



เขียนโปรแกรมภาษา C++ สำหรับผู้เริ่มต้น 2022



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>



Ol-P Maint • 1 เดือนที่ผ่านมา

ขอบคุณมากๆเลยครับ
ผมขอปูพื้นฐาน C/C++ พอจะเป็นไปได้ไหมครับ :)
ขอให้มีสื่อการสอนดีๆแบบนี้ต่อไปนะครับ

ตอบกลับ การตอบกลับ 0 รายการ 3 0 0



Joe • 2 เดือนที่ผ่านมา

อยากให้ช่วยสอน C++ มากเลยครับ แต่ถ้ามารบกวนเกินไปก็ขออภัยด้วยครับ สู้ๆสำหรับคลิปต่อไปในอนาคตนะครับ
ช่องที่มีคุณภาพมากๆเลยครับ

ตอบกลับ การตอบกลับ 0 รายการ 1 0 0



Finjez • 2 เดือนที่ผ่านมา

รอที่สอน C++ อยู่ครับ เพราะผมเรียนจากหนังสือด้วยตัวเอง ก็เลยไม่รู้ว่าจะพลาดอะไรเล็กๆน้อยๆที่สำคัญๆไปบ้าง

ตอบกลับ การตอบกลับ 0 รายการ 2 0 0



Sherox Q • 6 เดือนที่ผ่านมา

ขอบคุณมากค่ะ กราบงามๆเลย เพิ่งมาดูคลิปนี้คลิปแรก เดียวจะไปย้อนดูการเขียนภาษา C++ เบื้องต้นของพี่ด้วย

ตอบกลับ การตอบกลับ 0 รายการ 8 0 0



ประพันธ์ บรรณราช • 9 เดือนที่ผ่านมา

สอนภาษา C++ น้อยที่ บ้างเจอนไข พอไปRun คิดแก้แล้วแก้ก็ยังไม่ผ่าน

ตอบกลับ การตอบกลับ 0 รายการ 0 0 0



555 666 • 1 ปีที่แล้ว

C++ แบบ full คอร์ส ครับ เหมือน py



Yes Omain • 3 เดือนที่ผ่านมา

ขอ Unreal Engine C++ นะครับ Blueprint มั่นขำ~

ตอบกลับ การตอบกลับ 0 รายการ 2 0 0



king pro • 4 เดือนที่ผ่านมา

อยากให้สอนภาษา C++

ตอบกลับ การตอบกลับ 0 รายการ 0 0 0



KongRuksiam Official • 3 เดือนที่ผ่านมา

ตอนนี้ทางช่องมีแผนจะจัดทำเนื้อหา
การสร้างเกมด้วย Unreal Engine (Full Course)

ในช่วงต้นปีหน้า เลยอยากสอบถามความคิดเห็น Subscriber ทุกท่านว่ามีความสนใจหัวข้อดังกล่าวมากน้อยเพียงใด
สามารถโหลดหรือคอมเม้นรายละเอียดได้โพสต์นี้เลยครับ (ส่วนเนื้อหาที่จะจัดทำในปีนี้ทางช่องได้วางแผนทำไว้แล้ว ส่วนใหญ่จะเกี่ยวกับเว็บทั้งหมดครับ)
แสดงน้อยลง

838 คะแนน



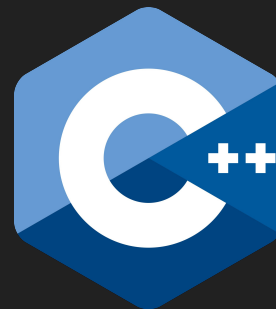
<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

เหมาะสำหรับ

- ผู้ที่สนใจเรียนรู้การเขียนโปรแกรมภาษา C++ ด้วยตนเอง
- บุพื้นฐานสำหรับใช้งานในโปรแกรม Unreal Engine
- ไม่มีความรู้เรื่องการเขียนโปรแกรมก็เรียนได้
- เนื้อหาทั้งหมดเรียนฟรี!!



1 ขอบเขตของเนื้อหา

- โครงสร้างและไวยากรณ์ภาษารวมถึงทฤษฎีการเขียนโปรแกรมด้วยภาษา C++
- เรียนรู้การเขียนโปรแกรมเพื่อแก้โจทย์ปัญหาในงานจริงได้

2 เครื่องมือที่ใช้

3 Visual Studio Community 2022



4องค์ประกอบพื้นฐานของภาษา C++

- ไฟล์ที่เก็บโค้ดภาษา C++ จะมีนามสกุลไฟล์ .cpp
- ฟังก์ชัน `main()` คือ ฟังก์ชันพิเศษ กลุ่มคำสั่งที่อยู่ในฟังก์ชันนี้ จะถูกสั่งให้ทำงานโดยอัตโนมัติเป็นลำดับแรกเสมอ

องค์ประกอบพื้นฐานของภาษา C++

- **ขอบเขต (Block)** ใช้สัญลักษณ์ {} เพื่อบอกขอบเขตการทำงานของ
ของกลุ่มคำสั่งว่ามีจุดเริ่มต้นและสิ้นสุดที่ตำแหน่งใด
- **เครื่องหมายสิ้นสุดคำสั่ง** ใช้สัญลักษณ์ ;
- **คำอธิบาย (Comment)** ใช้สัญลักษณ์ // หรือ /* */

โครงสร้างคำสั่งภาษา C++

คำสั่งของ C++ ที่เขียนในไฟล์นามสกุล .cpp จะถูกเรียก
มาจากส่วนที่เรียกว่าไลบรารี (Library)



