

2110101 : การทำโปรแกรมคอมพิวเตอร์

Selection : if, if-else, switch-case

ภาควิชาวิศวกรรมคอมพิวเตอร์

จุฬาลงกรณ์มหาวิทยาลัย

# หัวข้อ

---

---

- คำสั่ง if
- คำสั่ง if-else
- Boolean expression
  - Relational operator
  - Boolean operator
  - operator precedence
- Nested if
- Statement & block
- switch-case
- การเปรียบเทียบจำนวนจริงและสตริง

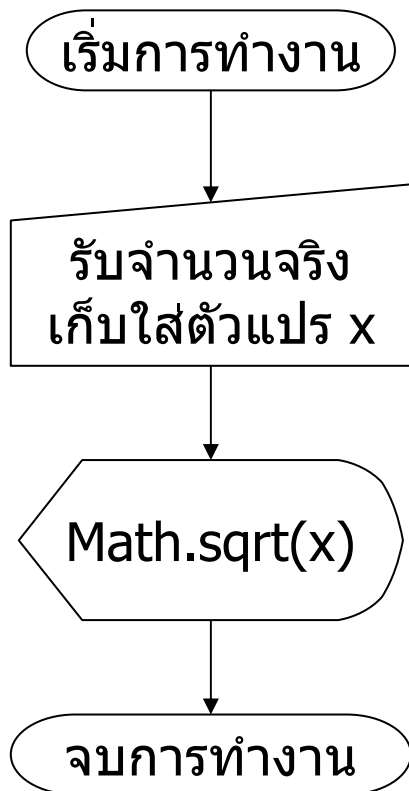
# โปรแกรมหารากที่สอง

---

---

- Requirement
  - อ่านจำนวนจากผู้ใช้ แล้วแสดงรากที่สองของจำนวนนั้น
- Analysis
  - รับจำนวนจริงผ่านทางแป้นพิมพ์
  - ใช้ `Math.sqrt` ในการหารากที่สอง
  - แสดงรากที่สองที่หาได้ทางจอภาพ
- Design
  - เขียนผังงาน
- Implementation
  - เขียนโปรแกรม

# ผังงาน โปรแกรม และการทดสอบ



```
import jlab.JLabIO;

public class Sqrt {
    public static void main(String [] args) {
        double x = JLabIO.readDouble("x = ");
        System.out.println("sqrt(x) = " +
            Math.sqrt(x) );
    }
}
```

```
JLab>java Sqrt
```

```
x = 1.44
```

```
sqrt(x) = 1.2
```

```
JLab>
```

## การทดสอบ

---

---

- อะไรจะเกิดขึ้นถ้าผู้ใช้ป้อนจำนวนติดลบ

```
import jlab.JLabIO;

public class Sqrt {
    public static void main(String [] args) {
        double x = JLabIO.readDouble("x = ");
        System.out.println("sqrt(x) = " +
            Math.sqrt(x) );
    }
}
```

```
JLab>java Sqrt
x = -8
sqrt(x) = NaN
JLab>
```

เมื่อ  $x < 0$  `Math.sqrt(x)` จะคืนค่าที่มีความหมายว่าไม่ใช่จำนวน (not a number) ค่านี้จะถูกเปลี่ยนเป็น "NaN" เมื่อแสดงทางจอภาพ ผู้ใช้ทั่วไปอาจไม่เข้าใจความหมาย NaN ควรแก้โปรแกรมให้สื่อความหมาย

