

Design and Analysis of Data Structures and Algorithm

Warin Wattanapornprom Ph.D.

กฎการฟังบรรยาย



K



ประมวลรายวิชา (Course Syllabus)

- CSS121 Design and Analysis of Data Structures and Algorithms
CSS121 การออกแบบและวิเคราะห์โครงสร้างข้อมูลและขั้นตอนวิธี
- ชื่อผู้สอน อ.ดร.วรินทร์ วัฒนพรพรหม
- จำนวน 4 หน่วยกิต สถานภาพวิชา วิชาบังคับ มีวิชาต่อเนื่อง
- ชื่อหลักสูตร วิทยาศาสตร์บัณฑิต
- วิชาระดับปริญญาตรี

ประมวลรายวิชา (ต่อ)

วัตถุประสงค์ทั่วไป

- เพื่อให้นักศึกษาสามารถเข้าใจข้อดีข้อเสียและข้อจำกัดของโครงสร้างข้อมูลและขั้นตอนวิธี
- เพื่อให้นักศึกษาสามารถวิเคราะห์โครงสร้างข้อมูลและอัลกอริทึมที่ใช้ได้
- เพื่อให้นักศึกษาสามารถประยุกต์ใช้งานโครงสร้างข้อมูลรวมถึงอัลกอริทึมที่เหมาะสมให้เข้ากับการใช้งานโครงสร้างข้อมูลนั้นๆ ได้

ประมวลรายวิชา (ต่อ)

เนื้อหาวิชาโดยละเอียด

1. ความรู้เบื้องต้นเกี่ยวกับการเขียนโปรแกรม
2. ประสิทธิภาพของโครงสร้างข้อมูล
3. คณิตศาสตร์พื้นฐาน
4. การวิเคราะห์อัลกอริทึม
5. ลิสต์, สแต็ก, คิว (รายการ, กองซ้อน, แถวคอย)
6. ทรี (ต้นไม้)
7. การเรียงลำดับ
8. การค้นหา
9. กราฟ
10. การวิเคราะห์อัลกอริทึมขั้นสูง
11. Divide and Conquer
12. Dynamic Programming
13. Greedy Algorithms
14. Branch and Bound
15. Probabilistic Algorithms
16. NP-Completeness

ประมวลรายวิชา (ต่อ)

ตำราเรียนและหนังสืออ่านประกอบ

- Data Structures & Algorithm Analysis by Clifford A. Shaffer
 - <http://people.cs.vt.edu/shaffer/Book/>
 - ดาวน์โฮลดได้ฟรี
- โครงสร้างข้อมูลแบบจาวา โดย สมชาย ประสิทธิ์จูตระกูล
 - <https://www.cp.eng.chula.ac.th/~somchai/ULearn/DataStructures/>
 - อ่านได้ออนไลน์ฟรี
- Introduction to Algorithms 4 nd edition, T. Cormen, C. Leiserson, R. Rivest, C. Stein, MIT Press & McGraw-Hill, 2020.
- การออกแบบและวิเคราะห์อัลกอริทึม สมชาย ประสิทธิ์จูตระกูล

ประมวลรายวิชา (ต่อ)

- การประเมิน

- คะแนนเก็บ 20 คะแนน
- สอบกลางภาค 40 คะแนน
- สอบปลายภาค 40 คะแนน
- โบนัส
 - คะแนนเกินเกณฑ์ให้

- เกณฑ์โดยประมาณ

- A 85
- B+ 80
- B 75
- C+ 70
- C 65
- D+ 60
- ต่ำกว่านั้น F เลย ไม่มี D

ประวัติผู้สอน

อ.ดร.วรินทร์ วัฒนพรพรหม

E-Mail : warin.wat@kmutt.ac.th

Tel : 095-495-6245

รองชนะเลิศอันดับที่สองการประกวดซอฟต์แวร์ระดับชาติครั้งที่หนึ่ง
ประเภทซอฟต์แวร์เกม บัณฑิตเกาเทปออกอากาศช่อง 11 ปี 2542



Research Area:

- Multimodal and Multi-Objective Optimization
- Multiclass and Multilabel Classification
- Cloud and Fog Computing
- Evolvable and Explainable AI
- Bioinformatics and Computational Biology
- Distributed Autonomous Mobile Robots
- AI for Game Design



Education:

- PhD Computer Engineering
- MS Computer Science
- BSc Information Technology
- BA Information Science
- BBA Marketing
- BLit Chinese for Economy and Trade

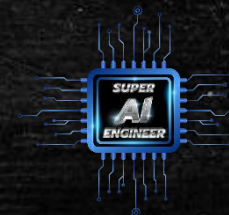


Warin Wattanapornprom PhD.

A Lecturer, a CTO and a CIO

A Data Engineer, a Data Scientist and an AI Engineer

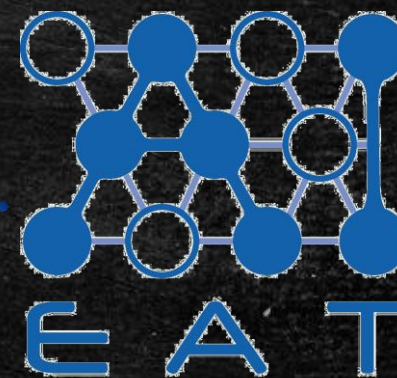
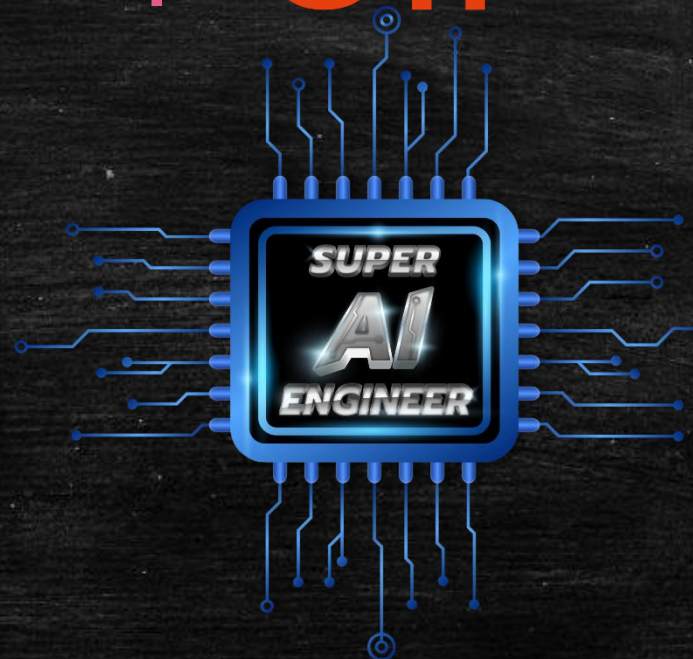
An ERP Developer and a Digital Marketer



ประวัติผู้สอน (ต่อ) ตำแหน่ง

- บริษัท ไทยโอเชียนอุตสาหกรรม จำกัด
หัวหน้าส่วนวิจัยและพัฒนา ผลิตภัณฑ์ และกระบวนการ
หัวหน้าฝ่ายพัฒนาระบบเทคโนโลยีสารสนเทศ
- บริษัท เซฟตี้แบรด์ 555 ออโต้ จำกัด บริษัทไทยวาพลาสติก จำกัด และ
บริษัท วีซีเคอินดัสเตรียสไลน์ จำกัด
ที่ปรึกษาระบบสารสนเทศ
- บริษัท อิลิกเซอร์ซอฟต์แวร์ จำกัด ไทพ์สตูดิโอ และ บริษัท อินโนเซนต์ โซลูชั่น จำกัด
ผู้ก่อตั้ง
- หน่วยปฏิบัติการวิจัยบริหารอุตสาหกรรมและเทคโนโลยี จุฬาลงกรณ์มหาวิทยาลัย
นักวิจัย
วิทยากรพิเศษ
- ผู้ทรงคุณวุฒิ
กรรมการการประกวดซอฟต์แวร์แห่งชาติครั้งที่10-16 ปี 2550 - 2559

