

+ - * / %
- -- ++ < <= >
>= == != && |
= += -= *= /=
%=

Computer Programming For Information System

C++

214106

02

OPERATORS

Operators

- Review problem
- What is an operator?
- Arithmetic operators
- Unary operators
- Operator precedence groups
- Relational and logical operators
- Assignment operators
- Cast operators
- Sizeof operator
- Problem for This Week

Operators

Review problem: 1

<ภาพเคลื่อนไหว 1.1: การอนุมัติบัตรเครดิต>

■ หน้าหลัก ■ ภาพเคลื่อนไหว ■

[กองทุนนวัตกรรมและสิ่งประดิษฐ์ สมเด็จพระเทพรัตนราชสุดาฯ สยามบรมราชกุมารี]

บทที่ 1 --- ผังงาน

1.1 การอนุมัติบัตรเครดิต

ธนาคารแห่งหนึ่งต้องการออกบัตรเครดิตในวงเงิน 100,000 บ. ให้กับลูกค้าของธนาคารคนหนึ่ง แต่มีเงื่อนไขดังนี้คือ

ธนาคารจะออกบัตรเครดิตให้ถ้าลูกค้า

1. มีทรัพย์สินไม่ต่ำกว่า 500,000 บ. หรือ
2. มีรายได้ไม่ต่ำกว่าเดือนละ 50,000 บ.

จึงวิเคราะห์ปัญหาและเขียนผังงานในการสรุปผลที่ทางธนาคารจะอนุมัติหรือไม่อนุมัติในการออกบัตรเครดิตให้กับลูกค้ารายนี้

แนวทางวิเคราะห์ปัญหา

กระดานข้อความ

กดปุ่ม

"แนวทางวิเคราะห์ปัญหา"

ทางด้านซ้ายมือ

เพื่อเริ่มการวิเคราะห์ปัญหา

โปรแกรมภาพเคลื่อนไหวสำหรับการเรียนการสอนวิชาการโปรแกรมคอมพิวเตอร์ภาษาซี

เบเกอรี่


▶ คณะ ชาญศิลป์ - สาขาวิชาวิศวกรรมคอมพิวเตอร์ - สำนักวิชาวิศวกรรมศาสตร์ - มหาวิทยาลัยเทคโนโลยีสุรนารี 2548 ◀

Operators

Review problem: 2

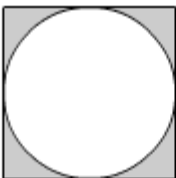
<ภาพเคลื่อนไหว 1.2: วงกลมในสี่เหลี่ยม>

■ หน้าหลัก ■ ภาพเคลื่อนไหว ■ [กองทุนนวัตกรรมและสิ่งประดิษฐ์ สมเด็จพระเทพรัตนราชสุดาฯ สยามบรมราชกุมารี]

บทที่ 1 --- ผังงาน 

1.2 วงกลมในสี่เหลี่ยม

จงเขียนผังงานในการคำนวณหาพื้นที่ของรูปสี่เหลี่ยมในส่วนที่แรเงาดังในรูป "Circle in The Square" ซึ่งผู้ใช้จะป้อนรัศมีของวงกลมเข้ามา โดยแสดงผลลัพธ์ออกที่จอภาพ



ตัวอย่างการรันโปรแกรม
Please enter radius : 3
The grey area: 7.725691

แนวทางวิเคราะห์ปัญหา

กระดานข้อความ

กดปุ่ม
"แนวทางวิเคราะห์ปัญหา"
ทางด้านซ้ายมือ
เพื่อเริ่มการวิเคราะห์ปัญหา

โปรแกรมภาพเคลื่อนไหวสำหรับการเรียนการสอนวิชาการโปรแกรมคอมพิวเตอร์ภาษาซี

▶ คณา ชาญศิลป์ - สาขาวิชาวิศวกรรมคอมพิวเตอร์ - สำนักวิชาวิศวกรรมศาสตร์ - มหาวิทยาลัยเทคโนโลยีสุรนารี 2548 ◀

Operators

Review problem: 3 <ภาพเคลื่อนไหว 2.1: การรับและแสดงข้อมูลโดยใช้ Input/Output>

[กองทุนนวัตกรรมและสิ่งประดิษฐ์ สมเด็จพระเทพรัตนราชสุดาฯ สยามบรมราชกุมารี]

