาคอมพิวเตอร์



Citation: Yale N. Patt, Sanjay J. Patel. Introduction to Computing Systems 2nd Edition Pg 294.

สวัสดีชาวโลก

Hello World !!!

ex01_01.c

```c
#include<stdio.h>

int main() {
   printf("Hello World");
   return 0;
}
```

ex01_02.cpp

```cpp
#include<iostream>

using namespace std;

int main() {
   cout << "Hello World" << endl;
   return 0;
}
```

สวัสดีชาวโลกหลายๆ ครั้ง

Hello World More!!!

ex01_03.c

```c
#include<stdio.h>

int main() {
   int a = 0;
   while (a < 10) {
      printf("Hello World");
      a++;
   }

   return 0;
}
```

ex01_04.cpp

```cpp
#include<iostream>

using namespace std;

int main() {
   int a = 0;
   while (a < 10) {
      cout << "Hello World" << endl;
      a++;
   }

   return 0;
}
```

โปรแกรมสำหรับภาษาอื่นๆ

Other Computer Languages

ex01_05.java

```
class HelloWorld {
   public static void main(String [] args) {
      int a = 0;
      while (a < 10) {
         System.out.println("Hello World");
         a++;
      }
   }
}
```

ex01_07.py

```
for a in range(0, 10):
   print "Hello World"
```

ex01_08.php

```
for ($a = 0; $a < 10; $a++)
   print "Hello World";
```

- ดูเพิ่มเติมที่ http://www.99-bottles-of-beer.net
- จะเห็นว่าภาษาคอมพิวเตอร์แต่ละชนิดคล้ายคลึงกัน ถ้าเข้าใจภาษาหนึ่งก็จะเข้าใจที่เหลือได้โดยเร็ว

การออกแบบการเขียนโปรแกรม

Computer Programming Design

การแก้ปัญหาด้วยคอมพิวเตอร์

Solving Problems using Computers

- จะบอกคอมพิวเตอร์ให้ทำงานตามที่เราต้องการได้อย่างไร ?
- การแปลงการแก้ปัญหาออกมาเป็นขั้นตอนต่างๆ
- การนำเอาขั้นตอนที่ได้นั้นไปพัฒนาเป็นโปรแกรม

อ. เอก ตังคนานนท์

ทำปัญหาให้ชัดเจน (1/2)

Problem Statement

- ปัญหาส่วนใหญ่ไม่ชัดเจน คลุมเครือ
  - ให้หาผลรวมของจำนวนเต็ม 1 ถึง 10
    - ระหว่าง หรือ ตั้งแต่ ?
  - ให้หาพื้นที่ใต้กราฟของสมการ $f(x) = x^2 + 5$
    - ช่วงไหน ?
  - ให้นับจำนวนตัวอักษรทั้งหมดใน slide หน้านี้
    - สัญลักษณ์ต่างๆ ช่องว่าง ฯลฯ นับด้วยหรือไม่ ?