ประวัติผู้สอน (ต่อ) ตำแหน่ง



ประวัติผู้สอน (ต่อ) การศึกษา

- วิศวกรรมศาสตรดุษฎีบัณฑิต
สาขาวิศวกรรมคอมพิวเตอร์
จุฬาลงกรณ์มหาวิทยาลัย
- บริหารธุรกิจบัณฑิต
สาขาวิทยาการจัดการ เอกการตลาด
มหาวิทยาลัยสุโขทัยธรรมมาธิราช
- ศิลปศาสตรบัณฑิต
สาขาสารสนเทศศาสตร์ เอกสารสนเทศสำนักงาน
มหาวิทยาลัยสุโขทัยธรรมมาธิราช
- วิทยาศาสตร์มหาบัณฑิต
สาขาวิทยาศาสตร์คอมพิวเตอร์
จุฬาลงกรณ์มหาวิทยาลัย
- วิทยาศาสตร์บัณฑิต
สาขาเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีนานาชาติสิรินธร
มหาวิทยาลัยธรรมศาสตร์
- ศิลปศาสตรบัณฑิต
สาขาภาษาจีนเพื่อการค้าระหว่างประเทศ
มหาวิทยาลัยภาษาและวัฒนธรรมปักกิ่ง ประเทศจีน

ประวัติผู้สอน (ต่อ) วิจัย

- Multimodal and Multi-Objective Optimization
- Multiclass and Multilabel Classification
- Cloud and Fog Computing
- Evolvable and Explainable AI
- Bioinformatics and Computational Biology
- Distributed Autonomous Mobile Robots
- AI for Game Design

คุยกันก่อนเรียน

ทบทวนสิ่งที่เรียนมาเกี่ยวกับคอมพิวเตอร์

- คอมพิวเตอร์ประกอบด้วยหน่วยประมวลผล <CPU> หน่วยความจำ <Memory> และอื่นๆ
- มองให้ลึกเข้าไปคอมพิวเตอร์ยังมีส่วนประกอบที่เรียกว่า Context Switch ซึ่งทำหน้าที่แบ่งเวลาให้กับนาฬิกาภายในเครื่อง
- โปรแกรมแบ่งเป็นระบบปฏิบัติการ OS และโปรแกรมรรถประโยชน์ App
- ข้อมูลส่วนเล็กที่สุดคือ 0 กับ 1 และ 0 กับ 1 นี้เองที่ประกอบกันเป็นทุกสิ่งทุกอย่างที่เราเห็น
- เราประกอบ 0 กับ 1 นี้ขึ้นมาเอง

กติกาของวิชานี้

- Font ถ้าเป็นภาษาไทยผสมอังกฤษใช้ TH Sarabun
- Font ถ้าเป็นภาษาอังกฤษล้วนใช้ **Time**
- Font ถ้าเป็นภาษาโปรแกรมมิ่ง ใช้ Courier New
- Font ถ้าเป็นตัวแปรหรือสมการทางคณิตศาสตร์ใช้ *Time* ตัวเอียง, **Math**

ฮาร์ดแวร์คอมพิวเตอร์

1. หน่วยประมวลผลกลาง CPU (Central Processing Unit)
2. หน่วยความจำหลัก Primary storage (memory)
3. หน่วยความจำสำรอง Secondary storage (disks, tapes etc.)
4. อุปกรณ์รับข้อมูล Input devices (mouse, keyboard etc.)
5. อุปกรณ์แสดงผล Output devices (screen, printer, plotter, etc.)
6. หน่วยประมวลผลเฉพาะทาง เช่น GPU (Graphic Processing Unit)

ข้อมูลในระบบคอมพิวเตอร์

ลักษณะของข้อมูลจะเป็นบิต (Bit) มาประกอบกัน ข้อมูล 1 บิตมีเพียงสองสถานะ ใช้สัญลักษณ์ 1 หรือ 0 แทน ซึ่งตรงกับเลขฐานสอง (Binary number)

โดยทั่วไปข้อมูลหนึ่งตัวอักษร (Character) หรือเรียกว่า 1 ไบต์(Byte) นั้นประกอบด้วยข้อมูล 8 บิต ดังนั้นข้อมูล 1 ไบต์มีความแตกต่างกัน 2^8 หรือ 256 แบบ

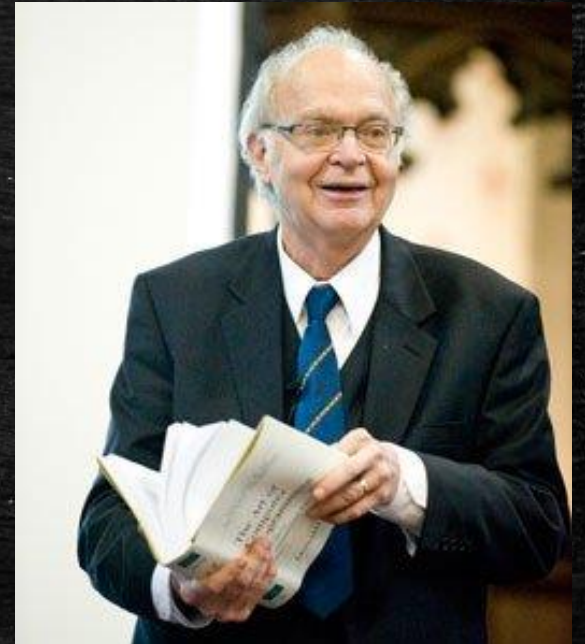
- bit = **binary digit** (เลขฐานสอง)
 - มีค่าเป็น 0 หรือ 1 (ตามสถานะของความต่างศักย์ไฟฟ้าดิจิทัล)
- byte = 8 bits

ข้อมูล และ โครงสร้าง

- ข้อมูล (Data) หมายถึง ข้อเท็จจริงต่างๆ ซึ่งอาจจะเป็นตัวเลขหรือไม่เป็นก็ได้
- โครงสร้าง (Structure) หมายถึง ความสัมพันธ์ของสมาชิกในกลุ่ม

“Computer programming is an art, because it applies accumulated knowledge to the world, because it requires skill and ingenuity, and especially because it produces objects of beauty.”

- Donald Knuth



[This Photo](#) by Unknown Author is licensed under [CC BY-NC](#)

ความหมายของโครงสร้างข้อมูล

- โครงสร้างข้อมูล หมายถึง การรวมประเภทข้อมูล (data type) เข้าไว้ด้วยกันจนกระทั่งกลายเป็นประเภทข้อมูลและมีกำหนดคำนิยามของความสัมพันธ์ภายในกลุ่มข้อมูลไว้อย่างชัดเจน ซึ่งการรวมกลุ่มนั้นอาจจะเป็นการรวมกลุ่มระหว่างข้อมูลประเภทเดียวกัน, ต่างประเภทหรือต่างโครงสร้างกันก็ได้
(ขนิษฐา นามิ, 2548, หน้า 3)
- โครงสร้างข้อมูล หมายถึง รูปแบบการจัดระเบียบของข้อมูลที่ได้จากการดำเนินการทางคณิตศาสตร์ (Operation) เพื่อให้สามารถจัดการกับข้อมูลที่ใช้กับระบบคอมพิวเตอร์ได้
(วิวัฒน์ อภิสิตธิภูโย, อมร มุกสิกसार, 2548, หน้า 12)

ความหมายของโครงสร้างข้อมูล

- สรุปรูป โครงสร้างข้อมูล หมายถึงการจัดรูปแบบข้อมูลหรือวิธีการจัดระบบข้อมูลเข้าด้วยกัน มีการกำหนดนิยามความสัมพันธ์ไว้อย่างชัดเจนภายในกลุ่ม เพื่อให้ข้อมูลมีลักษณะที่เหมาะสมสำหรับใช้ในระบบคอมพิวเตอร์