# โปรแกรมหารากที่สอง (ฉบับปรับปรุง)

---

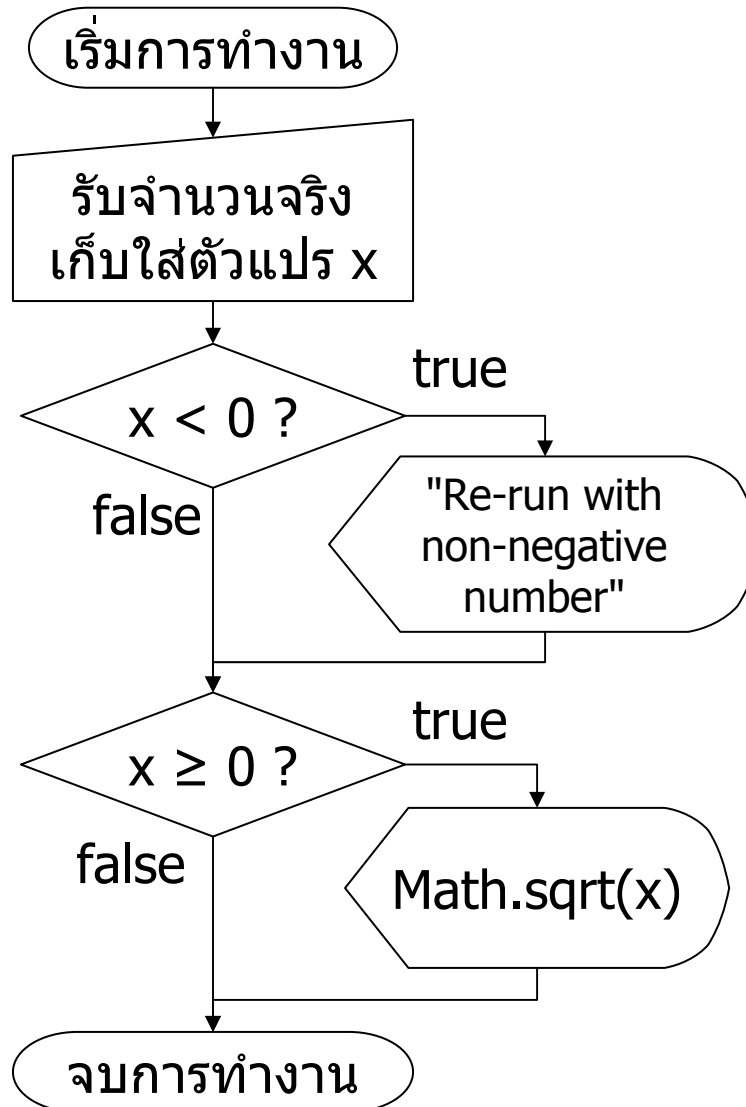
---

- Requirement
  - อ่านจำนวนจากผู้ใช้ แล้วแสดงรากที่สองของจำนวนนั้น ถ้าเป็นจำนวนไม่ติดลบ
- Analysis
  - รับจำนวนจริงผ่านทางแป้นพิมพ์
  - ถ้าติดลบ ให้แสดง "Re-run with non-negative number"
  - ถ้าเป็นบวกหรือ 0 ให้แสดงรากที่สองที่หาได้ทางจอภาพ
- Design
  - เขียนผังงาน
- Implementation
  - เขียนโปรแกรม

# ผังงาน

---

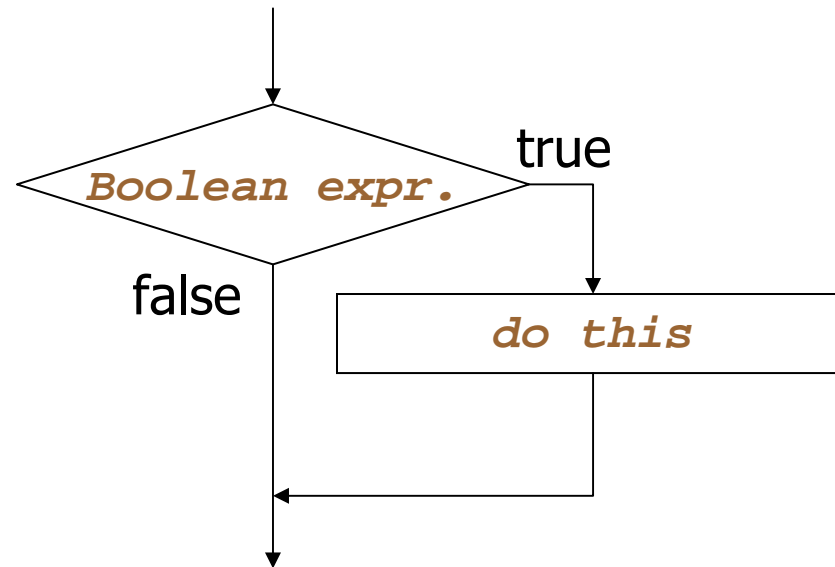
---



# Selection : if statement

---

---



```
if ( Boolean expr ) {  
    do this ;  
}
```



# โปรแกรม และการทดสอบ

```
import jlab.JLabIO;

public class Sqrt {
    public static void main(String [] args) {
        double x = JLabIO.readDouble("x = ");
        if (x < 0) {
            System.out.println("Re-run with non-negative number");
        }
        if (x >= 0) {
            System.out.println("sqrt(x) = " + Math.sqrt(x) );
        }
    }
}
```

```
JLab>java Sqrt
```

```
x = -8
```

```
Re-run with non-negative number
```

```
JLab>
```

```
JLab>java Sqrt
```

```
x = 1.21
```

```
sqrt(x) = 1.1
```