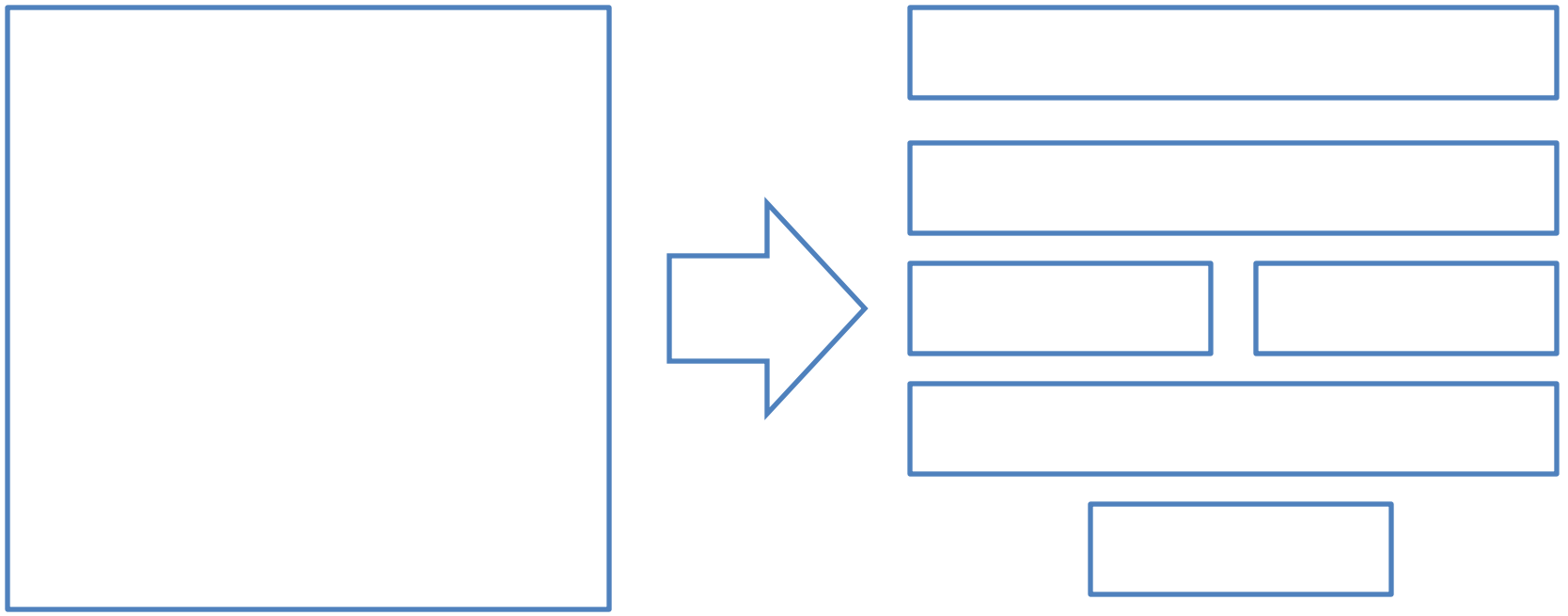
ทำปัญหาให้ชัดเจน (2/2)

Problem Statement

- การจัดการกับปัญหาที่ไม่ชัดเจน คลุมเครือ
  - ถาม
  - ทำตามที่เข้าใจและอธิบายเอาไว้ในส่วนที่คลุมเครือ (ไม่ควรใช้วิธีนี้ใน CP111)

แก้ปัญหาอย่างมีระบบ (1/2)