ประเภทของโครงสร้างข้อมูล

- โครงสร้างข้อมูลแบบเป็นเชิงเส้น
(Linear Data Structure)
- โครงสร้างข้อมูลแบบไม่เป็นเชิงเส้น
(Non Linear Data Structure)

ประเภทของโครงสร้างข้อมูล

โครงสร้างข้อมูลแบบเชิงเส้น

โครงสร้างที่มีการจัดเก็บข้อมูลในลักษณะต่อเนื่องกัน ถ้าทราบตำแหน่งแรก
ของข้อมูลก็สามารถทราบตำแหน่งข้อมูลตัวถัดไปหรือข้อมูลตัวอื่นได้ ทาง
คณิตศาสตร์จะเรียกว่า เวกเตอร์ (Vector) ข้อมูลมีลักษณะเป็น 1 มิติ เช่น
อาร์เรย์, สแต็ก, คิว, ลิสต์

ประเภทของโครงสร้างข้อมูล

100	200	300	400
-----	-----	-----	-----

D(1) D(2) D(3) D(4)

อาร์เรย์ (Array)

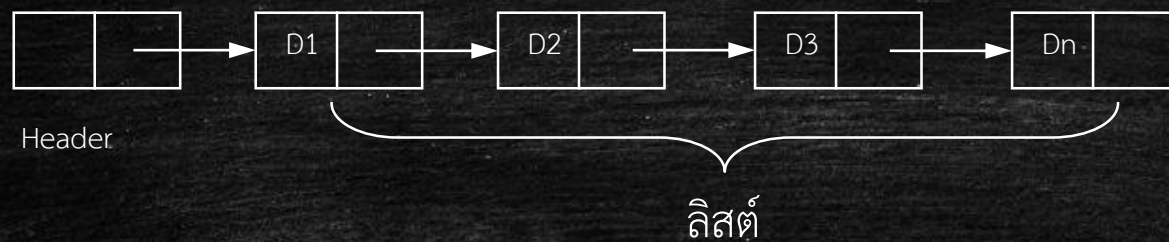


โครงสร้างของสแต็ก

ประเภทของโครงสร้างข้อมูล



คิว (queue)



โครงสร้างของลิสต์

ประเภทของโครงสร้างข้อมูล

โครงสร้างข้อมูลแบบไม่เชิงเส้น

โครงสร้างที่ไม่มีคุณสมบัติของเชิงเส้น สามารถใช้แสดงความสัมพันธ์ของข้อมูลที่ซับซ้อนได้มากกว่าโครงสร้างข้อมูลแบบเชิงเส้น เช่น ทรี กราฟ

การเลือกใช้โครงสร้างข้อมูล

ในการเลือกใช้โครงสร้างข้อมูลแบบใดนั้น จะต้องคำนึงถึง

1. โครงสร้างของข้อมูลนั้นสามารถสร้าง**ความสัมพันธ์**ให้กับชุดข้อมูลนั้นได้**สมบูรณ์ที่สุด**
2. โครงสร้างนั้นต้อง**ง่าย**ต่อการดำเนินการในระบบ
3. ต้นทุน**เวลา**
4. ต้นทุน**การจัดเก็บ**

ความหมายอัลกอริธึม

- อัลกอริธึม (Algorithms) หมายถึง การอธิบายขั้นตอนการทำงานของโปรแกรมที่จะพัฒนาขึ้น ซึ่งผ่านการวิเคราะห์และแยกแยะงาน เพื่อแก้ปัญหาใดปัญหาหนึ่ง โดยอธิบายอย่างเป็นลำดับ ขั้นตอน ให้ผู้เป็นเจ้าของหรือผู้รับผิดชอบได้ตรวจสอบความถูกต้องในแต่ละขั้นตอนของโปรแกรมได้ โดยที่ผู้เป็นเจ้าของหรือผู้รับผิดชอบไม่จำเป็นต้องเขียนโปรแกรม (วุฒิพงษ์ เขื่อนดิน, โครงสร้างข้อมูลและอัลกอริทึม, หน้า 3)
- ซึ่งถ้าปฏิบัติตามขั้นตอนอย่างถูกต้องแล้ว จะต้องสามารถช่วยแก้ปัญหาหรือประมวลผลตามต้องการได้สำเร็จ

อัลกอริธึม

- ในการเขียนอธิบายอัลกอริธึมสามารถเขียนได้หลายแบบ เช่น ในรูปของ Flow Chart, Pseudo และภาษาคอมพิวเตอร์
- ในปัญหาเดียวกันเราสามารถคิดอัลกอริธึมเพื่อมาแก้ปัญหาได้หลายแบบ ซึ่งในแต่ละแบบ เครื่องคอมพิวเตอร์ก็จะใช้หน่วยความจำ และเวลาในการประมวลผลไม่เท่ากัน ดังนั้น การเปรียบเทียบว่าโปรแกรมคอมพิวเตอร์ใดเก่งกว่ากันนั้นจึงใช้ในการเปรียบเทียบประสิทธิภาพของอัลกอริธึมนั่นเอง

ประสิทธิภาพของอัลกอริธึม

- จะพิจารณาอยู่ 2 ส่วนหลักๆ ดังนี้
 - หน่วยความจำ (memory) ที่จะต้องใช้ในการประมวลผล
 - เวลา (time) ที่ใช้ในการประมวลผล
- โปรแกรมสองโปรแกรมให้คำตอบเหมือนกันแต่อาจใช้เวลาต่างกันเป็นวันเป็นปี

ภาษา C++ พื้นฐาน

- โครงสร้างหลัก

```
#include <iostream>
using namespace std;

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

ภาษา C++ พื้นฐาน

- Semicolons จุดสิ้นสุดคำสั่ง

```
x = y;  
y = y+1;  
add(x, y);
```

- Blocks รวบรวมคำสั่งเป็นก้อน

```
{  
    cout << "Hello World";  
    return 0;  
}
```

ภาษา C++ พื้นฐาน

■ Keywords ห้ามตั้งชื่อดังนี้

asm	else	new	this
auto	enum	operator	throw
bool	explicit	private	true
break	export	protected	try
case	extern	public	typedef
catch	false	register	typeid
char	float	reinterpret	typename
class	for	return	union
const	friend	short	unsigned
const_cast	goto	signed	using
continue	if	sizeof	virtual
default	inline	static	void
delete	int	static_cast	volatile
do	long	struct	wchar_t
double	mutable	switch	while
dynamic_cast	namespace	template	

ภาษา C++ พื้นฐาน

Comments การแทรกโน้ตใน Code

- แบบมีขอบเขต

```
/*This is a comment */  
  
/* C++ comments can also  
span multiple lines */
```

- แบบในบรรทัด

```
cout << "Hello World"; //prints Hello World
```

ภาษา C++ พื้นฐาน

Comments การแทรกโน้ตใน Code

- การซ่อน Code ด้วย Comment

```
/*Comment out printing of Hello World:  
cout << "Hello World"; //prints Hello World  
*/
```

ภาษา C++ พื้นฐาน

- Data type ดั้งเดิม

Type	Keyword
Boolean	bool
Character	char
Integer	int
Floating point	float
Double floating point	double
Valueless	void

ภาษา C++ พื้นฐาน

- Data type พื้นฐาน

Type	Typical Bit Width	Typical Range
char	1byte	-128 to 127 or 0 to 255
unsigned char	1byte	0 to 255
signed char	1byte	-128 to 127
int	4bytes	-2147483648 to 2147483647
unsigned int	4bytes	0 to 4294967295
signed int	4bytes	-2147483648 to 2147483647
short int	2bytes	-32768 to 32767

Type	Typical Bit Width	Typical Range
unsigned short int	2bytes	0 to 65,535
signed short int	2bytes	-32768 to 32767
long int	4bytes	-2,147,483,648 to 2,147,483,647
signed long int	4bytes	-2,147,483,648 to 2,147,483,647
unsigned long int	4bytes	0 to 4,294,967,295
float	4bytes	+/- 3.4e +/- 38 (~7 digits)
double	8bytes	+/- 1.7e +/- 308 (~15 digits)
long double	8bytes	+/- 1.7e +/- 308 (~15 digits)

ภาษา C++ พื้นฐาน

- Data type ดั้งเดิม