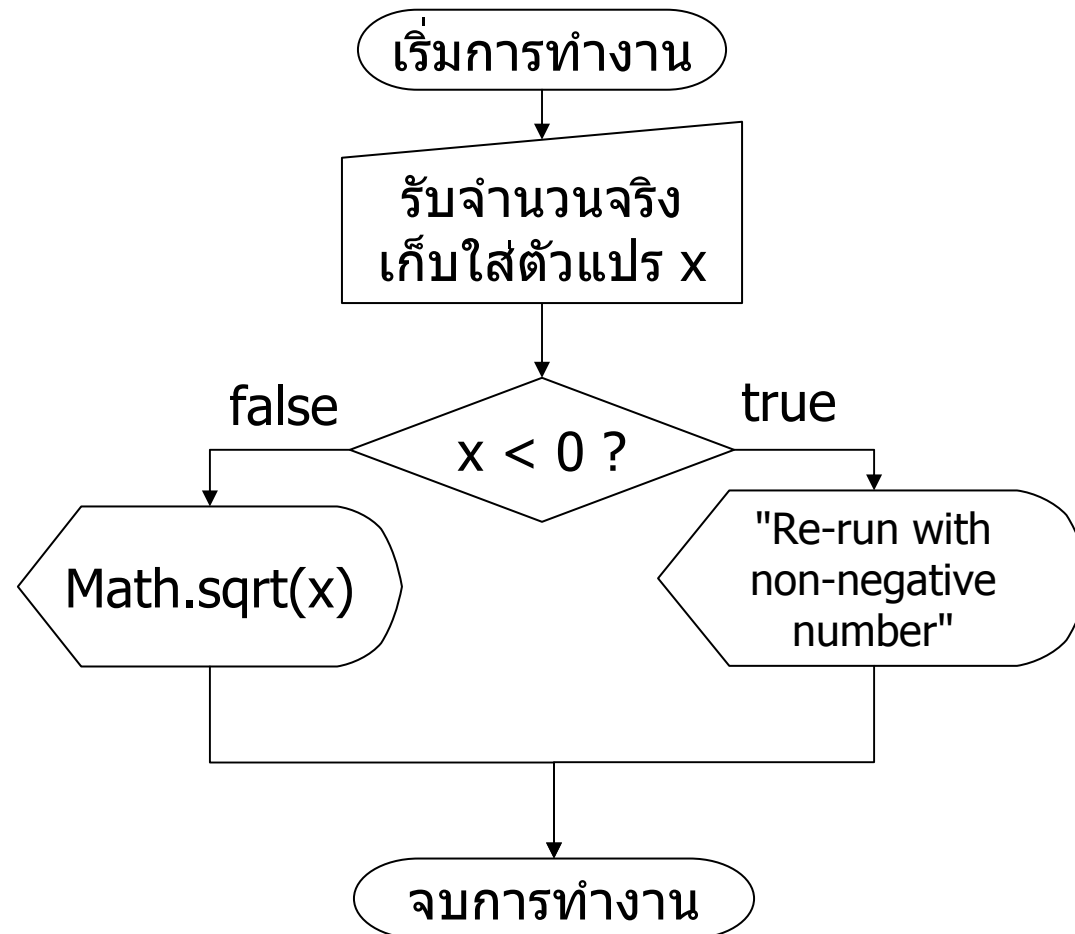
```
JLab>
```

ข้อสังเกต : สองกรณีนี้ต้องเป็นจริงกรณีใดกรณีหนึ่งแน่ๆ (และไม่มีทางเป็นจริงพร้อมกันทั้งคู่)

# ผังงาน

---

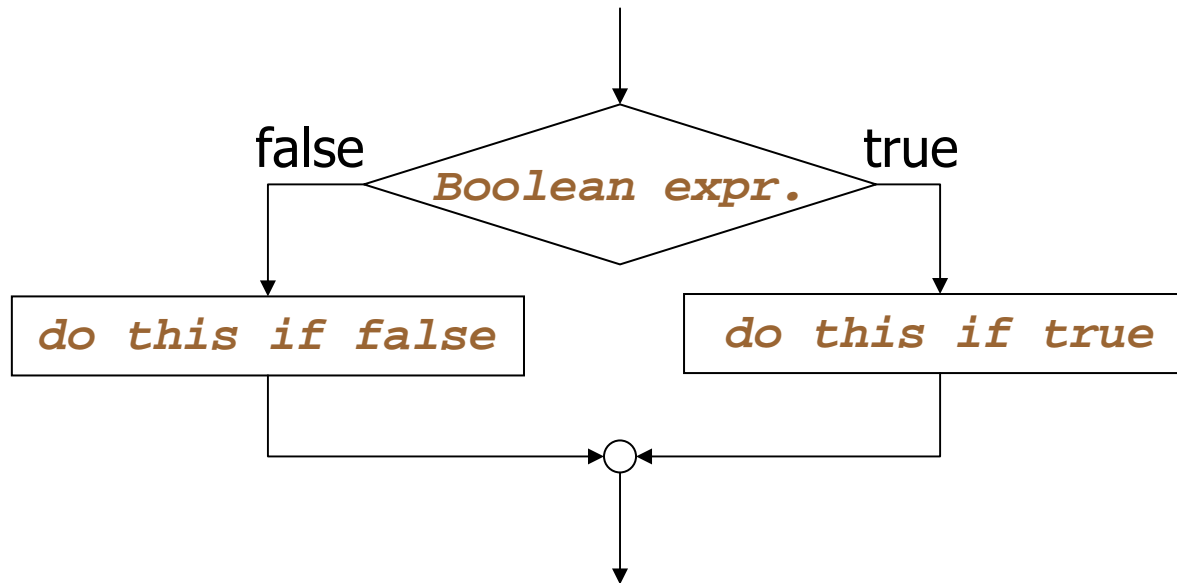
---



# Selection : if-else statement

---

---



```
if ( Boolean expr ) {  
    do this if true ;  
} else {  
    do this if false ;  
}
```

# โปรแกรม และการทดสอบ

```
import jlab.JLabIO;

public class Sqrt {
    public static void main(String [] args) {
        double x = JLabIO.readDouble("x = ");
        if (x < 0) {
            System.out.println("Re-run with non-negative number");
        } else {
            System.out.println("sqrt(x) = " + Math.sqrt(x) );
        }
    }
}
```

```
JLab>java Sqrt
```

```
x = -8
```

```
Re-run with non-negative number
```

```
JLab>
```

```
JLab>java Sqrt
```

```
x = 1.21
```

```
sqrt(x) = 1.1
```

```
JLab>
```

# นิพจน์บูลีน (Boolean Expression)

---

---

- ได้ผลลัพธ์เป็นประเภท **boolean**
- สร้างมาจากการใช้
  - relational operator : ตัวเปรียบเทียบข้อมูลแบบพื้นฐาน เช่น มากกว่า น้อยกว่า เท่ากัน
  - method ที่คืนค่าประเภท **boolean**
  - boolean operator : ตัวเชื่อมความสัมพันธ์นิพจน์บูลีน เช่น and or not
- **boolean** เปลี่ยนเป็น **String** ได้เท่านั้น

```
public class TrueFalse {  
    public static void main(String [] args) {  
        System.out.println( true + " " + false );  
    }  
}
```

```
JLab>java TrueFalse  
true false  
JLab>
```



# Boolean Operators

---

---

- ใช้เครื่องหมาย && แทน and
- ใช้เครื่องหมาย || แทน or
- ใช้เครื่องหมาย ! แทน not
- and or เป็นตัวปฏิบัติการเพื่อเชื่อมนิพจน์บูลีนสองนิพจน์ ได้เป็นนิพจน์บูลีนใหม่
  - ให้ A และ B เป็นนิพจน์บูลีน
  - $A \ \&\& \ B$  เป็น true เมื่อ ทั้ง A และ B เป็น true ทั้งคู่
  - $A \ || \ B$  เป็น true เมื่อ A หรือ B เป็น true สักหนึ่งตัว
- not เป็นตัวปฏิบัติการเพื่อเปลี่ยนผลของนิพจน์บูลีนที่กำหนดให้ จากจริงเป็นเท็จ หรือจากเท็จเป็นจริง

# โปรแกรมแสดงชื่อเดือน

- รับหมายเลขเดือน (มีค่าได้ตั้งแต่ 1 ถึง 12)
- แสดงชื่อเดือนตามหมายเลขที่ได้รับ
- ให้  $m$  แทนหมายเลขเดือนที่ได้รับ
- สตริงย่อยตั้งแต่ตัวอักษรตัวที่  $9(m-1)$  ถึง  $9m-1$  ก็คือชื่อเดือน
- ถ้าผู้ใช้ป้อนเดือนที่ 2 ก็หยิบสตริงย่อยข้างล่างนี้

										1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3		
J	A	N	U	A	R	Y				F	E	B	R	U	A	R	Y			M	A	R	C	H	...