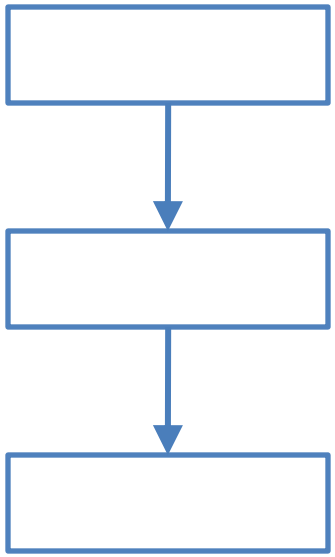
- แบ่งปัญหาหลักออกเป็นปัญหาย่อยๆ



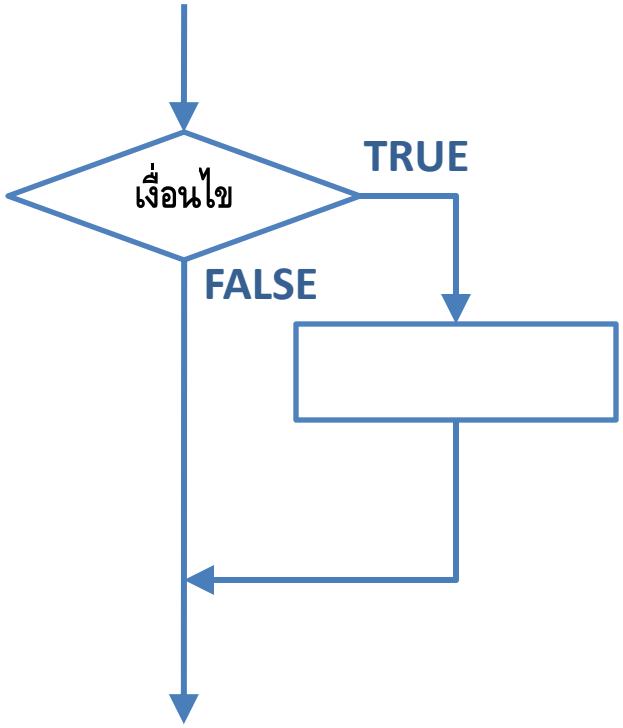
แก้ปัญหาอย่างมีระบบ (2/2)