บทที่ 2 --- ชนิดข้อมูล, การรับ, และการแสดงผล

หน้าหลักภาพเคลื่อนไหว

C
source code

การรับและแสดงข้อมูลโดยใช้ printf() และ scanf()

ตัวอย่าง การประมวลค่าตัวแปร เพื่อรับข้อมูลของนักศึกษา และรายละเอียดต่างๆ ดังต่อไปนี้

-- ชื่อนักศึกษา (ไม่เกิน 15 ตัวอักษร)

-- อายุ

-- เกรดเฉลี่ย

-- เกรดที่คาดว่าจะได้รับจากวิชา Com Pro. (ไม่เกิน 1 ตัวอักษร)

หลังจากนั้นให้พิมพ์ข้อมูลที่หมดออกทั้งหมด

ตัวอย่างการรันโปรแกรม

Name: Somsak

Age: 19

GPA: 3.25

Expected Grade: A

Hi... Somsak, You are 19 years old.

Your GPA is 3.250000, and you have expected A.

C
source code

กระดานข้อความ

กดปุ่ม C source code เพื่อดูซอร์สโค้ด

จอแสดงผล

หน่วยความจำ = 1 ไบต์

โปรแกรมหุ่นยนต์สำหรับการเรียนการสอนวิชาการโปรแกรมคอมพิวเตอร์ภาษาซี

คณะ ชาวนิลปี - สาขาวิศวกรรมคอมพิวเตอร์ - สำนักวิศวกรรมการผลิต - มหาวิทยาลัยเทคโนโลยีสุรนารี 2548

Operators

Review problem: 4

<ภาพเคลื่อนไหว 2.3: การหาพื้นที่ของรูปสามเหลี่ยม>

■ หน้าหลัก ■ ภาพเคลื่อนไหว ■ [กองทุนนวัตกรรมและสิ่งประดิษฐ์ สมเด็จพระเทพรัตนราชสุดาฯ สยามบรมราชกุมารี]


บทที่ 2 --- ชนิดข้อมูล, การรับ, และการแสดงผล

2.3 การหาพื้นที่ของรูปสามเหลี่ยม

โปรแกรมแสดงการคำนวณหาพื้นที่ของรูปสามเหลี่ยม โดยรับข้อมูลความยาวของฐานและความสูงจากผู้ใช้ แล้วแสดงผลลัพธ์ออกที่หน้าจอ

สูตรการหาพื้นที่ของรูปสามเหลี่ยมคือ $\frac{1}{2} * \text{ฐาน} * \text{สูง}$

ตัวอย่างการรันโปรแกรม
 Base: 7
 Height: 5
 The area is: 17.50



กระดานข้อความ


กดปุ่ม
C
source code

เพื่อดูซอร์สโค้ด

--	--	--	--

หน่วยความจำ ■ = 1 ไบต์

โปรแกรมภาพเคลื่อนไหวสำหรับการเรียนการสอนวิชาการโปรแกรมคอมพิวเตอร์ภาษาซี



▶ คณา ชานูศิลป์ - สาขาวิชาวิศวกรรมคอมพิวเตอร์ - สำนักวิชาวิศวกรรมศาสตร์ - มหาวิทยาลัยเทคโนโลยีสุรนารี
2548 ▶

Operators

What is an operator?

- An operator: a symbol used to form an expression

+	-	*	/	%	=
--	++	<	<=	>	>=
-	==	!=	&&		!
+=	-=	*=	/=	%=	

- Examples:

```
sum = a + b;  
total = total + 1;  
dif = a - b;  
area = (bas * height) / 2;  
count++;  
i = j = k = 0;  
x = x + 5;  
z += y;  
--num;  
j *= (i - 3);  
i = 'x';  
n = 'A' + 'B';
```

```
count = count+1;
```

```
z = z + y;
```

```
j = j * (i-3);
```

Operators

Arithmetic operators

ตัวดำเนินการ	ความหมาย	ตัวอย่าง	ผลลัพธ์
-	การลบ	10-5	5
+	การบวก	10+5	15
*	การคูณ	10*5	50
/	การหาร	10/5	2
%	การหารคิดเฉพาะเศษ	9%2	1

Operators

Unary operators

- An operator that needs only one operand

Operator	Purpose	Example
-	use with one operand to produce a new value	-743, -(x+y)
++	adds 1 to its operand	++n, total++
--	subtracts 1 to its operand	--x, count--

- Examples:

Prefix: alters its value before its value is used

Example: suppose `i = 1`

```
cout << "i = " << i << endl;      /* Output will be: i = 1 */
cout << "i = " << ++i << endl;     /* Output will be: i = 2 */
cout << "i = " << i;               /* Output will be: i = 2 */
```

Postfix: alters its value after its value is used

Example: suppose `i = 1`

```
cout << "i = " << i << endl;      /* Output will be: i = 1 */
cout << "i = " << i++ << endl;     /* Output will be: i = 1 */
cout << "i = " << i;               /* Output will be: i = 2 */
```

การเพิ่มหรือลดค่าทีละหนึ่ง

เครื่องหมาย	การดำเนินการ
++	เพิ่มค่าทีละหนึ่ง (Increment)
--	ลดค่าทีละหนึ่ง (Decrement)

ตัวอย่างการใช้งาน	ความหมาย
$y = ++x$	บวกค่าในตัวแปร x เพิ่มขึ้น 1 ก่อนที่จะกำหนดค่า x ให้กับตัวแปร y
$y = x++$	กำหนดค่าในตัวแปร x ให้กับตัวแปร y ก่อนที่จะบวกค่า x เพิ่มขึ้น 1
$y = --x$	ลบค่าในตัวแปร x ลง 1 ก่อนที่จะกำหนดค่า x ให้กับตัวแปร y
$y = x--$	กำหนดค่าในตัวแปร x ให้กับตัวแปร y ก่อนที่จะลดค่า x ลง 1

ตัวอย่างการนำไปใช้

ผลลัพธ์

z=6, x=5

```
#include <stdio.h>
```

```
int main(void) {
```

```
    int x=5, y=5, z;    //สร้างตัวแปรชนิดตัวเลขจำนวนเต็ม (int) ขึ้นมา 3 ตัว คือ x มีค่า 5, y มีค่า 5 และ z
```

```
        z = ++x;        //บวกค่าของ x เพิ่มขึ้น 1 ก่อนที่จะกำหนดให้ z ดังนั้น x= z=6
```

```
        x = y++;        //กำหนดค่า y ให้ x ก่อนที่จะบวกค่าของ y เพิ่มขึ้น 1 ดังนั้น x=5 ในขณะที่ y = 6
```

```
    cout << "z=" << z << ", x=" << x;    //แสดงค่าในตัวแปร Z และตัวแปร X
```

```
    return 0;
```

```
}
```

เครื่องหมายประเภทที่เรียกว่าลวดรูป

เครื่องหมาย	ตัวอย่างการใช้งาน	มีค่าเท่ากับ	ความหมาย
+=	y += x	$(y=y+x)$	บวกค่าในตัวแปร y ด้วยค่า ในตัวแปร x ผลลัพธ์ที่ได้กำหนดกลับไปให้ y
- =	y -= x	$(y=y-x)$	ลบค่าในตัวแปร y ด้วยค่า ในตัวแปร x ผลลัพธ์ที่ได้กำหนดกลับไปให้ y
*=	y *= x	$(y=y*x)$	คูณค่าในตัวแปร y ด้วยค่า ในตัวแปร x ผลลัพธ์ที่ได้กำหนดกลับไปให้ y
/=	y /= x	$(y=y/x)$	หารค่าในตัวแปร y ด้วยค่า ในตัวแปร x ผลลัพธ์ที่ได้กำหนดกลับไปให้ y
%=	y %= x	$(y=y\%x)$	หารค่าในตัวแปร y ด้วยค่า ในตัวแปร x เศษจากการหารเป็นผลลัพธ์กำหนดกลับไปให้ y

ตัวอย่างการนำไปใช้

ผลลัพธ์

a=12, b=6, c=6, d=1, e=2

```
int main() {  
    int a=7, b=10, c=1, d=3, e=8;  
    a += 5 ;           // a=7+5;  
    b -=4;             // b=10-4;  
    c *=6;             // c=1*6;  
    d /=3;             // d=3/3;  
    e %=3;             // e=8%3;  
    cout << "a=" << a << ", b=" << b << ", c=" << c << ",  
d=" << d << ", e=" << e;  
    return 0;  
}
```

Operator precedence groups

เครื่องหมายคณิตศาสตร์	ลำดับการประมวลผล	หมายเหตุ
()	1	ถ้าเครื่องหมายมีลำดับการประมวลผลระดับเดียวกันให้ดำเนินการจากซ้ายไปขวา
++, -- (ใช้แบบ prefix)	2	
- (เครื่องหมายหน้าตัวเลข)	3	
* / %	4	
+ -	5	
+= -= *= /= %=	6	