```
#include <iostream>
using namespace std;
int main()
{
    cout << "Size of char :" << sizeof(char) << endl;
    cout << "Size of int :" << sizeof(int) << endl;
    cout << "Size of short int :" << sizeof(short int) << endl;
    cout << "Size of long int :" << sizeof(long int) << endl;
    cout << "Size of float :" << sizeof(float) << endl;
    cout << "Size of double :" << sizeof(double) << endl;
    return 0;
}
```

ภาษา C++ พื้นฐาน

- Data type ดั้งเดิม

Size of char : 1

Size of int : 4

Size of short int : 2

Size of long int : 8

Size of float : 4

Size of double : 8

ภาษา C++ พื้นฐาน

- การแจกแจง variable
- `type variable_list;`

```
int    i, j, k;  
char   c, ch;  
float  f, salary;  
double d;
```

ภาษา C++ พื้นฐาน

- การกำหนดค่า variable สามารถทำในขณะแจกแจง
- `type variable_name = value;`

```
int d = 3, f = 5;  
char x = 'x';
```

ภาษา C++ พื้นฐาน

- ตัวอย่าง

```
#include <iostream>
using namespace std;
int main()
{
    int a, b;
    int c; float f;
    a = 10;
    b = 20;
    c = a + b;
    cout << c << endl ;
    f = 70.0/3.0;
    cout << f << endl ;
    return 0;
}
```

ภาษา C++ พื้นฐาน

- Typedef การตั้งชื่อ type ใหม่เพื่อใช้งานเอง
- `typedef type newname;`

```
typedef int feet;  
feet distance;
```

ภาษา C++ พื้นฐาน

- Enumerated Types
ตั้ง Type เองโดยมีค่าเป็นสมาชิกในเซต
- `enum enum-name { list of names } var-list;`

```
enum color { red, green, blue } c;  
c = blue;
```

ภาษา C++ พื้นฐาน

- Local และ Global Variable

```
#include <iostream>
using namespace std;
int g;
int main ()
{
    int a, b;
    a = 10;
    b = 20;
    g = a + b;
    cout << g;
    return 0;
}
```

ภาษา C++ พื้นฐาน

- Arithmetic Operators

- Assume variable A holds 10 and variable B holds 20, then:

Operator	Description	Example
+	Adds two operands	A + B will give 30
-	Subtracts second operand from the first	A - B will give -10
*	Multiplies both operands	A * B will give 200
/	Divides numerator by de-numerator	B / A will give 2
%	Modulus Operator and remainder of after an integer division	B % A will give 0
++	Increment operator , increases integer value by one	A++ will give 11
--	Decrement operator , decreases integer value by one	A-- will give 9

ภาษา C++ พื้นฐาน

- Relational Operators

- Assume variable A holds 10 and variable B holds 20, then:

Operator	Description	Example
==	Checks if the values of two operands are equal or not, if yes then condition becomes true.	(A == B) is not true.
!=	Checks if the values of two operands are equal or not, if values are not equal then condition becomes true.	(A != B) is true.
>	Checks if the value of left operand is greater than the value of right operand, if yes then condition becomes true.	(A > B) is not true.
<	Checks if the value of left operand is less than the value of right operand, if yes then condition becomes true.	(A < B) is true.
>=	Checks if the value of left operand is greater than or equal to the value of right operand, if yes then condition becomes true.	(A >= B) is not true.
<=	Checks if the value of left operand is less than or equal to the value of right operand, if yes then condition becomes true.	(A <= B) is true.

ภาษา C++ พื้นฐาน

- Logical Operators
 - Assume variable A holds 1 and variable B holds 0, then:

Operator	Description	Example
&&	Called Logical AND operator. If both the operands are non-zero, then condition becomes true.	(A && B) is false.
	Called Logical OR Operator. If any of the two operands is non-zero, then condition becomes true.	(A B) is true.
!	Called Logical NOT Operator. Use to reverses the logical state of its operand. If a condition is true, then Logical NOT operator will make false.	!(A && B) is true.

ภาษา C++ พื้นฐาน

- Assignment Operators

- Assume variable A holds 1 and variable B holds 0, then:

Operator	Description	Example
=	Simple assignment operator, Assigns values from right side operands to left side operand	<code>C = A + B</code> will assign value of <code>A + B</code> into <code>C</code>
+=	Add AND assignment operator, It adds right operand to the left operand and assign the result to left operand	<code>C += A</code> is equivalent to <code>C = C + A</code>
-=	Subtract AND assignment operator, It subtracts right operand from the left operand and assign the result to left operand	<code>C -= A</code> is equivalent to <code>C = C - A</code>
*=	Multiply AND assignment operator, It multiplies right operand with the left operand and assign the result to left operand	<code>C *= A</code> is equivalent to <code>C = C * A</code>
/=	Divide AND assignment operator, It divides left operand with the right operand and assign the result to left operand	<code>C /= A</code> is equivalent to <code>C = C / A</code>

ลำดับการทำงาน

ลำดับความสำคัญ	เครื่องหมาย	การดำเนินการ
1	()	Left to Right
2	!, ++, --, (typecast)	<u>Right to Left</u>
3	*, /, %	Left to Right
4	+, -	Left to Right
5	<, <=, >, >=	Left to Right
6	=, !=	Left to Right
7	&&	Left to Right
8		Left to Right
9	*=, /=, %=, +=, -=	Left to Right

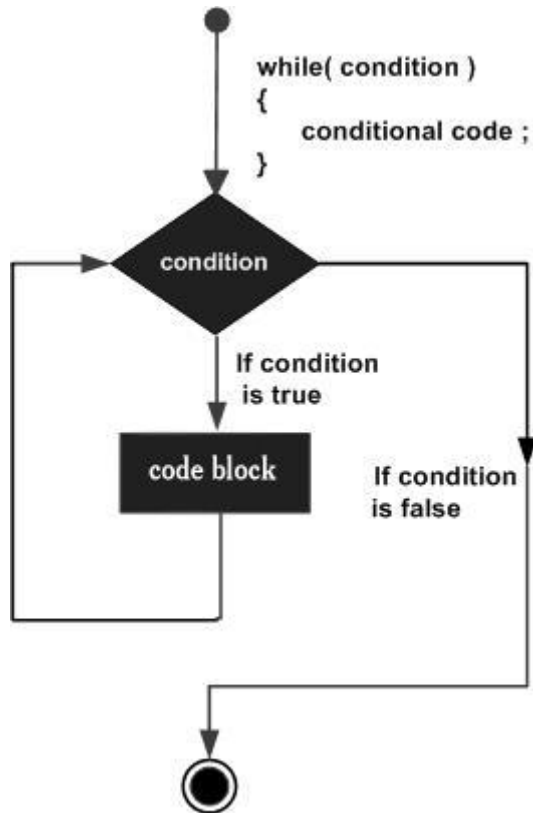
ในกรณีที่มีการคำนวณที่ซับซ้อน จะต้องพิจารณาลำดับการคำนวณ โดยสัญลักษณ์แต่ละตัว จะมีลำดับการคำนวณไม่เท่ากัน

ภาษา C++ พื้นฐาน

- เงื่อนไข และการตั้งค่า
- การทดสอบเงื่อนไข
 - ใช้เครื่องหมาย \geq , $>$, $=$, \leq , $<$
- การตั้งค่า
 - ใช้เครื่องหมาย $=$ เดียว

ภาษา C++ พื้นฐาน

- while loop



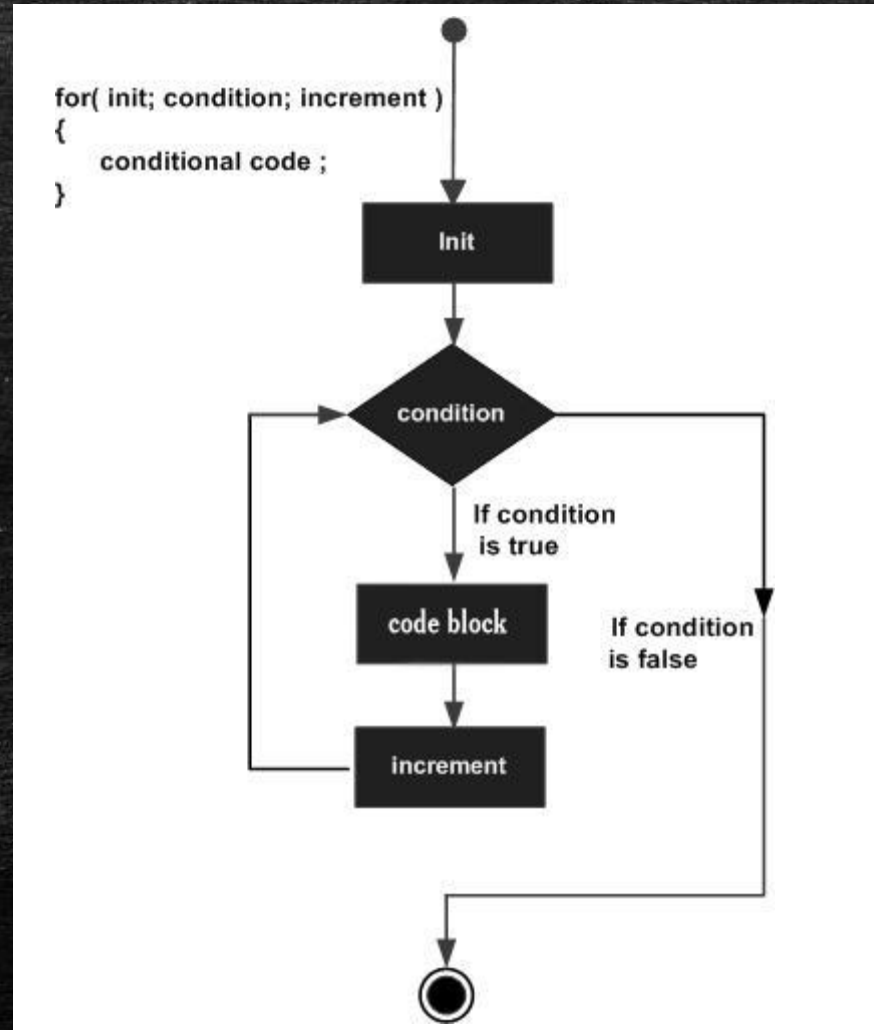
ภาษา C++ พื้นฐาน

- while loop