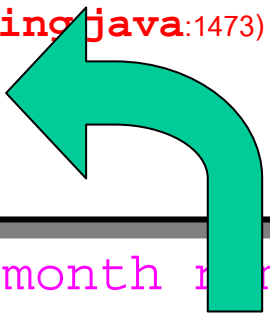
# โปรแกรมแสดงชื่อเดือน

- อะไรจะเกิดขึ้นถ้าผู้ใช้ป้อนเลขเดือนที่ 100

```
import jlab.JLabIO;

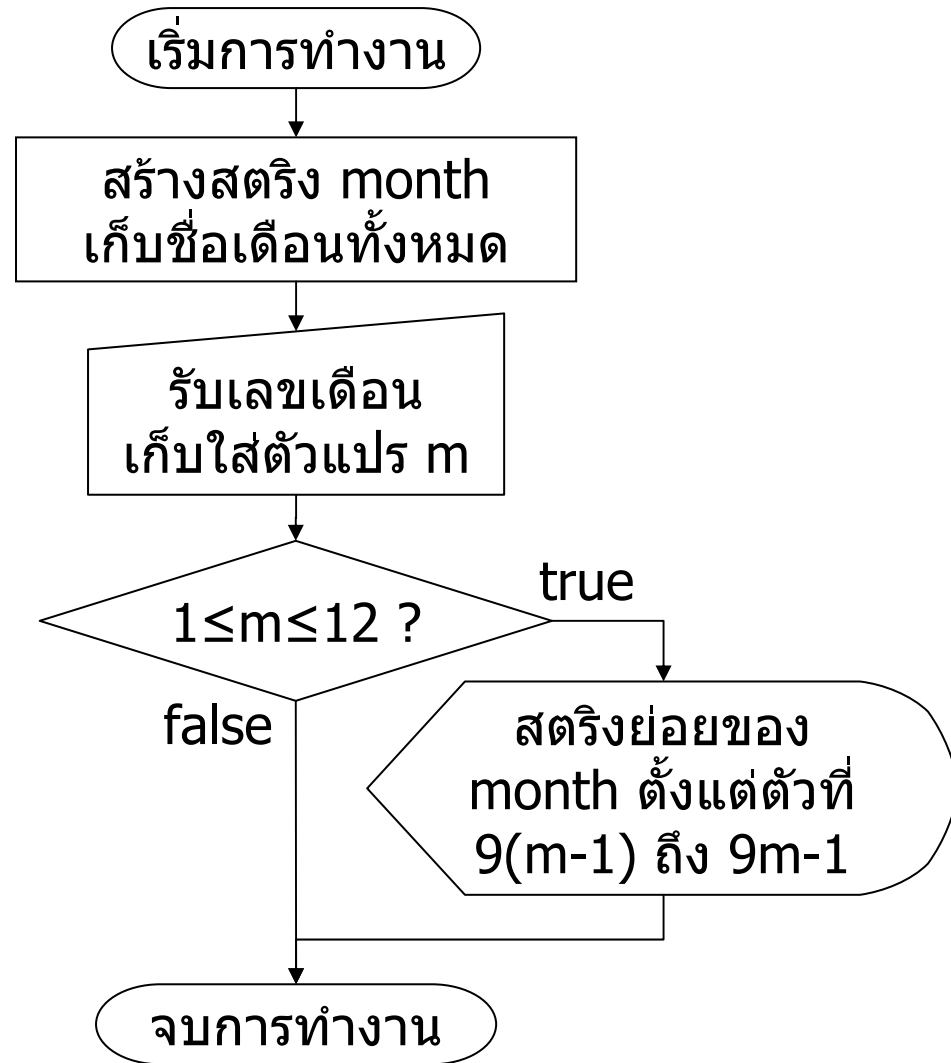
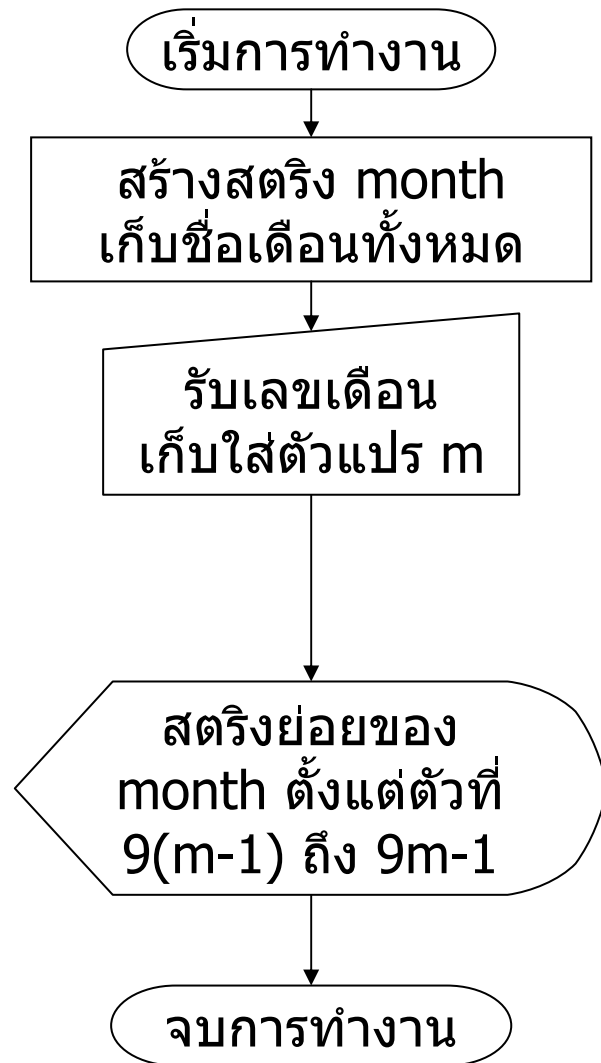
public class Month {
    public static void main(String[] args) {
        int m = JLabIO.readInt("Enter month number : ");
        System.out.println(month.substring(9*(m-1), 9*m));
    }
}
```