$(A + B) * C$

1
2

$A * B / (C / D)$

1
2
3

$21 \times 5 - 40 + 8 \div 2$

105 4
65
69

$6 + (9 + 1) \times 4 - 2^3$

10 8
40
46
38

Operator precedence groups

ชนิดข้อมูล	อันดับ	ตัวอย่าง
long double	<div> <div>สูงสุด</div> <div>↓</div> <div>ต่ำสุด</div> </div>	int * long = long
double		char + int = int
float		int + float = float
long		int * double = double
int		float + double = double
char		long + long double = long double

- Suppose $i = 7$ (integer),
 $f = 5.5F$ (floating-point number)
or $f = 5.5$ (double),
and $c = 'w'$ ASCII code of 'w' = 119

Expression	Value	Type
$i+f$	12.5	float or double
$i+c$	126	integer
$i+c-'y'$	5	integer
$(i+c) - (2*f/5)$	123.8	float or double
$i \% 3$	1	integer
$f \% 5$	0.5	float or double

Operators

Relational and logical operators

■ Suppose

```
int  a = 5, b = 7, num = 9;
```

```
char ch = 'a';
```

Type	Operator	Meaning	Expression	Interpretation
Relational	<	less than	num < 5	0 (False)
	<=	less than or equal to	num <= 10	1 (True)
	>	greater than	a > b	0 (False)
	>=	greater than or equal to	num >= b	1 (True)
Equality	==	equal to	num == 10	0 (False)
	!=	not equal to	ch != 'A'	1 (True)
Logical	&&	and	(a==b) && (num>b)	0 (False)
		or	(ch=='a') (num==20)	1 (True)
Logical not	!	not	!(a == 5)	0 (False)
			!(a > b)	1 (True)

Operators

■ Relational and logical operators (cont.)

■ Truth Table for && (and) ----- || (or) Operators

P	Q	P&&Q
T	T	T
T	F	F
F	T	F
F	F	F

P	Q	P Q
T	T	T
T	F	T
F	T	T
F	F	F

- Logical operator AND (&&) will be TRUE if and only if both conditions are TRUE
- Logical operator OR (||) will be FALSE if and only if both conditions are FALSE

Operators

■ Assignment operators

- Assignment operator is used to assign the value to an identifier.

- Examples:

```
a = 5;  
count = count + 1;  
count += 1;  
x *= (y + 1);  
avg = sum / n;
```

- Suppose: i and j are integer-type variables where j = 5 and the ASCII value of a character 'x' is 120

Expression	Value
i = 3.3;	Error
i = -3.9;	Error
i = j + 1;	6
i = j * j;	25
i = 2 * j / 2;	5
i = j / 2;	2
i = j % 2;	1
i = 2 * (j / 2);	4
i = 'u' + 'x';	237

Operators

Cast operators

- Form of the cast operator:

(data type) expression

- Meaning:

- converts an expression to be a specific data type

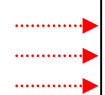
- Example:

```
int x1, x2, x3;
double y1, y2, y3;

x1 = 3 / 2;
y1 = 3 / 2;
cout << "x1 = " << x1 << ", y1 = " << y1 << endl;

x2 = 3 / 2 + 3 / 2;
y2 = 3 / 2 + (double)3 / 2;
cout << "x2 = " << x2 << ", y2 = " << y2 << endl;

x3 = 3 / 2 + 3 / 2;
y3 = (int)3.4 / 2 + 3 / 2;
cout << "x3 = " << x3 << ", y3 = " << y3 << endl;
```



```
x1 = 1, y1 = 1
x2 = 2, y2 = 2.5
x3 = 2, y3 = 2
—
```

Operators

Sizeof operator

- Form of the sizeof operator:

sizeof (operand)

- Meaning:

- returning the size of its operand in byte(s)

```
x uses memory 4 byte(s)
y uses memory 4 byte(s)
z uses memory 8 byte(s)
c uses memory 1 byte(s)
s uses memory 10 byte(s)
```

```
int x; float y; double z; char c; char s[] = "Suranaree";
```

```
cout << "x uses memory " << sizeof(int) << " byte(s)" << endl;
cout << "y uses memory " << sizeof(float) << " byte(s)" << endl;
cout << "z uses memory " << sizeof(double) << " byte(s)" << endl;
cout << "c uses memory " << sizeof(char) << " byte(s)" << endl;
cout << "s uses memory " << sizeof(s) << " byte(s)";
```