```
int i=10;
while (i>=1)
{
    cout << "Hello world";
    i--;
}
```

ภาษา C++ พื้นฐาน

- For loop



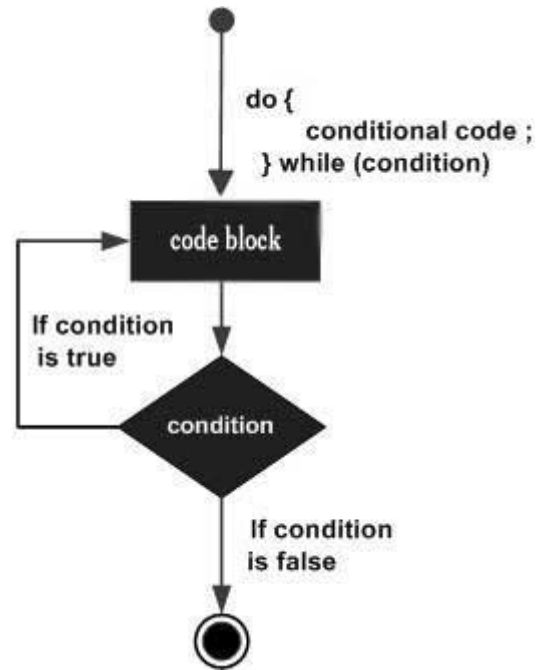
ภาษา C++ พื้นฐาน

- For loop

```
for (i=0;i<=n;i++)  
{  
    cout << "Hello" << i <<endl;  
}  
for (i=n;i>=0;i--)  
{  
    cout << "Hello" << i <<endl;  
}
```

ภาษา C++ พื้นฐาน

- Do...While loop



ภาษา C++ พื้นฐาน

- Do...While loop

```
int i = 1;
do
{
    cout << "Hello" << i << endl;
    i = i + 1;
} while(i < 20 );
```

ภาษา C++ พื้นฐาน

- The infinite loop

```
for( ; ; )  
{  
    cout<< "This loop will run forever " <<endl;  
}  
for(i=1;i>=0;i++) //fake infinite loop  
{  
    cout<< "I love you " <<endl;  
} //integer overflow
```

จีบสาว/หนุ่มเน็ต 101

- The infinite loop

```
#include <iostream>
#define ever ;;
using namespace std;

int main ()
{
    for(ever)
    {
        cout<< "I love you." <<endl;
    }
    return 0;
}
```

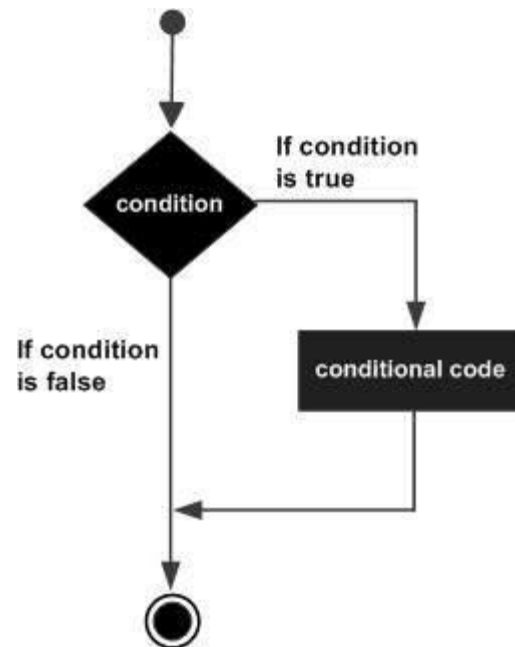
ภาษา C++ พื้นฐาน

- เงื่อนไข if

```
#include <iostream>
using namespace std;
int main () {
    int a = 10;
    if( a < 20 ) {
        cout << "a is less than 20;" << endl;
    }
    cout << "value of a is :"<< a << endl;
    return 0;
}
```

ภาษา C++ พื้นฐาน

- เงื่อนไข if



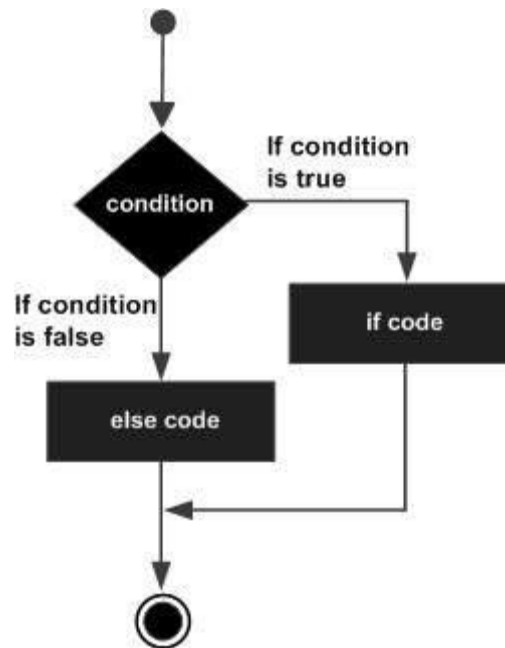
ภาษา C++ พื้นฐาน

- เงื่อนไข if...else