โครงสร้างคำสั่งภาษา C++

ไลบรารี (Library) ประกอบด้วย 3 องค์ประกอบ ได้แก่

- **เครื่องหมาย # (Preprocessor)** คือการกำหนดให้ตัวแปลภาษา C++ ทราบว่าต้องแปลความหมายของไลบรารีอะไร
- **ไคเรกทีฟ (Directive)** คำสั่งที่ต่อจาก # คือการ include หรืออ้างอิงไฟล์ไลบรารี ที่อยู่ระหว่างเครื่องหมาย < กับ >
- **Header File** หมายถึงชื่อไลบรารีที่ต้องการอ้างอิง เช่น iostream



โครงสร้างคำสั่งภาษา C++

```
#include <iostream>

int main()
{
    std::cout << "Hello World!\n";
}
```



โครงสร้างคำสั่งภาษา C++

```
#include <iostream>
```

นำคำสั่งพื้นฐานที่อยู่ในไลบรารี `iostream` เข้ามาทำงาน เช่น
กลุ่มคำสั่งที่ต้องการแสดงผลออกทางจอภาพ เป็นต้น



โครงสร้างคำสั่งภาษา C++

```
#include <iostream>

int main()
{
    std::cout << "Hello World!\n";
}
```



โครงสร้างคำสั่งภาษา C++

```
#include <iostream>

int main()
{
    //แสดงผลข้อความ Hello World ออกทางจอภาพ
    std::cout << "Hello World!\n";
}
```



โครงสร้างคำสั่งภาษา C++

```
#include <iostream>
```

```
int main()
```

```
{
```

std คือชื่อ namespace

```
std::cout << "Hello World!\n";
```

```
}
```

cout คือ ฟังก์ชันสำหรับแสดงผลข้อความ



5หมายเหตุ (Comment)

จุดประสงค์

- อธิบายหน้าที่หรือความหมายของโค้ดที่เขียน
- ยกเลิกโค้ดชั่วคราว ส่งผลให้ตัวแปลภาษาไม่สนใจโค้ดในบรรทัดที่ถูกทำหมายเหตุ



หมายเหตุ (Comment)

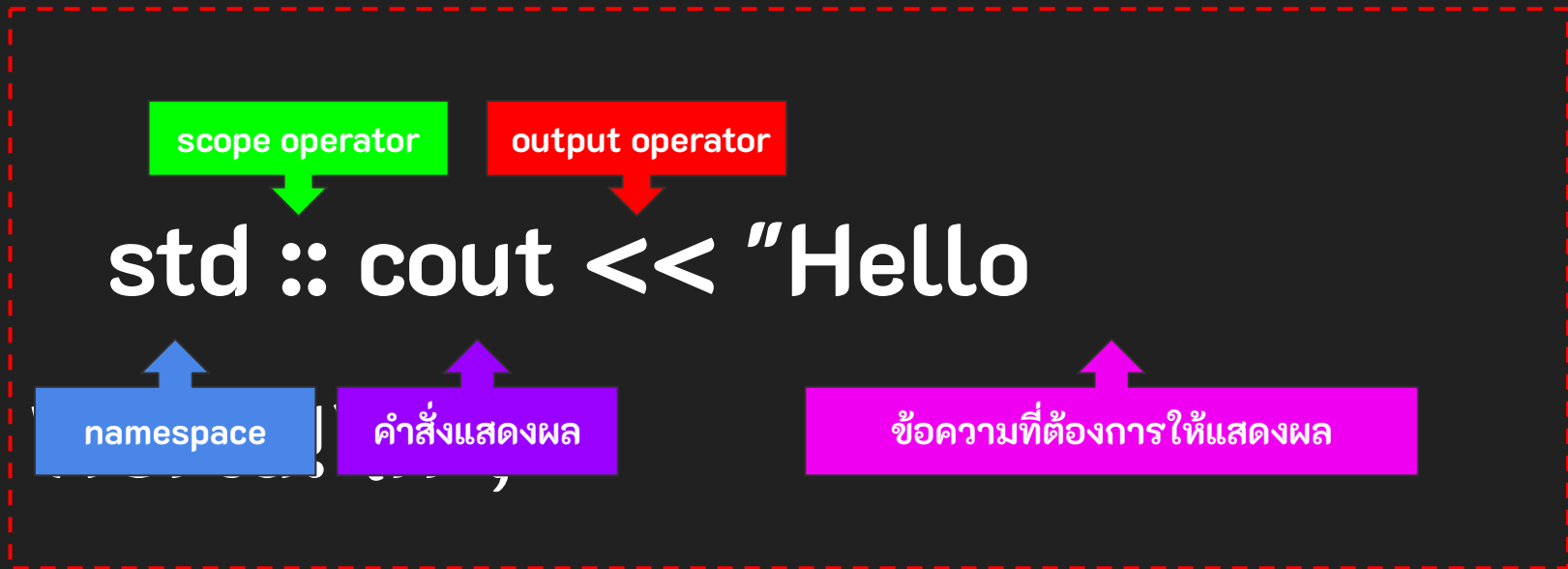
วิธีที่ 1 โดยใช้เครื่องหมาย Slash (/) ใช้ในการอธิบายคำสั่งสั้นๆ
ในรูปแบบบรรทัดเดียว

วิธีที่ 2 เขียนคำอธิบายไว้ในเครื่องหมาย /* ... */ ใช้ในการอธิบาย
คำสั่งยาวๆหรือแบบหลายบรรทัด

6.ตัวดำเนินการส่งออก (<< Output Operator)

```
std::cout << "Hello World!\n";
```

ตัวดำเนินการส่งออก (<< Output Operator)



ตัวดำเนินการส่งออก (<< Output Operator)

- `cout<< "kongruksiam";` แสดงผล
- `cout<<"kongruksiam"<<"studio";` แสดงผลและนำค่ามาต่อกัน
- `cout<<"kongruksiam"<<"studio\n";` แสดงผลและขึ้นบรรทัดใหม่
- `cout<<"kongruksiam"<<"studio";`แสดงผล
- `cout << endl;`ขึ้นบรรทัดใหม่
- `cout<<"kongruksiam"<<endl;` แสดงผลและขึ้นบรรทัดใหม่