JLab>java Month  
Enter month number : 100  
java.lang.StringIndexOutOfBoundsException: String index out of range: 900  
 at java.lang.String.substring(String.java:1473)  
 at Month.main(Month.java:10)  
Exception in thread "main"  
JLab>



เกิด **StringIndexOutOfBoundsException**

# การตรวจสอบเงื่อนไข

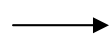


# โปรแกรมแสดงชื่อเดือน

```
import jlab.JLabIO;

public class Month {
    public static void main(String [] args) {
        String month = "";
        month += "JANUARY  FEBRUARY MARCH    APRIL    ";
        month += "MAY        JUNE      JULY     AUGUST   ";
        month += "SEPTEMBER OCTOBER  NOVEMBER DECEMBER ";
        int m = JLabIO.readInt("Enter month number : ");
        if ((1 <= m) && (m <= 12)) {
            System.out.println(month.substring(9*(m-1), 9*m));
        }
    }
}
```

$1 \leq m \leq 12$



$(1 \leq m) \ \&\& \ (m \leq 12)$

# เงื่อนไขหนึ่ง เขียนได้หลายแบบ

---

---

- $(1 \leq m) \ \&\& \ (m \leq 12)$
- $(0 < m) \ \&\& \ (m < 13)$
- $(12 \geq m) \ \&\& \ (1 \leq m)$
- $(m == 1) \ || \ (m == 2) \ || \ (m == 3) \ || \ (m == 4) \ || \ (m == 5) \ || \ (m == 6) \ || \ (m == 7) \ || \ (m == 8) \ || \ (m == 9) \ || \ (m == 10) \ || \ (m == 11) \ || \ (m == 12)$
- $! \ ( \ !(1 \leq m) \ || \ !(m \leq 12) \ )$
- $! \ ( \ (m < 1) \ || \ (m > 12) \ )$

## ตารางความจริง

---

---

- $(1 \leq m) \ \&\& \ (m \leq 12)$  คือ  $!((m < 1) \ || \ (m > 12))$  ?

$(1 \leq m)$	$(m \leq 12)$	$(1 \leq m) \ \&\& \ (m \leq 12)$
true	true	true
true	false	false
false	true	false
false	false	false

$(m < 1)$	$(m > 12)$	$(m < 1) \    \ (m > 12)$	$!((m < 1) \    \ (m > 12))$
false	false	false	true
false	true	true	false
true	false	true	false
true	true	true	false

# กฎการพิจารณาลำดับการปฏิบัติการ

---

---

- ภายในวงเล็บ
- unary + - ++ -- !
- type casting
- \* / %
- binary + -
- < > <= >=
- == !=
- &&
- ||
- = \*= /= %= += -=

```
int k;
...
boolean b = ! k==0; // wrong
...

if ( 3<=k+1 && k<=5 || k>=9 ) {
    // k=2,3,4,5, 9,10,11,...
}
```

# โปรแกรมแสดงจำนวนวันของเดือน ก.พ.

---

---

- Requirement
  - หาว่าเดือนก.พ. ของปีที่ให้มามี 28 หรือ 29 วัน
- Analysis
  - รับเลขปี (เป็น ค.ศ.) ผ่านทางแป้นพิมพ์
  - ใช้การทดสอบปีอธิกสุรทินเพื่อแสดงจำนวนวันในเดือน ก.พ.
- Design
  - เขียนผังงาน
- Implementation
  - เขียนโปรแกรม