Basic Library Functions

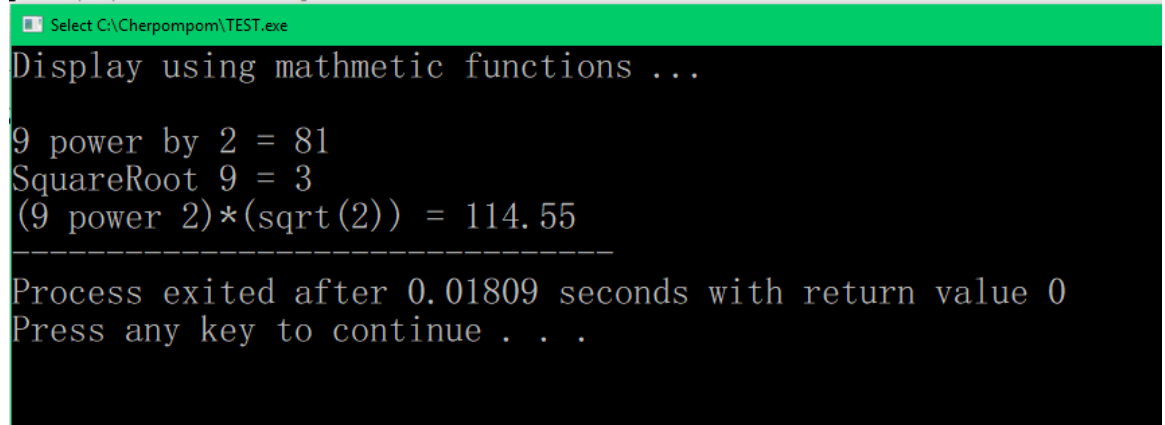
การคำนวณโดยใช้ฟังก์ชันทางคณิตศาสตร์ (mathmetic function)

```
/* LANG: C++ */
#include <iostream>
#include <math.h>
#include <conio.h>
#include <iomanip>
using namespace std;
int main()
{
```

```
    double x = 9.0, y = 2.0;
```

```
    cout << "Display using mathmetic functions ..." << endl << endl;
    cout << x << " power by " << y << " = " << pow(x, y) << endl;
    cout << "SquareRoot " << x << " = " << sqrt(x) << endl;
    cout << "(" << x << " power " << y << ")*(sqrt(" << y << ")) = "
    << fixed << setprecision(2) << pow(x,y) * (sqrt(y)) ;

    return 0;
}
```



The screenshot shows a Windows command prompt window titled "Select C:\Cherpompom\TEST.exe". The output of the program is as follows:

```
Display using mathmetic functions ...
9 power by 2 = 81
SquareRoot 9 = 3
(9 power 2)*(sqrt(2)) = 114.55
-----
Process exited after 0.01809 seconds with return value 0
Press any key to continue . . .
```

Operators

Problem for This Week

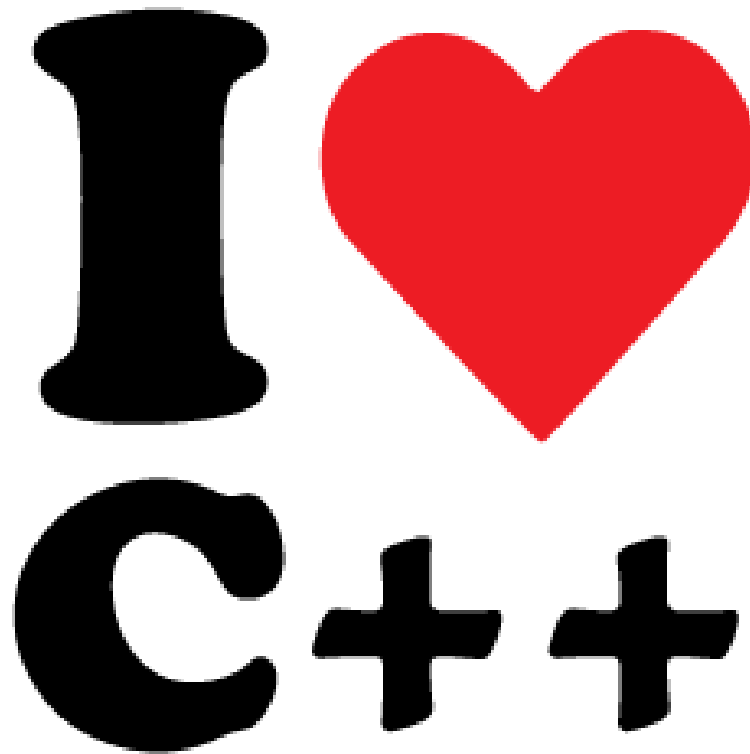
ข้อที่ 1. จงหาผลลัพธ์จากโปรแกรมต่อไปนี้

```
int main() {  
    int j=5, k=5;  
  
    cout << "j = " << j << ", k = " << k << endl;  
    cout << "j = " << j++ << ", k = " << k-- << endl;  
    cout << "j = " << j << ", k = " << k << endl;  
    cout << "j = " << ++j << ", k = " << --k;  
  
    return 0;  
}
```