```
if( a < 20 ){  
    cout << "a is less than 20;" << endl;  
}else{  
    cout << "a is not less than 20;" << endl;  
}
```

ภาษา C++ พื้นฐาน

- เงื่อนไข if...else



ภาษา C++ พื้นฐาน

■ เงื่อนไข switch

```
char grade = 'D';
switch(grade){
    case 'A' :
        cout << "Excellent!" << endl;
        break;
    case 'B' :
    case 'C' :
        cout << "Well done" << endl;
        break;
    case 'D' :
        cout << "You passed" << endl;
        break;
    case 'F' :
        cout << "Try again" << endl;
        break;
    default :
        cout << "Invalid grade" << endl;
```

ภาษา C++ พื้นฐาน

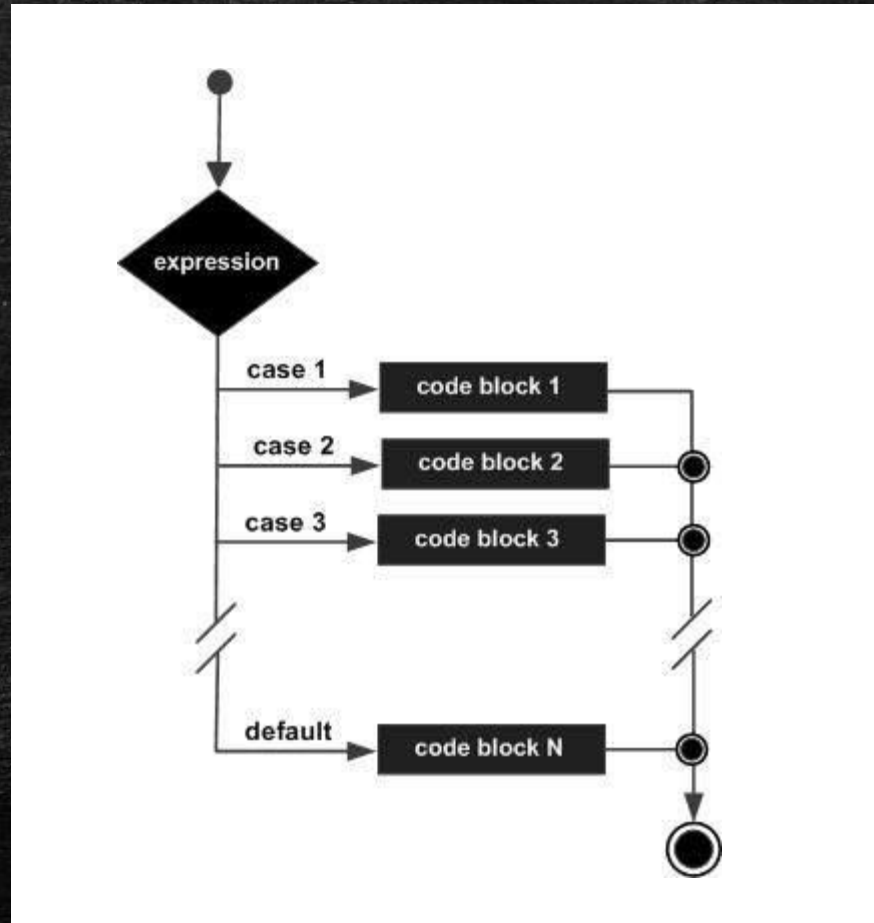
- goto และ label

```
char grade = 'D';

Hell:
    cout << "this is hell" ;
    if (grade=='D')
        goto Hell;
    else
        goto Heaven;
Heaven:
    cout << "this is heaven" ;
}
```

ภาษา C++ พื้นฐาน

- เงื่อนไข switch



ภาษา C++ พื้นฐาน

- การนิยามฟังก์ชัน
- `return_type function_name(parameter list)`
 {
 body of the function
 }

ตัวอย่าง

```
int max(int num1, int num2) {  
    int result;  
    if (num1 > num2)  
        result = num1;  
    else  
        result = num2;  
    return result;  
}
```

ภาษา C++ พื้นฐาน

- การเรียกใช้ฟังก์ชัน

```
#include <iostream>
using namespace std;
int max(int num1, int num2);

int main () {
    int a = 100;
    int b = 200;
    int ret;
    ret = max(a, b);
    cout << "Max value is :" << ret << endl;
    return 0;
}
```

- Max value is : 200

ภาษา C++ พื้นฐาน

- mathematical operations

```
#include <iostream>
#include <cmath>           //calling cmath
using namespace std;
void main () {
    short s = 10;
    int i = -1000;
    long l = 100000;
    float f = 230.47;
    double d = 200.374;

    cout << "sin(d):" << sin(d) << endl;
    cout << "abs(i):" << abs(i) << endl;
    cout << "floor(d):" << floor(d) << endl;
    cout << "sqrt(f):" << sqrt(f) << endl;
    cout << "pow(d,2):" << pow(d, 2) << endl;
}
```

ภาษา C++ พื้นฐาน

- mathematical operations
- `sin(d) : -0.634939`
`abs(i) : 1000`
`floor(d) : 200`
`sqrt(f) : 15.1812`
`pow(d,2) : 40149.7`

ภาษา C++ พื้นฐาน

- Random Number

```
#include <iostream>
#include <ctime>
#include <cstdlib>
using namespace std;

void main () {
    int i,j;
    srand( (unsigned)time( NULL ) ); // set the seed
    for( i = 0; i < 10; i++ ){ // generate 10 random numbers.
        j= rand();
        cout <<" Random Number :" << j << endl;
    }
}
```

ภาษา C++ พื้นฐาน

- Random Number
- Random Number : 1748144778
Random Number : 630873888
Random Number : 2134540646
Random Number : 219404170
Random Number : 902129458
Random Number : 920445370
Random Number : 1319072661
Random Number : 257938873
Random Number : 1256201101
Random Number : 580322989

ภาษา C++ พื้นฐาน

- ตัวแปรแบบอาร์เรย์ (Array) หมายถึงตัวแปรซึ่งมีค่าได้หลายค่าโดยใช้ชื่ออ้างอิงเพียงชื่อเดียว ด้วยการใช้หมายเลขลำดับเป็นตัวจำแนกความแตกต่างของค่าตัวแปรแต่ละตัว ถ้าเราจะเรียกตัวแปรชนิดนี้ว่า "ตัวแปรชุด" ก็เห็นจะไม่ผิดนัก ตัวแปรชนิดนี้มีประโยชน์มาก
- ลองคิดถึงค่าข้อมูลจำนวน 100 ค่า ที่ต้องการเก็บไว้ในตัวแปรจำนวน 100 ตัว อาจทำให้ต้องกำหนดตัวแปรที่แตกต่างกันมากถึง 100 ชื่อ กรณีอย่างนี้ควรจะทำอย่างไรดี

ภาษา C++ พื้นฐาน

- ด้วยการใช้คุณสมบัติอาร์เรย์ เราสามารถนำตัวแปรหลาย ๆ ตัวมาอยู่รวมเป็นชุดเดียวกันได้ และสามารถเรียกใช้ตัวแปรทั้งหมดโดยระบุผ่านชื่อเพียงชื่อเดียวเท่านั้น ด้วยการระบุหมายเลขลำดับ หรือ ดัชนี(index) กำกับตามหลังชื่อตัวแปร ตัวแปรเพียงชื่อเดียวจึงมีความสามารถเทียบได้กับตัวแปรนับร้อยตัว พันตัว
- อาร์เรย์ใช้จองเนื้อที่ในหน่วยความจำ ตามที่ระบุขนาดหน่วยความจำในตัวแปร อาร์เรย์ ใช้ตัวระบุตำแหน่งเพื่อคำนวณระยะห่างไปของหน่วยความจำ

ภาษา C++ พื้นฐาน

- Array
- Declaring
 - `type arrayName [arraySize];`
- example