- ลักษณะความสัมพันธ์ของปัญหาย่อยๆ

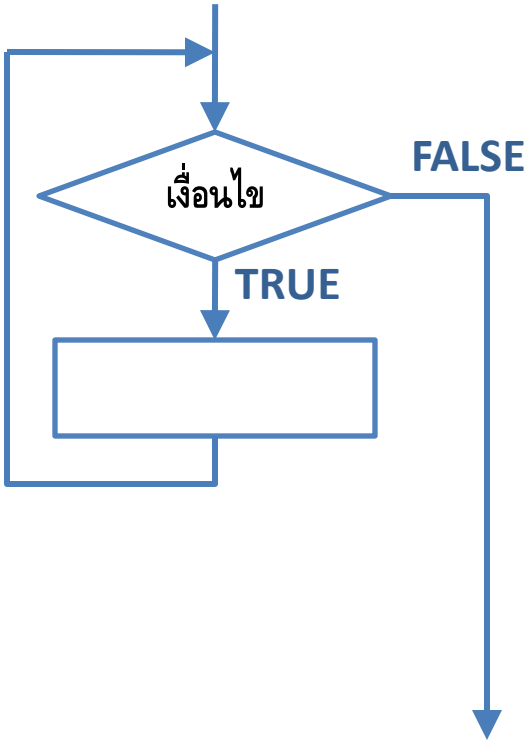
ทำตามลำดับ



ทำตามเงื่อนไข



ทำซ้ำตามเงื่อนไข

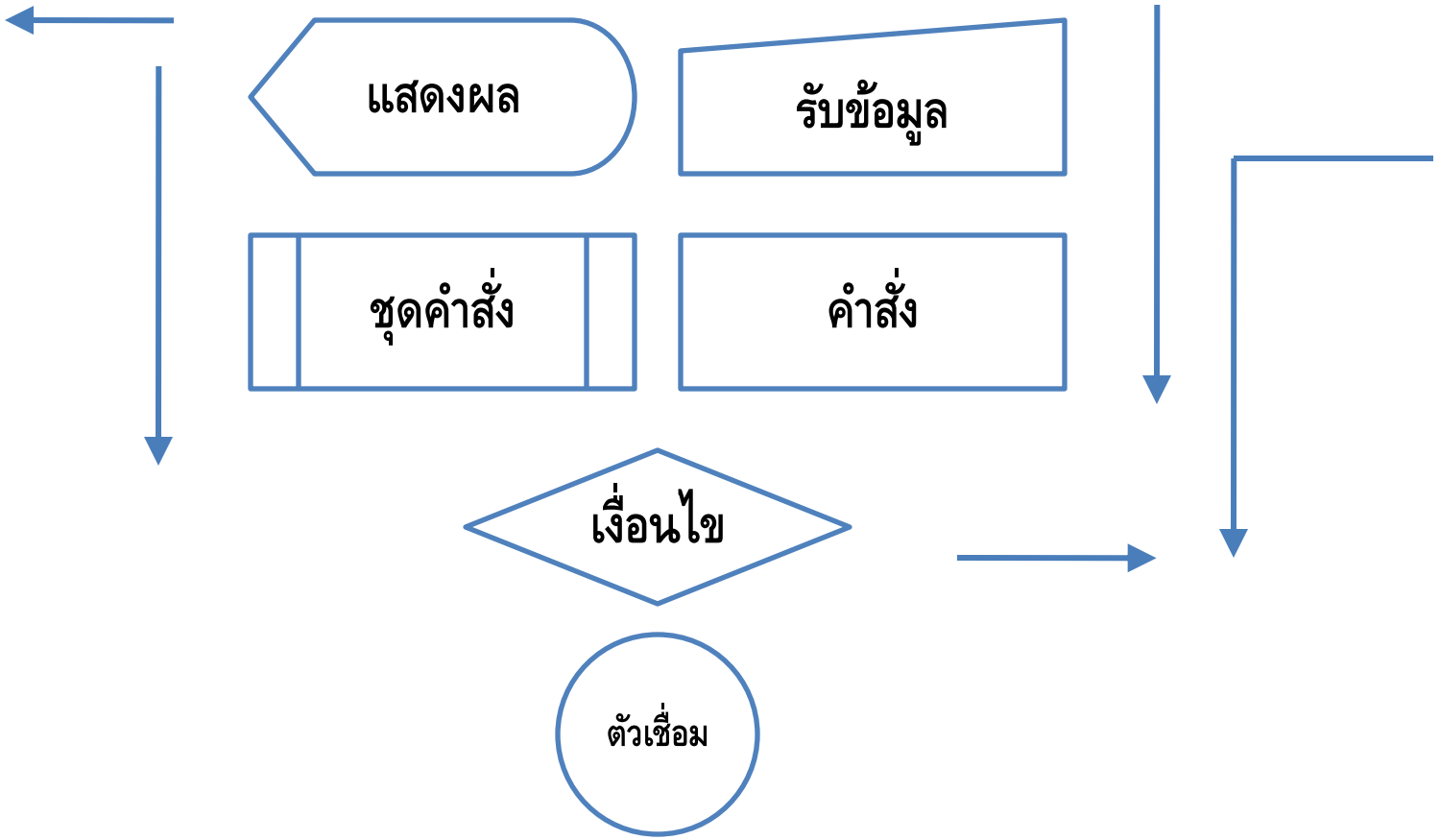


เครื่องมือในการอธิบายโปรแกรม

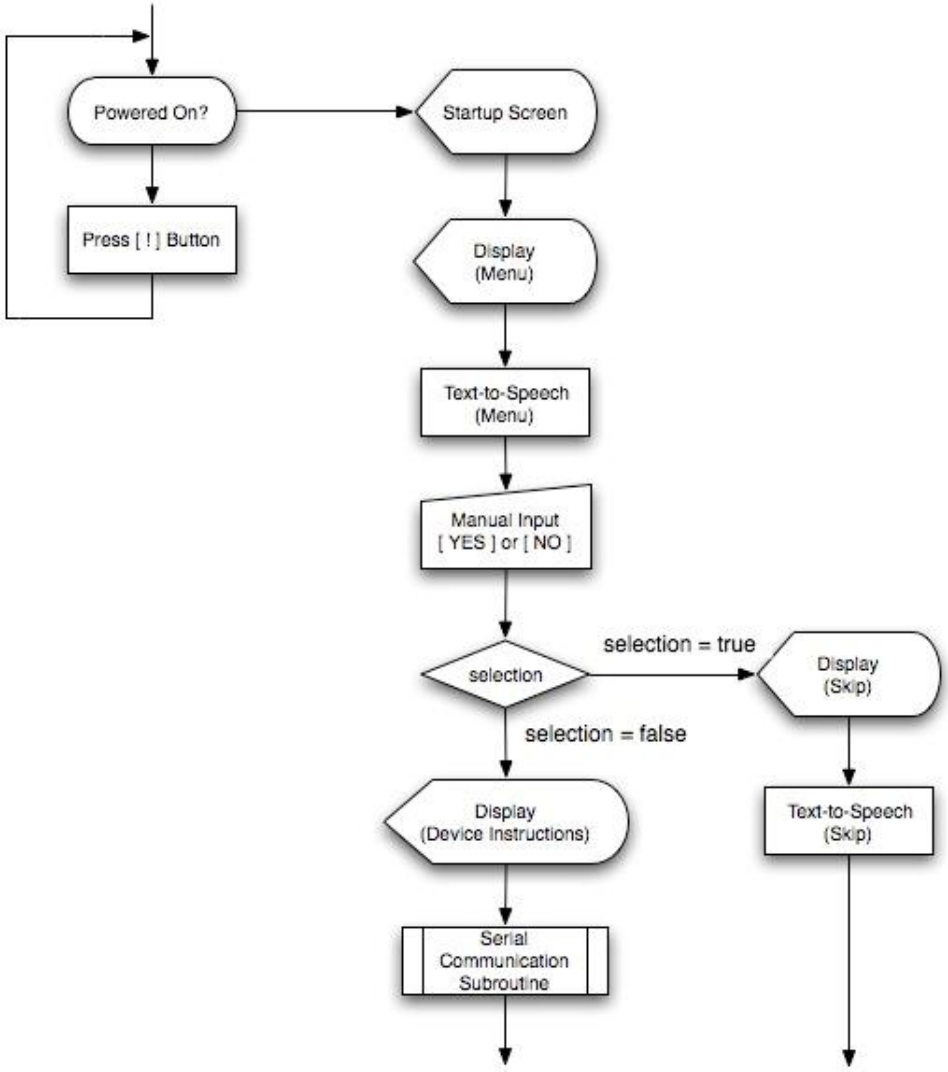
Algorithm Description Tools

Programming Flowchart

- ใช้แสดงการทำงานของโปรแกรมด้วยรูปภาพ



Programming Flowchart Example



Citation: http://www.trinity.edu/student_org/thetatau/report_files/image005.jpg

Programming Pseudocode

- ใช้แสดงการทำงานของโปรแกรมด้วยประโยคง่ายๆ โดยมีการจัดย่อหน้าที่เป็นระบบ

Programming Pseudocode Example

```
 1  function Dijkstra(Graph, source):
 2      for each vertex v in Graph:           // Initializations
 3          dist[v] := infinity               // Unknown distance function from source to v
 4          previous[v] := undefined          // Previous node in optimal path from source
 5      dist[source] := 0                     // Distance from source to source
 6      Q := the set of all nodes in Graph
        // All nodes in the graph are unoptimized - thus are in Q