ข้อที่ 2. จงเขียนโปรแกรมเพื่อรับข้อมูลจากผู้ใช้งาน ซึ่งผู้ใช้งานจะป้อนเลขจำนวนเต็ม 1 จำนวน มีค่าตั้งแต่ 1001 - 1999 แล้วพิมพ์ผลลัพธ์ดังตัวอย่างต่อไปนี้ <รูปภาพเคลื่อนไหว 3.4: การแยกเลขจำนวนเต็มออกเป็นหลักย่อย>

Enter a number (1001 - 1999) : 1234
1+2+3+4 = 10_

Enter a number (1001 - 1999) : 1924
1+9+2+4 = 16_



Week 2 : operators

Problem 1

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

    int sum=0,n=1;

    sum += n;
    sum += n++;
    sum += ++n;
    sum += n;
    n += sum;
    cout << "Sum: " << sum << endl << "N: " << n;

    return 0;
}
```


Problem 1 (cont.)



sum=0 , n=1



1.	sum += n;	---->	sum = sum + n
2.	sum += n++;		1 = 0 + 1
3.	sum += ++n;		
4.	sum += n;		
5.	n += sum;		

sum=1, n=1

Problem 1 (cont.)



1.

`sum += n;`

2.

`sum += n++;` \dashrightarrow `sum = sum + n`

3.

`sum += ++n;`

4.

`sum += n;`

5.

`n += sum;`

$$2 = 1 + 1$$

$$\underline{n = n + 1}$$

$$2 = 1 + 1$$

sum=1, n=1

sum=2, n=2

Problem 1 (cont.)



1.

sum += n;

sum=2, n=2

2.

sum += n++;

3.

sum += ++n;

n = n + 1

4.

sum += n;

3 = 2 + 1

5.

n += sum;

sum = sum + n

5 = 2 + 3

sum=5, n=3

Problem 1 (cont.)



1.

sum += n;

2.

sum += n++;

3.

sum += ++n;

4.

sum += n; -----> sum = sum + n

5.

n += sum;

sum=5, n=3

8 = 5 + 3

sum=8, n=3

Problem 1 (cont.)



1.

`sum += n;`

2.

`sum += n++;`

3.

`sum += ++n;`

4.

`sum += n;` -----> `n = n + sum`

5.

`n += sum;`

`11 = 3 + 8`

sum=8, n=3

sum=8, n=11

Problem 1 (cont.)