# การทดสอบปีอธิกสุรทิน

---

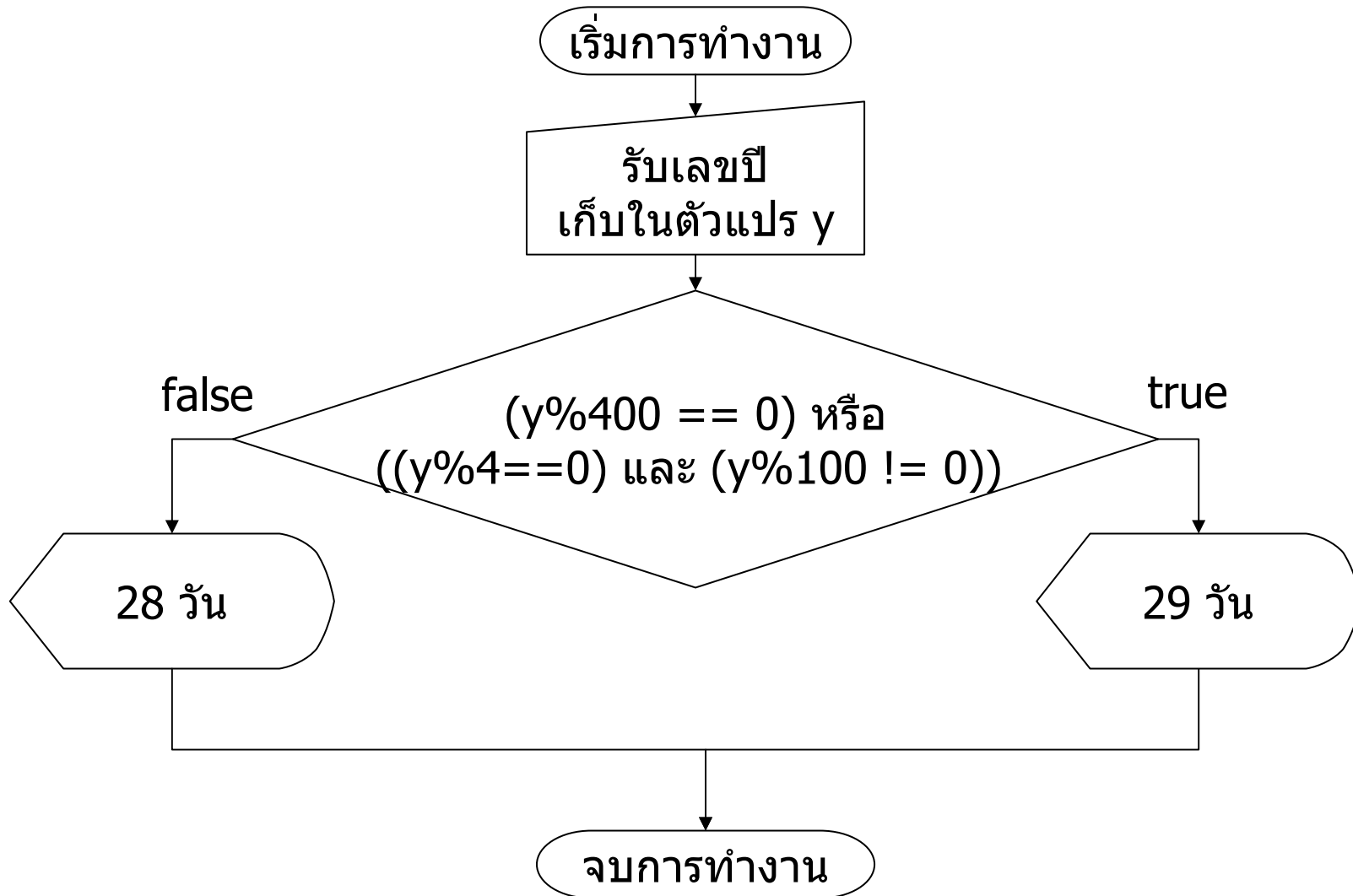
---

- เดือนกุมภาพันธ์ของปี  $y$  (ค.ศ.) มี 29 วันเมื่อ
  - เลขปี (ค.ศ.) ต้องหารด้วย 4 ลงตัว แต่ต้องหารด้วย 100 ไม่ลงตัว ยกเว้นกรณีที่หารด้วย 400 ลงตัว
    - ปี 2002 ก.พ. มี 28 วัน (หารด้วย 4 ไม่ลงตัว)
    - ปี 2004 ก.พ. มี 29 วัน (หารด้วย 4 ลงตัว และหารด้วย 100 ไม่ลง)
    - ปี 2100 ก.พ. มี 28 วัน (หารด้วย 100 ลงตัว)
    - ปี 2000 ก.พ. มี 29 วัน (หารด้วย 400 ลงตัว)
  - ให้  $y$  คือเลขปี (ค.ศ.)
    - $(y \% 4) == 0$  แทนการทดสอบว่าหารด้วย 4 ลงตัว ?
    - $(y \% 100) != 0$  แทนการทดสอบว่าหารด้วย 100 ไม่ลงตัว ?
    - $(y \% 400) == 0$  แทนการทดสอบว่าหารด้วย 400 ลงตัว ?

```
if ( (y%4 == 0 && y%100 != 0) ||  
      (y%400 == 0) )
```



# ผังงาน

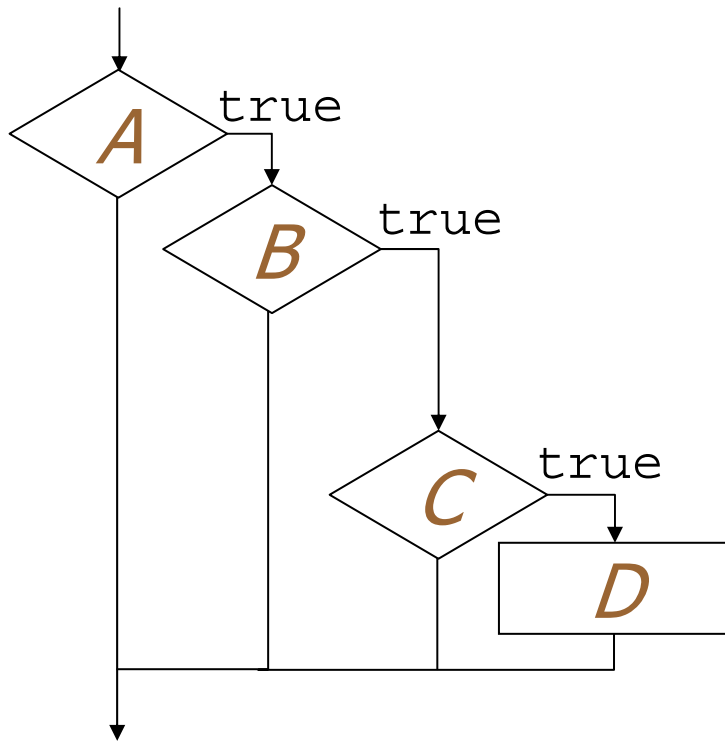


# โปรแกรมแสดงจำนวนวันของเดือน ก.พ.

```
import jlab.JLabIO;
public class Feb {
    public static void main(String [] args) {
        int y = JLabIO.readInt("year = ");
        if ((y%4==0 && y%100!=0) || (y%400==0)) {
            System.out.println("Feb. has " + 29 + " days.");
        } else {
            System.out.println("Feb. has " + 28 + " days.");
        }
    }
}
```