7 ตัวแปรและชนิดข้อมูล (Variable)

ตัวแปร (variable) คือ ชื่อที่ถูกระบุขึ้นมาเพื่อใช้เก็บค่าข้อมูล
สำหรับนำไปใช้งานในโปรแกรม โดยข้อมูลประกอบด้วย ข้อความ
ตัวเลข ตัวอักษร หรือผลลัพธ์จากการประมวลผลข้อมูลค่าที่เก็บใน
ตัวแปร สามารถเปลี่ยนแปลงค่าได้



ชนิดข้อมูลพื้นฐาน (Data Type)

ชนิดข้อมูล	คำอธิบาย
boolean	ค่าทางตรรกศาสตร์ (True = 1 / False = 0)
int	ตัวเลขที่ไม่มีจุดทศนิยม
long	ตัวเลขที่ไม่มีจุดทศนิยมมีขอบเขตมากกว่า int
float	ตัวเลขที่มีจุดทศนิยม
double	ตัวเลขที่มีจุดทศนิยมมีขอบเขตมากกว่า float
char	ตัวอักษร
string	ชุดข้อความ

8 การนิยามตัวแปร

- ชนิดข้อมูล ชื่อตัวแปร ;
- ชนิดข้อมูล ชื่อตัวแปร = ค่าเริ่มต้น;
- ชนิดข้อมูล ชื่อตัวแปร1 , ชื่อตัวแปร2 ;

ให้นำค่าทางขวามือของเครื่องหมาย = ไปเก็บไว้ในตัวแปรที่อยู่ด้านซ้ายมือ



9 ค่าคงที่ (Constant)

มีลักษณะการใช้งานคล้ายกับตัวแปร แต่ค่าคงที่คือค่าจะไม่สามารถเปลี่ยนแปลงได้ ตอนประกาศใช้งานค่าคงที่ต้องมีการประกาศค่าเริ่มต้นเสมอ

การนิยามค่าคงที่ (Constant)

const ชนิดข้อมูล ชื่อตัวแปร = ค่าเริ่มต้น;

ตัวอย่าง

```
const float PI = 3.14 ;
```

```
const int SIZE = 10;
```

การนิยามค่าคงที่มักจะนิยามเป็นตัวพิมพ์ใหญ่

Break

10 กฎการตั้งชื่อตัวแปรและค่าคงที่

- ขึ้นต้นด้วยตัวอักษร A-Z หรือ a-z หรือ _ เครื่องหมายขีดเส้นใต้เท่านั้น
- อักษรตัวแรกห้ามเป็นตัวเลข
- ตัวพิมพ์เล็ก-พิมพ์ใหญ่มีความหมายต่างกัน (Case Sensitive)
- ห้ามใช้อักขระพิเศษมาประกอบเป็นชื่อตัวแปร เช่น { } , % ^ และช่องว่าง เป็นต้น
- ไม่ซ้ำกับคำสงวนในภาษา C++



คำสงวนในภาษา C++ (Keywords)

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

ตัวดำเนินการ (Operator)

กลุ่มของเครื่องหมายหรือสัญลักษณ์ที่ใช้ในการเขียนโปรแกรม

$$A+B$$

1. ตัวดำเนินการ (Operator)
2. ตัวถูกดำเนินการ (Operand)



11 ตัวดำเนินการทางคณิตศาสตร์

Operator	คำอธิบาย
+	บวก
-	ลบ
*	คูณ
/	หาร
%	หารเอาเศษ

Break

12 ตัวดำเนินการเพิ่มค่าและลดค่า

Operator	รูปแบบการเขียน	ความหมาย
++ (Prefix)	++a	เพิ่มค่าให้ a ก่อน 1 ค่าแล้วนำไปใช้
++ (Postfix)	a++	นำค่าปัจจุบันใน a ไปใช้ก่อนแล้วค่อยเพิ่มค่า
-- (Prefix)	--b	ลดค่าให้ b ก่อน 1 ค่าแล้วนำไปใช้
-- (Postfix)	b--	นำค่าปัจจุบันใน b ไปใช้ก่อนแล้วค่อยลดค่า



13 Compound Assignment

Assignment	รูปแบบการเขียน	ความหมาย
<code>+=</code>	<code>x+=y</code>	<code>x=x+y</code>
<code>-=</code>	<code>x-=y</code>	<code>x=x-y</code>
<code>*=</code>	<code>x*=y</code>	<code>x=x*y</code>
<code>/=</code>	<code>x/=y</code>	<code>x=x/y</code>
<code>%=</code>	<code>x%=y</code>	<code>x=x%y</code>



14 ตัวดำเนินการเปรียบเทียบ

ผลการเปรียบเทียบจะได้คำตอบ คือ True (1) หรือ False (0)

Operator	คำอธิบาย
==	เท่ากับ
!=	ไม่เท่ากับ
>	มากกว่า
<	น้อยกว่า
>=	มากกว่าเท่ากับ
<=	น้อยกว่าเท่ากับ



15 ลำดับความสำคัญของตัวดำเนินการ

ลำดับที่	เครื่องหมาย	ลำดับการทำงาน
1	:: , ()	
2	++ , --	ซ้ายไปขวา
3	* , / , %	ซ้ายไปขวา
4	+ , -	ซ้ายไปขวา
5	< , <= , > , >=	ซ้ายไปขวา
6	== , !=	ซ้ายไปขวา
7	&& (AND)	ซ้ายไปขวา
8	(OR)	ซ้ายไปขวา
9	= , += , -= , *= , /= , %=	ขวาไปซ้าย