 7      while Q is not empty:                 // The main loop
 8          u := vertex in Q with smallest dist[]
 9          if dist[u] = infinity:
10              break                         // all remaining vertices are inaccessible from source
11          remove u from Q
12          for each neighbor v of u:         // where v has not yet been removed from Q.
13              alt := dist[u] + dist_between(u, v)
14              if alt < dist[v]:             // Relax (u,v,a)
15                  dist[v] := alt
16                  previous[v] := u
17      return dist[]
```

ตัวอย่าง

Example

- จงนับจำนวนสระภาษาไทยที่ปรากฏบนสไลด์แผ่นนี้
- จงคำนวนค่าส่วนสูงเฉลี่ยของนักศึกษาในห้องนี้
- จงหาวันคล้ายวันเกิดของนักศึกษาในห้องนี้คนถัดไป

คำถามท้ายบท

จะขึ้นสไลด์ท้ายชั่วโมงที่ทำการสอน

จบบทที่ 1

The End of Chapter 1

บทที่ 2 ตัวแปร ประเภทตัวแปร ตัวดำเนินการ

จะเขียนโปรแกรมให้เก็บข้อมูลระหว่างการคำนวนได้อย่างไร ประเภทของข้อมูลที่จัดเก็บมีแบบไหนบ้าง ลักษณะของการคำนวนสามารถทำอะไรได้บ้างนอกจากบวก ลบ คูณ หาร