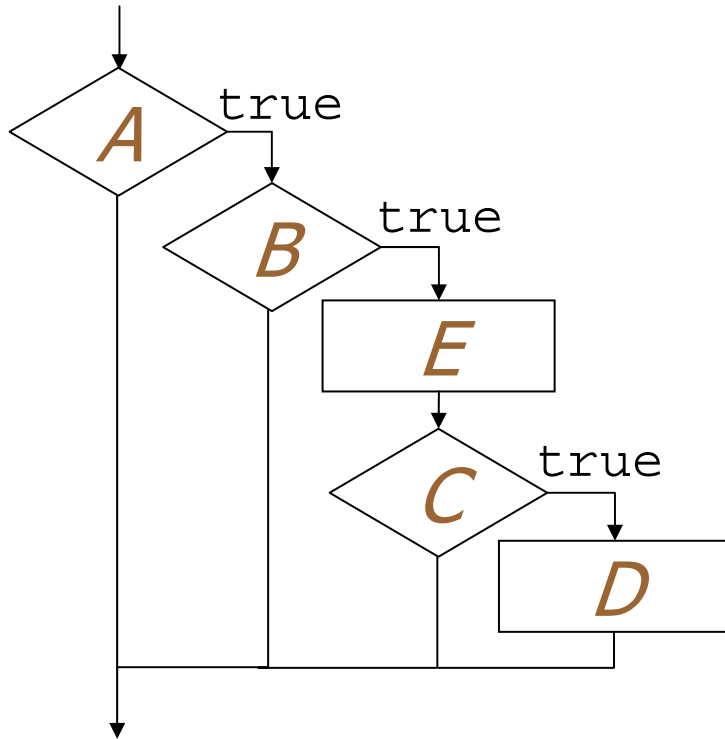
```
int y = JLabIO.readInt("year = ");
int numDays;
if ((y%4==0 && y%100!=0) || (y%400==0)) {
    numDays = 29;
} else {
    numDays = 28;
}
System.out.println("Feb. has " + numDays + " days.");
```

# Nested If



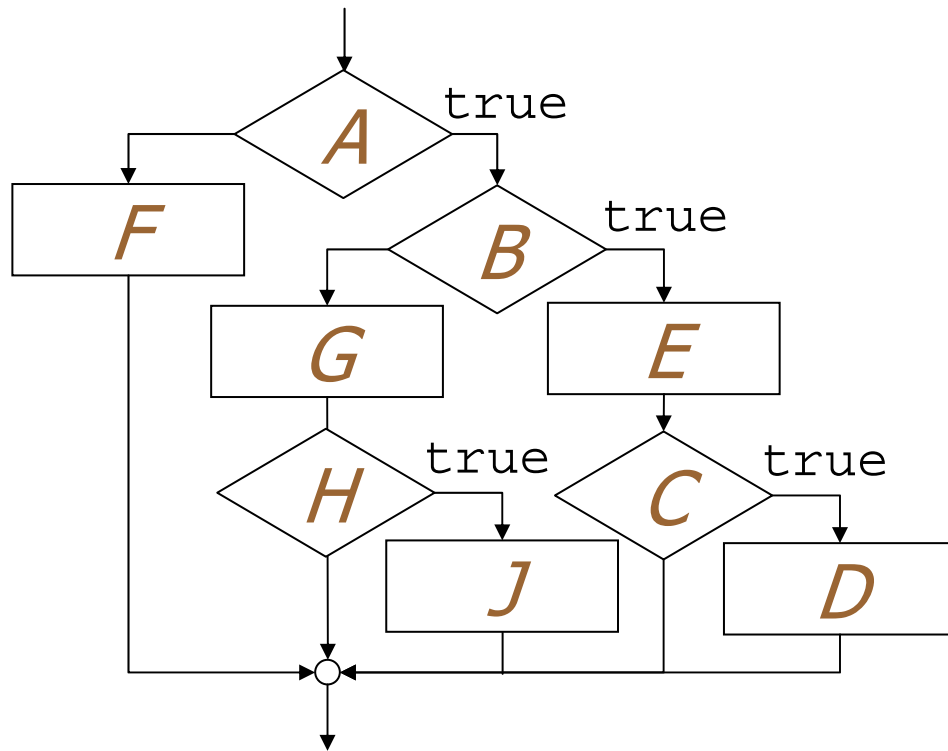
```
if ( A ) {  
  if ( B ) {  
    if ( C ) {  
      D ;  
    }  
  }  
}
```