ลำดับความสำคัญของตัวดำเนินการ

ลำดับที่	เครื่องหมาย	ลำดับการทำงาน
1	:: , ()	
2	++ , --	ซ้ายไปขวา
3	* , / , %	ซ้ายไปขวา
4	+ , -	ซ้ายไปขวา
5	< , <= , > , >=	ซ้ายไปขวา
6	== , !=	ซ้ายไปขวา
7	&& (AND)	ซ้ายไปขวา
8	(OR)	ซ้ายไปขวา
9	= , += , -= , *= , /= , %=	ขวาไปซ้าย



กรณีศึกษา (Case Study)

1. $2+8 \times 9 = ?$

2. $10 - 4+2 = ?$

3. $10 \times (2+5) = ?$

4. $5 \times 2 - 40 / 5 = ?$

5. $7+8/2+25 = ?$

Break

ตัวดำเนินการนำเข้า (Input Operator)

cin คือ คำสั่งสำหรับรับค่าผ่านทางคีย์บอร์ดและเก็บค่าดังกล่าวลงในตัวแปร ยกตัวอย่าง เช่น

```
int number;
```

```
cout << "Input Number = ";
```

```
cin >> number;
```

```
cout << "Output is: " << number;
```



ตัวดำเนินการนำเข้า (Input Operator)

cin คือ คำสั่งสำหรับรับค่าผ่านทางคีย์บอร์ดและเก็บค่าดังกล่าวลงในตัวแปร ยกตัวอย่าง เช่น

```
int number;
```

```
cout << "Input Number = ";
```

```
cin >> number;
```

```
cout << "Output is: " << number;
```



โครงสร้างควบคุม (Control Structure)

คือ กลุ่มคำสั่งที่ใช้ควบคุมการทำงานของโปรแกรม

- แบบลำดับ (Sequence)
- แบบมีเงื่อนไข (Condition)
- แบบทำซ้ำ (Loop)

โครงสร้างควบคุม (Control Structure)

คือ กลุ่มคำสั่งที่ใช้ควบคุมการทำงานของโปรแกรม

- แบบลำดับ (Sequence)
- แบบมีเงื่อนไข (Condition)
- แบบทำซ้ำ (Loop)

แบบมีเงื่อนไข (Condition)

กลุ่มคำสั่งที่ใช้ตัดสินใจในการเลือกทำงานตามเงื่อนไข
ต่างๆ ภายในโปรแกรม

- if
- Switch..Case

รูปแบบคำสั่งแบบเงื่อนไขเดียว

If Statement

เป็นคำสั่งที่ใช้กำหนดเงื่อนไขในการตัดสินใจทำงานของโปรแกรม
ถ้าเงื่อนไขเป็นจริงจะทำตามคำสั่งต่างๆ ที่กำหนดภายใต้เงื่อนไขนั้นๆ



รูปแบบคำสั่งแบบเงื่อนไขเดียว

```
if(condition){ //เงื่อนไข  
    คำสั่งเมื่อเงื่อนไขเป็นจริง ;  
}
```



โจทย์ปัญหา

คำนวณคะแนนสอบวิชาคอมพิวเตอร์ของนักเรียนในห้องโดยมี
คะแนนเต็ม 100 คะแนน ต้องการอยากรทราบว่านักเรียนคนใด
สอบผ่านบ้าง ใช้เกณฑ์ดังนี้ คือ

- คะแนนตั้งแต่ 50 คะแนนขึ้นไป => สอบผ่าน
- คะแนนน้อยกว่า 50 คะแนน => สอบไม่ผ่าน



If...Else Statement

```
if(เงื่อนไข){  
    คำสั่งเมื่อเงื่อนไขเป็นจริง ;  
}  
else{  
    คำสั่งเมื่อเงื่อนไขเป็นเท็จ ;  
}
```



โจทย์ปัญหา

คำนวณคะแนนสอบวิชาคอมพิวเตอร์ของนักเรียนในห้องโดยมี
คะแนนเต็ม 100 คะแนน ต้องการอยากรทราบว่านักเรียนคนใด
สอบผ่านบ้าง ใช้เกณฑ์ดังนี้ คือ

- คะแนนตั้งแต่ 50 คะแนนขึ้นไป => สอบผ่าน
- คะแนนน้อยกว่า 50 คะแนน => สอบไม่ผ่าน



if..else แบบลดรูป (Ternary Operator)

```
if(เงื่อนไข){  
    //คำสั่งเมื่อเงื่อนไขเป็นจริง  
}else{  
    //คำสั่งเมื่อเงื่อนไขเป็นเท็จ  
}
```

ตัวแปร = (เงื่อนไข) ? คำสั่งเมื่อเงื่อนไขเป็นจริง : คำสั่งเมื่อเงื่อนไขเป็นเท็จ;

รูปแบบคำสั่งแบบหลายเงื่อนไข

```
if(เงื่อนไขที่ 1){  
    คำสั่งเมื่อเงื่อนไขที่ 1 เป็นจริง ;  
}else if(เงื่อนไขที่ 2){  
    คำสั่งเมื่อเงื่อนไขที่ 2 เป็นจริง ;  
}else if(เงื่อนไขที่ 3){  
    คำสั่งเมื่อเงื่อนไขที่ 3 เป็นจริง ;  
}else{  
    คำสั่งเมื่อทุกเงื่อนไขเป็นเท็จ ;  
}
```

โจทย์ปัญหา : โปรแกรมตัดเกรดอย่างง่าย

คำนวณเกรดจากคะแนนสอบของนักเรียน โดยมีเกณฑ์ดังนี้ คือ

- คะแนนตั้งแต่ 80 ขึ้นไป ได้เกรด A
- คะแนนตั้งแต่ 70 ขึ้นไป ได้เกรด B
- คะแนนตั้งแต่ 60 ขึ้นไป ได้เกรด C
- คะแนนตั้งแต่ 50 ขึ้นไป ได้เกรด D
- น้อยกว่า 50 คะแนน ได้เกรด F