```
double balance[10];
```

- Declaring + initialization

```
double balance[5] = {1000.0, 2.0, 3.4, 17.0, 50.0};
```

- A

```
double balance[] = {1000.0, 2.0, 3.4, 17.0, 50.0};
```

- ```
balance[4] = 50.0;
```

# ภาษา C++ พื้นฐาน

---

- Array

```
double balance[5] = {1000.0, 2.0, 3.4, 17.0, 50.0};
```

|         | 0      | 1   | 2   | 3   | 4    |
|---------|--------|-----|-----|-----|------|
| balance | 1000.0 | 2.0 | 3.4 | 7.0 | 50.0 |

# ภาษา C++ พื้นฐาน

- Array

```
#include <iostream>
using namespace std;
#include <iomanip>
using std::setw;

void main () {
 int n[10]; // n is an array of 10 integers
 // initialize elements of array n to 0
 for (int i=0; i<10; i++) {
 n[i] = i + 100;
 }
 cout << "Element" << setw(13) << "Value" << endl;
 for (int j=0; j<10; j++) {
 cout << setw(7) << j << setw(13) << n[j] << endl;
 }
}
```

# ภาษา C++ พื้นฐาน

---

- Array

- Output

| – Element | Value |
|-----------|-------|
| 0         | 100   |
| 1         | 101   |
| 2         | 102   |
| 3         | 103   |
| 4         | 104   |
| 5         | 105   |
| 6         | 106   |
| 7         | 107   |
| 8         | 108   |
| 9         | 109   |

# ภาษา C++ พื้นฐาน

- Multidimensional Array
- `type name[size1][size2]...[sizeN];`

```
int threedim[5][10][4];
```

```
int x=4;
```

```
int y=4;
```

```
type arrayName[x][y];
```

|       | Column 0 | Column 1 | Column 2 | Column 3 |
|-------|----------|----------|----------|----------|
| Row 0 | a[0][0]  | a[0][1]  | a[0][2]  | a[0][3]  |
| Row 1 | a[1][0]  | a[1][1]  | a[1][2]  | a[1][3]  |
| Row 2 | a[2][0]  | a[2][1]  | a[2][2]  | a[2][3]  |

# ภาษา C++ พื้นฐาน

---

- Multidimensional Array

```
int a[3][4] = {
 {0, 1, 2, 3} , /* row indexed by 0 */
 {4, 5, 6, 7} , /* row indexed by 1 */
 {8, 9, 10, 11} /* row indexed by 2 */
};
```

# ภาษา C++ พื้นฐาน

- Multidimensional Array

```
#include <iostream>
using namespace std;
void main () {

 int a[5][2]={{0,0},{1,2},{2,4},{3,6},{4,8}};

 for(int i=0;i<5;i++)
 for(int j=0;j<2;j++){
 cout << "a[" << i << "][" << j << "]: ";
 cout << a[i][j]<< endl;
 }
}
```

# ภาษา C++ พื้นฐาน

---

- Multidimensional Array

- Output

- ```
a[0][0]: 0
a[0][1]: 0
a[1][0]: 1
a[1][1]: 2
a[2][0]: 2
a[2][1]: 4
a[3][0]: 3
a[3][1]: 6
a[4][0]: 4
a[4][1]: 8
```

ภาษา C++ พื้นฐาน

- การบ้าน โบนัส 2 คะแนน ให้เขียนโปรแกรมสร้างแผนที่ขนาด 10x10 ด้วย array แล้วพิมพ์แผนที่ออกมา กำหนดให้
 - 0 แทนพื้นดินโล่งๆ
 - 1 แทนต้นไม้
 - 2 แทนฮีโร่
 - 3 แทนมอนสเตอร์
- จงคำนวณระยะทางจาก Hero ไปถึง Monster

ภาษา C++ พื้นฐาน

- 1 แทนต้นไม้
- 2 แทนอีโร่
- 3 แทนมอนสเตอร์

วัตถุ	X	Y
ต้นไม้	ตำแหน่งใดๆ	ตำแหน่งใดๆ
อีโร่	วันเกิด 1 ถึง 7	เลขท้ายรหัสนักศึกษา
มอนสเตอร์	หมายเลข 10 - วัน เกิด	รหัสนักศึกษาหลักที่ สองจากสุดท้าย

Distance

- การวัดระยะแบบยูคลิด (Euclidean distance)
- การวัดระยะแบบแมนฮัตตัน (Manhattan distance)
- การวัดระยะแบบเชบิเชฟ (Chebychev distance)

Euclidean distance

- One dimension

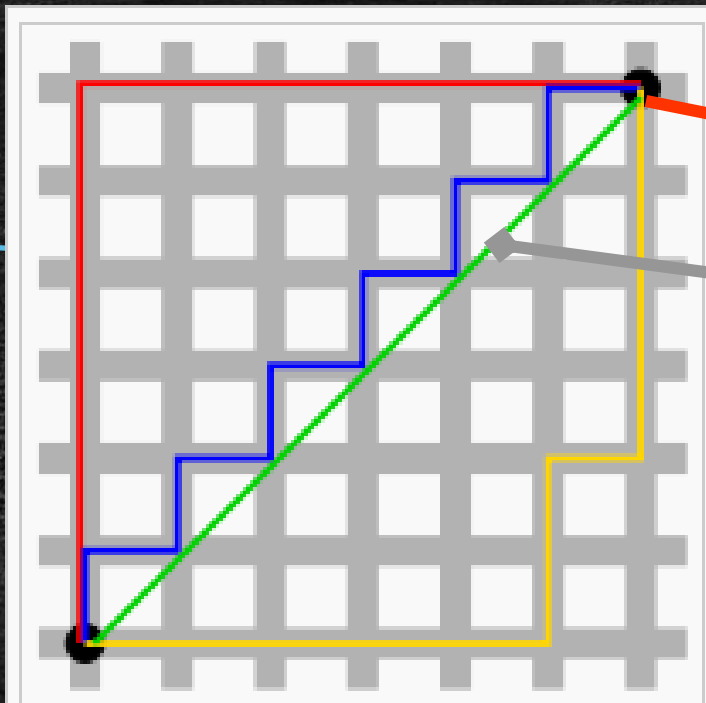
$$d(p, q) = \sqrt{(p - q)^2}.$$

- Two dimensions

$$d(p, q) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2}.$$

- Higher dimensions

$$d(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + (p_3 - q_3)^2}.$$




Manhattan distance versus Euclidean distance: The red, blue, and yellow lines have the same length (12) in both Euclidean and taxicab geometry. In Euclidean geometry, the green line has length $6 \times \sqrt{2} \approx 8.48$, and is the unique shortest path. In taxicab geometry, the green line's length is still 12, making it no shorter than any other path shown.

→ Taxicab Geometry (Manhattan distance)

→ (Euclidean distance)

Taxicab geometry, considered by Hermann Minkowski in the 19th century, is a form of geometry in which the usual metric of Euclidean geometry is replaced by a new metric in which the distance between two points is the sum of the (absolute) differences of their coordinates. The taxicab metric is also known as **rectilinear distance**, **L1 distance** or **L1 norm** (see L_p space), **city block distance**, **Manhattan distance**, or **Manhattan length**, with corresponding variations in the name of the geometry.[1] The last name alludes to the grid layout of most streets on the island of Manhattan, which causes the shortest path a car could take between two points in the city to have length equal to the points' distance in taxicab geometry.

	a	b	c	d	e	f	g	h	
8	5	4	3	2	2	2	2	2	8
7	5	4	3	2	1	1	1	2	7
6	5	4	3	2	1		1	2	6
5	5	4	3	2	1	1	1	2	5
4	5	4	3	2	2	2	2	2	4
3	5	4	3	3	3	3	3	3	3
2	5	4	4	4	4	4	4	4	2
1	5	5	5	5	5	5	5	5	1
	a	b	c	d	e	f	g	h	

The Chebyshev distance between two spaces on a [chess](#) board is the minimum number of moves a [king](#) requires to move between them. Above are the Chebyshev distances of each square from the square f6.

Chebyshev distance

In [mathematics](#), **Chebyshev distance** (or **Tchebychev distance**), or [L_∞ metric](#)^[1] is a [metric](#) defined on a [vector space](#) where the distance between two [vectors](#) is the greatest of their differences along any coordinate dimension.^[2] It is named after [Pafnuty Chebyshev](#). It is also known as **chessboard distance**, since in the game of [chess](#) the **minimum number of moves needed** by a [king](#) to go from one square on a chessboard to another equals the Chebyshev distance between the centers of the squares, if the squares have side length one, as represented in 2-D spatial coordinates with axes aligned to the edges of the board

Taxicab Geometry in Game

- OGRE battle / FF Tactics



ภาษา C, C++, Java, C# พื้นฐาน

- ตัวอย่าง เกิดวันอาทิตย์ รหัส xxxxx59