```
int sum=0, n=1;
```

```
1. sum += n;
```

```
2. sum += n++;
```

```
3. sum += ++n;
```

```
4. sum += n;
```

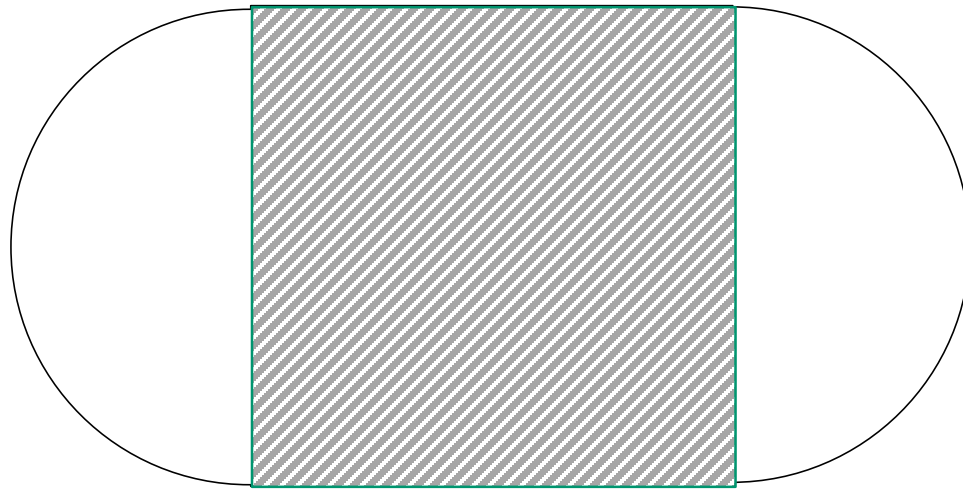
```
5. n += sum;
```

```
cout << "Sum: " << sum << endl << "N: " << n;
```



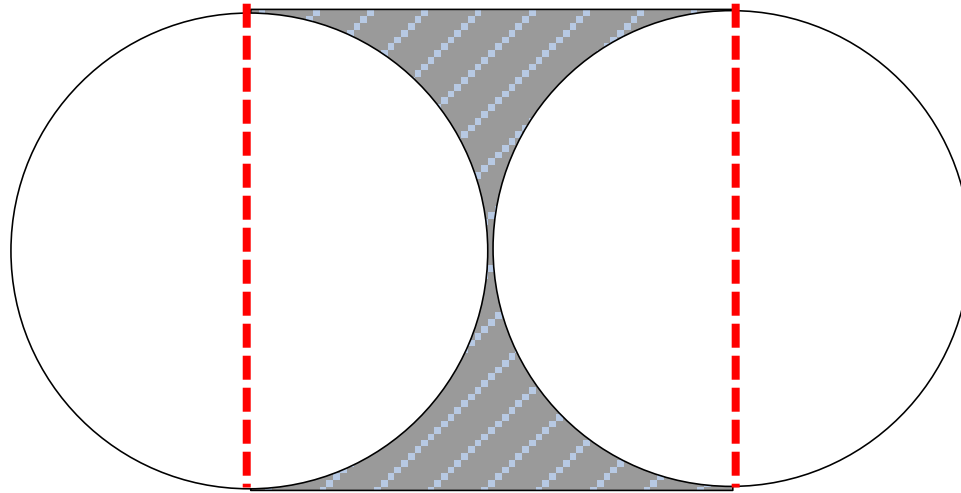
Sum: 8
N: 11

Problem 2

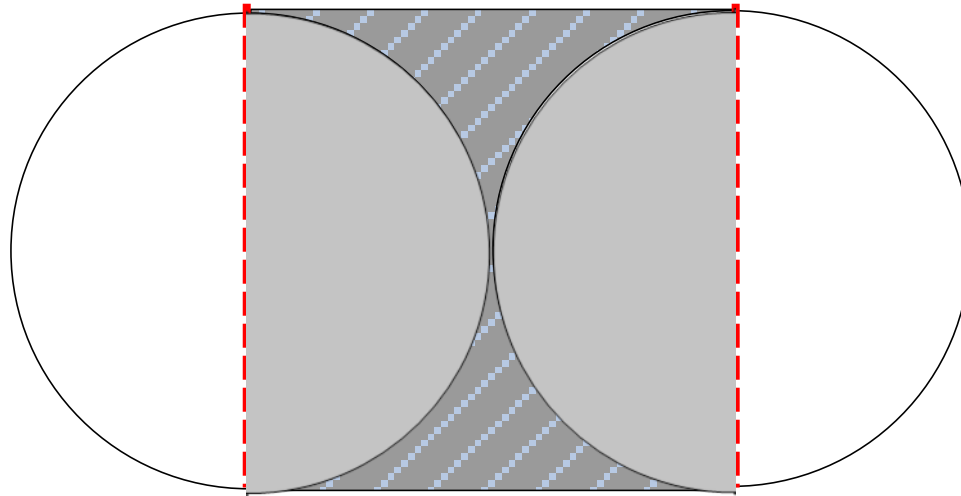


Problem 2 (cont.)

- “ความยาวด้าน” ของรูปสี่เหลี่ยม
- “เส้นผ่านศูนย์กลาง” ของรูปวงกลม



Problem 2 (cont.)



Problem 2 (cont.)