ข้อควรระวังการเขียน if เพื่อตรวจสอบเงื่อนไข

```
if(เงื่อนไข){  
    คำสั่งเมื่อเงื่อนไขเป็นจริง ;  
}if(เงื่อนไข){  
    คำสั่งเมื่อเงื่อนไขเป็นจริง ;  
}
```



Break

ตัวดำเนินการทางตรรกศาสตร์

Operator	คำอธิบาย
&& (AND)	และ
(OR)	หรือ
! (NOT)	ไม่

ตัวดำเนินการทางตรรกศาสตร์

A	!A	A	B	A && B	A B
true	false	false	false	false	false
false	true	false	true	false	true
		true	false	false	true
		true	true	true	true



โจทย์ปัญหา : เกณฑ์การรับสมัครงาน

AND (และ) , &&

ผู้สมัครเป็นเพศชาย **และ** มีอายุ 20 ปีเป็นต้นไป (ผ่านเกณฑ์)

OR (หรือ) , ||

ผู้สมัครเป็นเพศชาย **หรือ** มีอายุ 20 ปีเป็นต้นไป (ผ่านเกณฑ์)

NOT (ไม่) , !

ผู้สมัคร**ไม่ได้**เป็นเพศชาย **และ** มีอายุ 20 ปีเป็นต้นไป



ตัวดำเนินการทางตรรกศาสตร์

- ให้ $A =$ เงื่อนไขที่ 1 (เพศ)
- ให้ $B =$ เงื่อนไขที่ 2 (อายุ)

ตัวดำเนินการทางตรรกศาสตร์

A	!A	A	B	A && B	A B
true	false	false	false	false	false
false	true	false	true	false	true
		true	false	false	true
		true	true	true	true



ตัวดำเนินการทางตรรกศาสตร์

เพศ (A)	!A	A	อายุ (B)	A && B	A B
ชาย	หญิง	หญิง	<20	ไม่ผ่านเกณฑ์	ไม่ผ่านเกณฑ์
หญิง	ชาย	หญิง	>=20	ไม่ผ่านเกณฑ์	ผ่านเกณฑ์
		ชาย	<20	ไม่ผ่านเกณฑ์	ผ่านเกณฑ์
		ชาย	>=20	ผ่านเกณฑ์	ผ่านเกณฑ์



โจทย์ปัญหา : โปรแกรมตัดเกรดอย่างง่าย

คำนวณเกรดจากคะแนนสอบของนักเรียน คะแนนเต็ม 100 คะแนน โดยมีเกณฑ์ดังนี้ คือ

- คะแนน 80 - 100 ได้เกรด A
- คะแนน 70 - 79 ได้เกรด B
- คะแนน 60 - 69 ได้เกรด C
- คะแนน 50 - 59 ได้เกรด D
- คะแนน 0 - 49 คะแนน ได้เกรด F
- ป้อนค่าอื่น ได้เกรด N (Input Error)



Nested-If

```
if(เงื่อนไขหลัก){  
    if(เงื่อนไขย่อยที่ n ){  
        คำสั่งเมื่อเงื่อนไขย่อยที่ n เป็นจริง ;  
    }  
}
```



โจทย์ปัญหา : โปรแกรมคำนวณส่วนลด

ถ้าชำระเงินเกิน 15,000 บาทจะได้รับส่วนลดแต่ถ้าน้อยกว่า
จะไม่ได้รับส่วนลด

- ชำระเงิน 15,000 บาท ส่วนลด 10%
- ชำระเงิน 20,000 บาท ส่วนลด 20%
- ชำระเงิน 30,000 บาท ส่วนลด 30%



Break

แบบมีเงื่อนไข (Condition)

กลุ่มคำสั่งที่ใช้ตัดสินใจในการเลือกทำงานตามเงื่อนไข
ต่างๆ ภายในโปรแกรม

- if
- Switch..Case

แบบมีเงื่อนไข (Condition)

กลุ่มคำสั่งที่ใช้ตัดสินใจในการเลือกทำงานตามเงื่อนไข
ต่างๆ ภายในโปรแกรม

- if

- Switch..Case

Switch..Case

Switch เป็นคำสั่งที่ใช้กำหนดเงื่อนไขคล้ายๆกับ if แต่จะเลือกเพียงหนึ่งทางเลือกออกมาทำงานโดยนำค่าในตัวแปรมากำหนดเป็นทางเลือกผ่านคำสั่ง case (ตัวแปรควบคุม)



รูปแบบคำสั่ง

```
switch (ค่าที่เก็บในตัวแปรควบคุม) {
```

```
    case ค่าที่ 1 : คำสั่งที่ 1;
```

```
        break;
```

```
    case ค่าที่ 2 : คำสั่งที่ 2;
```

```
        break;
```

```
    .....
```

```
    case ค่าที่ N : คำสั่งที่ N;
```

```
        break;
```

```
    default : คำสั่งเมื่อไม่มีค่าที่ตรงกับที่ระบุใน case
```

```
}
```

*** คำสั่ง

break

จะทำให้โปรแกรมกระโดด

ออกไปทำงานนอกคำสั่ง switch

ถ้าไม่มีคำสั่ง break โปรแกรมจะทำ

คำสั่งต่อไปเรื่อยๆ จนจบการทำงาน



โจทย์ปัญหา : เลือกใช้บริการระบบธนาคาร

ให้ป้อนหมายเลขตัวเลือกเพื่อใช้บริการ

- ถ้าป้อนเลข 1 คือ เปิดบัญชีธนาคาร (Create Account)
- ถ้าป้อนเลข 2 คือ ถอน-ฝากเงิน (Withdraw and Deposit)
- ถ้าพิมพ์ตัวเลขอื่น แจ้งว่าข้อมูลไม่ถูกต้อง (Invalid)