```
- 0200000000 //Hero(1,9)
0000000000
0000000000
0000000000
00000000003 //Monster(9,5)
0000000000
0000010000 //Tree(r,r)
0000000000
0000000000
0000000000
```

Taxicab = $\text{abs}(9-1) + \text{abs}(5-9) = 12$

Euclidean = $\text{sqrt}(\text{pow}(9-1, 2) + \text{pow}(5-9, 2)) = 22.9$

```
Chebyshev = if(abs(9-1) > abs(5-9))
              {d=abs(9-1)}
              else{d=abs(5-9)}=8
```

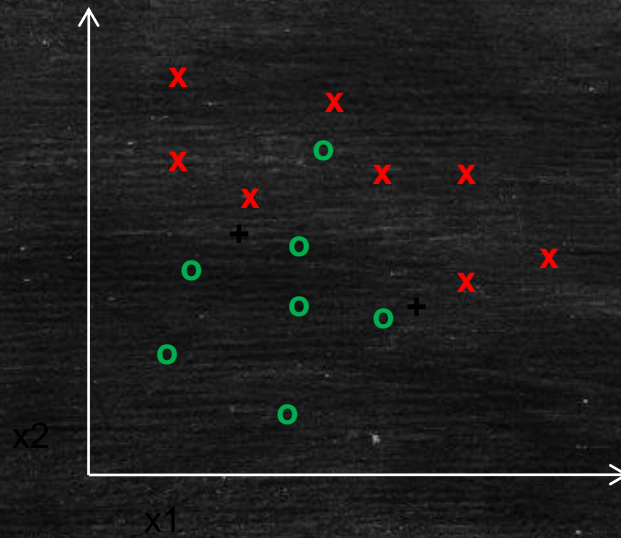
วัตถุ	X	Y
ต้นไม้	Rand(9)	Rand(9)
ฮีโร่	1	9
มอนสเตอร์	10-1=9	5

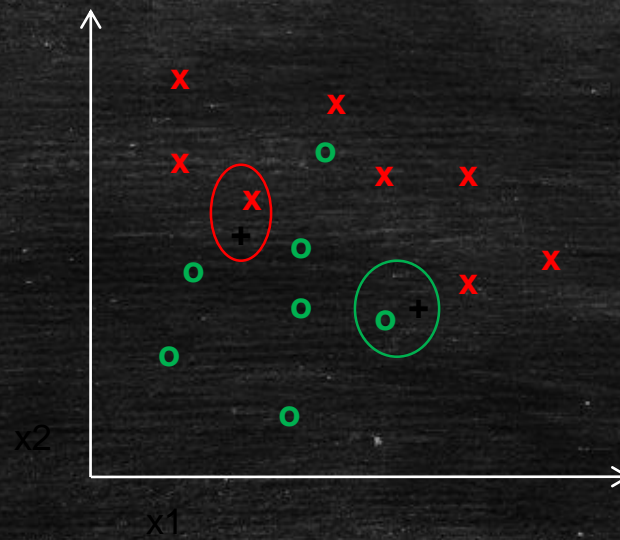
Simple recommender system with KNN

No	Feature1	Feature2	Feature3	...	FeatureN	Class
1	4	3	2		4	X
2	5	6	5		4	X
3	2	3	7		6	Y
4	5	5	2		7	Y
5	4	9	7		8	X
6	2	7	2		2	Y

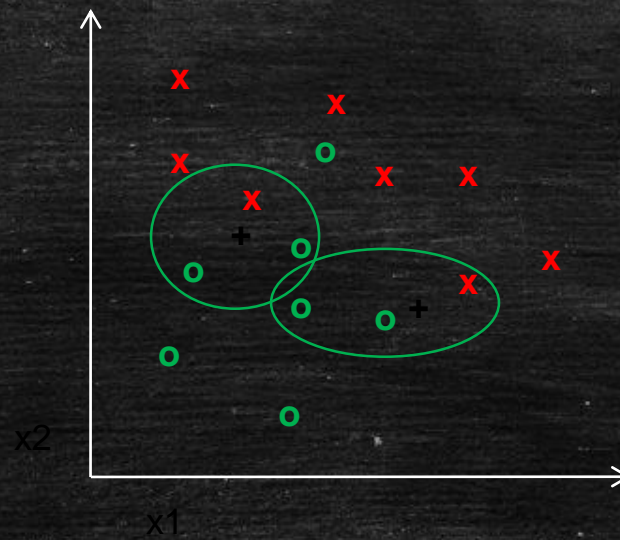
No	Feature1	Feature2	Feature3	...	FeatureN	Class
1	3	5	2		4	?
2	6	6	5		1	?
3	7	3	4		4	?

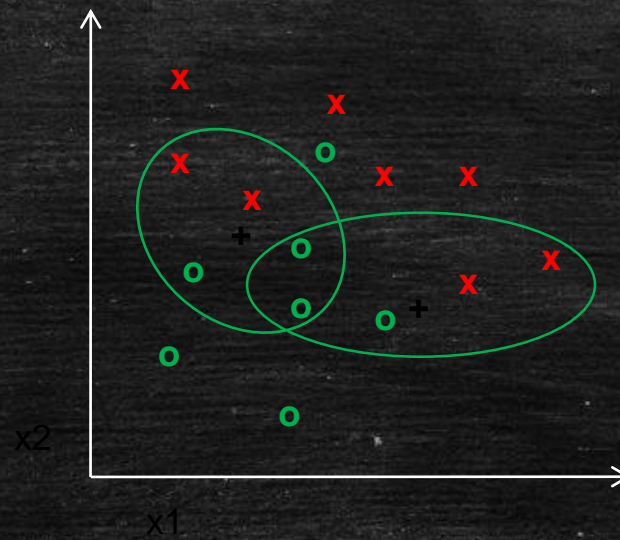
K-nearest neighbor



[illegible]

3-nearest neighbor



[illegible]

การบ้าน 5 คะแนน

- เขียนโปรแกรมหา Class จากการวัดระยะ Euclidean Distance เลือกโหนดจากตัวที่ใกล้ที่สุดสามตัว หรือจำนวนเลขคือ