Break

โครงสร้างควบคุม (Control Structure)

คือ กลุ่มคำสั่งที่ใช้ควบคุมการทำงานของโปรแกรม

- แบบลำดับ (Sequence)
- แบบมีเงื่อนไข (Condition)
- แบบทำซ้ำ (Loop)

โครงสร้างควบคุม (Control Structure)

คือ กลุ่มคำสั่งที่ใช้ควบคุมการทำงานของโปรแกรม

- แบบลำดับ (Sequence)
- แบบมีเงื่อนไข (Condition)
- แบบทำซ้ำ (Loop)

แบบทำซ้ำ (Loop)

กลุ่มคำสั่งที่ใช้ในการวนรอบ (loop) โปรแกรมจะทำงานไปเรื่อยๆ จนกว่าเงื่อนไขที่กำหนดไว้จะเป็นเท็จ จึงจะหยุดทำงาน

- While
- For
- Do..While

แบบทำซ้ำ (Loop)

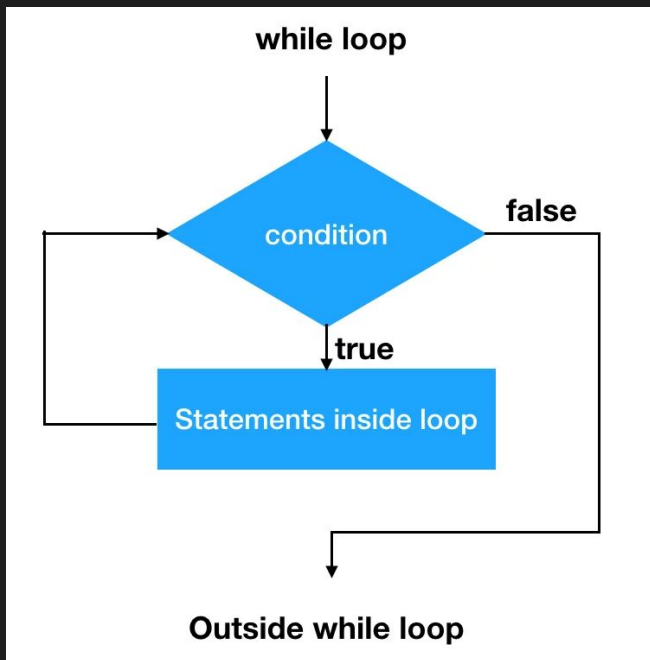
กลุ่มคำสั่งที่ใช้ในการวนรอบ (loop) โปรแกรมจะทำงานไปเรื่อยๆ จนกว่าเงื่อนไขที่กำหนดไว้จะเป็นเท็จ จึงจะหยุดทำงาน

- While
- For
- Do..While

While Loop

จะทำงานตามคำสั่งภายใน while ไปเรื่อยๆเมื่อเงื่อนไขที่กำหนดเป็นจริง

```
while(เงื่อนไข){  
    คำสั่งที่จะทำซ้ำเมื่อเงื่อนไขเป็นจริง ;  
}
```



<https://cdn.journaldev.com/wp-content/uploads/2017/10/while-loop-java.png.webp>

1. เช็คเงื่อนไขถ้าเป็นจริงให้ทำ
คำสั่งซ้ำใน Statement
2. ถ้าเป็นเท็จให้ออกจาก Loop



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

```
while(condition){  
    //statement  
}
```

Output

แสดงข้อความ “Hello C++”
จำนวน 3 ครั้ง



```
int count = 1;  
while(count<=3){  
    cout << "Hello C++";  
    count++;  
}
```

Output



```
int count = 1;  
while(count<=3){  
    cout << "Hello C++";  
    count++;  
}
```

Output



```
int count = 1;  
while(count<=3){  
    cout << "Hello C++";  
    count++;  
}
```

Output



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

```
int count = 1;  
while(count<=3){  
    cout << "Hello C++";  
    count++;  
}
```

Output

- Hello C++



```
int count = 1;  
while(count<=3){  
    cout << "Hello C++";  
    count++;  
}
```

Output

- Hello C++


```
int count = 1;  
while(count<=3){  
    cout << "Hello C++";  
    count++;  
}
```

Output

- Hello C++ , count =2



```
int count = 1;  
while(count<=3){  
    cout << "Hello C++";  
    count++;  
}
```

Output

- Hello C++ , count =2



```
int count = 1;  
while(count<=3){  
    cout << "Hello C++";  
    count++;  
}
```

Output

- Hello C++ , count =2
- Hello C++



```
int count = 1;  
while(count<=3){  
    cout << "Hello C++";  
    count++;  
}
```

Output

- Hello C++ , count = 2
- Hello C++ , count = 3



```
int count = 1;  
while(count<=3){  
    cout << "Hello C++";  
    count++;  
}
```

Output

- Hello C++ , count = 2
- Hello C++ , count = 3



```
int count = 1;  
while(count<=3){  
    cout << "Hello C++";  
    count++;  
}
```

Output

- Hello C++ , count = 2
- Hello C++ , count = 3
- Hello C++



```
int count = 1;
while(count<=3){
    cout << "Hello C++";
    count++;
}
```

Output

- Hello C++ , count = 2
- Hello C++ , count = 3
- Hello C++ , count = 4



```
int count = 1;  
while(count<=3){  
    cout << "Hello C++";  
    count++;  
}
```

Output

- Hello C++ , count = 2
- Hello C++ , count = 3
- Hello C++ , count = 4




```
int count = 1;  
while(count<=3){  
    cout << "Hello C++";  
    count++;  
}
```

Output

- Hello C++ , count = 2
- Hello C++ , count = 3
- Hello C++ , count = 4



```
int count = 1;  
while(count<=3){  
    cout << "Hello C++";  
    count++;  
}  
// จบโปรแกรม
```

Output

- Hello C++ , count = 2
- Hello C++ , count = 3
- Hello C++ , count = 4

แบบทำซ้ำ (Loop)

กลุ่มคำสั่งที่ใช้ในการวนรอบ (loop) โปรแกรมจะทำงานไปเรื่อยๆ จนกว่าเงื่อนไขที่กำหนดไว้จะเป็นเท็จ จึงจะหยุดทำงาน

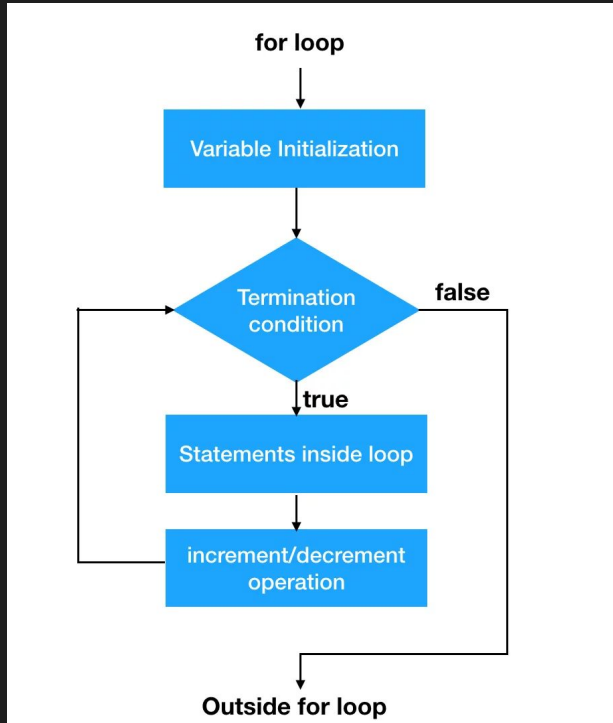
- While
- For
- Do..While

For Loop

เป็นรูปแบบการซ้ำที่ใช้ในการตรวจสอบเงื่อนไขการทำงาน มีการกำหนดค่าเริ่มต้นและเปลี่ยนค่าไปพร้อมๆกัน เมื่อเงื่อนไขในคำสั่ง for เป็นจริงก็จะทำงานตามคำสั่งที่แสดงไว้ภายในคำสั่ง for ไปเรื่อยๆ

```
for(ค่าเริ่มต้นของตัวแปร; เงื่อนไข; เปลี่ยนแปลงค่าตัวแปร) {  
    คำสั่งเมื่อเงื่อนไขเป็นจริง;  
}
```





<https://cdn.journaldev.com/wp-content/uploads/2017/10/java-for-loop.png.webp>

1. กำหนดค่าเริ่มต้น
2. เช็คเงื่อนไขถ้าเป็นจริงให้ทำคำสั่งซ้ำใน Statement
3. ถ้าเป็นเท็จให้ออกจาก Loop



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

```
for(ค่าเริ่มต้นของตัวแปร; เงื่อนไข; เปลี่ยนแปลงค่าตัวแปร){  
    cout << "Hello C++";  
}
```

Output

แสดงข้อความ "Hello C++" จำนวน 3 ครั้ง



```
for(int count = 1; count <=3;count++){  
    cout << "Hello C++";  
}
```

Output

แสดงข้อความ "Hello C++" จำนวน 3 ครั้ง



```
for(int count = 1; count <=3;count++){  
    cout << "Hello C++";  
}
```




```
for(int count = 1; count <=3;count++){  
    cout << "Hello C++";  
}
```



```
for(int count = 1; count <=3;count++){  
    cout << "Hello C++";  
}
```



```
for(int count = 1; count <=3;count++){  
    cout << "Hello C++";  
}
```

- Hello C++



```
for(int count = 1; count <=3;count++){  
    cout << "Hello C++";  
}
```

- Hello C++



```
for(int count = 1; count <=3;count++){  
    cout << "Hello C++";  
}
```

- Hello C++ , count = 2



```
for(int count = 1; count <=3;count++){  
    cout << "Hello C++";  
}
```

- Hello C++ , count = 2



```
for(int count = 1; count <=3;count++){  
    cout << "Hello C++";  
}
```

- Hello C++ , count = 2



```
for(int count = 1; count <=3;count++){  
    cout << "Hello C++";  
}
```

- Hello C++ , count = 2
- Hello C++


```
for(int count = 1; count <=3;count++){  
    cout << "Hello C++";  
}
```

- Hello C++ , count = 2
- Hello C++



```
for(int count = 1; count <=3;count++){  
    cout << "Hello C++";  
}
```

- Hello C++ , count = 2
- Hello C++ , count = 3



```
for(int count = 1; count <=3;count++){  
    cout << "Hello C++";  
}
```

- Hello C++ , count = 2
- Hello C++ , count = 3



```
for(int count = 1; count <=3;count++){  
    cout << "Hello C++";  
}
```

- Hello C++ , count = 2
- Hello C++ , count = 3



```
for(int count = 1; count <=3;count++){  
    cout << "Hello C++";  
}
```

- Hello C++ , count = 2
- Hello C++ , count = 3
- Hello C++



```
for(int count = 1; count <=3;count++){  
    cout << "Hello C++";  
}
```

- Hello C++ , count = 2
- Hello C++ , count = 3
- Hello C++



```
for(int count = 1; count <=3;count++){  
    cout << "Hello C++";  
}
```

- Hello C++ , count = 2
- Hello C++ , count = 3
- Hello C++ , count = 4



```
for(int count = 1; count <=3;count++){  
    cout << "Hello C++";  
}
```

- Hello C++ , count = 2
- Hello C++ , count = 3
- Hello C++ , count = 4


```
for(int count = 1; count <=3;count++){  
    cout << "Hello C++";  
}
```

- Hello C++ , count = 2
- Hello C++ , count = 3
- Hello C++ , count = 4

```
for(int count = 1; count <=3;count++){  
    cout << "Hello C++";  
}
```

- Hello C++ , count = 2
- Hello C++ , count = 3
- Hello C++ , count = 4



Break

แบบทำซ้ำ (Loop)

กลุ่มคำสั่งที่ใช้ในการวนรอบ (loop) โปรแกรมจะทำงานไปเรื่อยๆ จนกว่าเงื่อนไขที่กำหนดไว้จะเป็นเท็จ จึงจะหยุดทำงาน

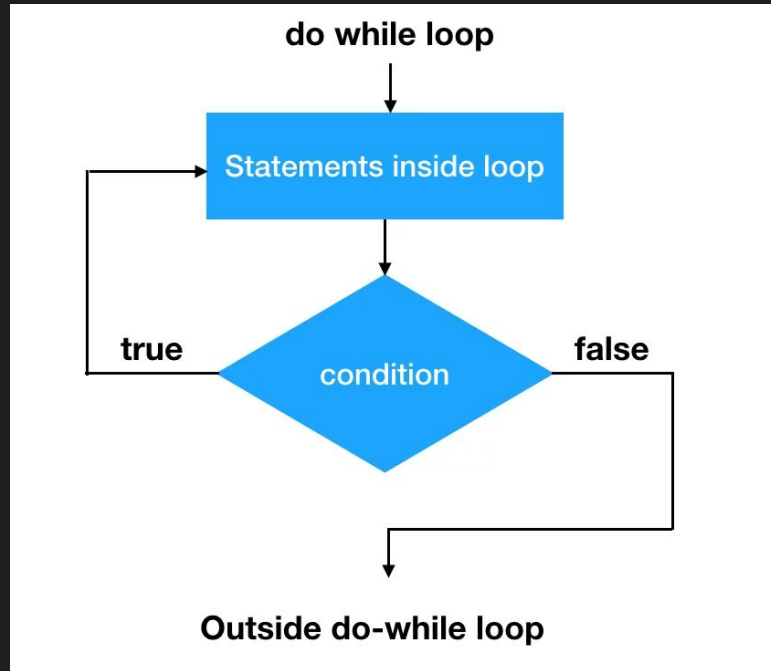
- While
- For
- Do..While

Do...While Loop

โปรแกรมจะทำงานตามคำสั่งอย่างน้อย 1 รอบ เมื่อทำงานเสร็จจะ
มาตรวจสอบเงื่อนไขที่คำสั่ง while ถ้าเงื่อนไขเป็นจริงจะวนกลับขึ้นไปทำ
งานที่คำสั่งใหม่อีกรอบ แต่ถ้าเป็นเท็จจะหลุดออกจากลูป

```
do {  
    คำสั่งต่างๆ เมื่อเงื่อนไขเป็นจริง;  
} while(เงื่อนไข);
```





<https://cdn.journaldev.com/wp-content/uploads/2017/10/java-do-while-loop-1.png.webp>

1. ทำงานคำสั่งใน Statement
2. เช็คเงื่อนไขถ้าเป็นจริงให้
กลับไปทำซ้ำใน
Statement
3. ถ้าเป็นเท็จให้ออกจาก Loop



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

คำสั่งที่เกี่ยวข้องกับ Loop

- **break** ถ้าโปรแกรมพบคำสั่งนี้จะหลุดจากการทำงานในลูปทันที เพื่อไปทำคำสั่งอื่นที่อยู่นอกลูป
- **continue** คำสั่งนี้จะทำให้หยุดการทำงานแล้วย้อนกลับไปเริ่มต้นการทำงานที่ต้นลูปใหม่



ข้อแตกต่างและการใช้งาน Loop

- For ใช้ในกรณีรู้จำนวนรอบที่ชัดเจน
- While ใช้ในกรณีที่ไม่รู้จำนวนรอบ
- Do..while ใช้ในกรณีที่ต้องการให้ลองทำก่อน 1 รอบ
แล้วทำซ้ำไปเรื่อยๆ ตราบเท่าที่เงื่อนไขเป็นจริง



Break

โจทย์ปัญหา : หาผลรวมตัวเลข

- จำนวนตัวเลข : 5 จำนวน
- รับค่าตัวเลขผ่านแป้นพิมพ์
- แสดงผลรวมตัวเลข 5 จำนวน (summation)

โจทย์ปัญหา : หาผลรวมตัวเลข (ไม่จำกัดจำนวน)

- จำนวนตัวเลข : ไม่จำกัด
- รับค่าตัวเลขผ่านแป้นพิมพ์
- ถ้าป้อนเลขติดลบให้จบการทำงาน
- แสดงผลรวมตัวเลขตามจำนวนที่ป้อน (summation)

เข้าเรียนเนื้อหา



เขียนโปรแกรมภาษา C++ สำหรับผู้เริ่มต้น [2022]

วิดีโอ 31 รายการ • การดู 948 ครั้ง • อัปเดตแล้ววันนี้



KongRuksiam
Official

ติดตามแล้ว



1

สอน C++ เบื้องต้น [2022] ตอนที่ 1 - แนะนำเนื้อหา

KongRuksiam Official



2

สอน C++ เบื้องต้น [2022] ตอนที่ 2 - ดาวน์โหลด Visual Studio

KongRuksiam Official



3

สอน C++ เบื้องต้น [2022] ตอนที่ 3 - สร้างโปรเจกต์ใหม่

KongRuksiam Official



4

สอน C++ เบื้องต้น [2022] ตอนที่ 4 - โครงสร้างภาษา C++

KongRuksiam Official



5

สอน C++ เบื้องต้น [2022] ตอนที่ 5 - หมายเหตุ (Comment)

KongRuksiam Official



สแกนเพื่อเข้าเรียน



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>