# Nested If



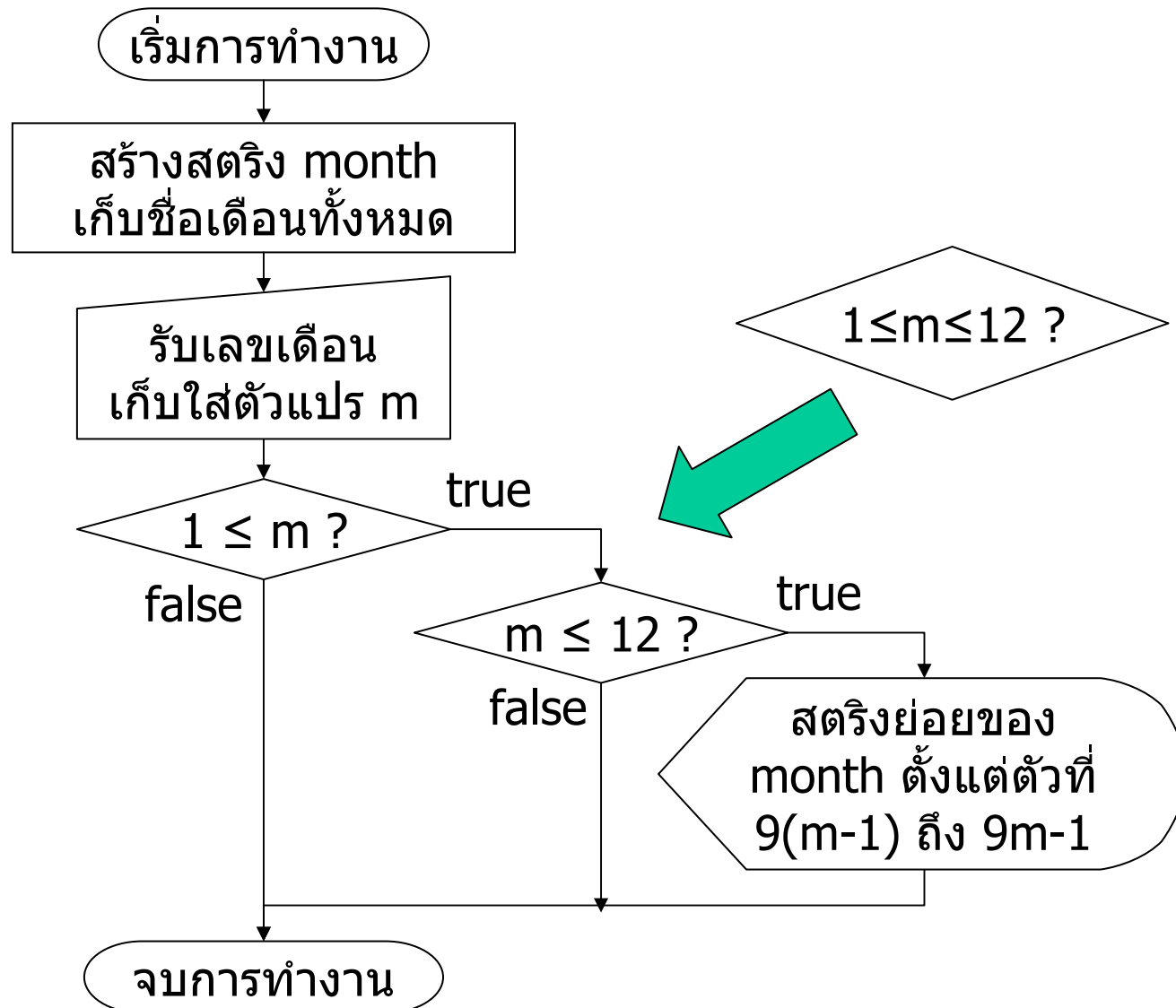
```
if ( A ) {  
    if ( B ) {  
        E ;  
        if ( C ) {  
            D ;  
        }  
    }  
}
```

# Nested If



```
if ( A ) {  
  if ( B ) {  
    E ;  
    if ( C ) {  
      D ;  
    }  
  } else {  
    G ;  
    if ( H ) {  
      J ;  
    }  
  }  
} else {  
  F ;  
}
```

# ผังงานแสดงชื่อเดือน



# โปรแกรมแสดงชื่อเดือน

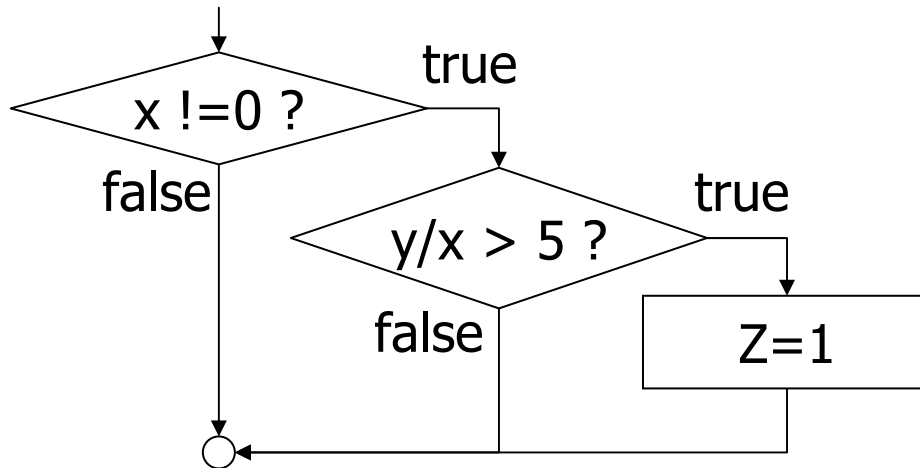
---

---

```
import jlab.JLabIO;

public class Month {
    public static void main(String [] args) {
        String month = "";
        month += "JANUARY  FEBRUARY MARCH    APRIL    ";
        month += "MAY      JUNE      JULY    AUGUST  ";
        month += "SEPTEMBER OCTOBER  NOVEMBER DECEMBER ";
        int m = JLabIO.readInt("Enter month number : ");
        if (1 <= m) {
            if (m <= 12) {
                System.out.println(month.substring(9*(m-1), 9*m));
            }
        }
    }
}
```

# Short-circuiting



```
if (x != 0) {  
    if (y/x > 5) {  
        z = 1;  
    }  
}
```

```
if (x != 0 && y/x > 5) {  
    z = 1;  
}
```

```
if (y/x > 5 && x != 0) {  
    z = 1;  
}
```

ถ้า  $x == 0$  จะไม่มีการคำนวณ  $y/x$

คำนวณ  $y/x$  ก่อนทดสอบ  $x != 0$

Short-circuiting (ไม่คิดหมด พยายามหาทางลัด)

- กรณี  $\&\&$  จะเลิกประเมิน expr ทางขวาถ้าพบ expr ที่เป็น false
- กรณี  $\|\|$  จะเลิกประเมิน expr ทางขวาถ้าพบ expr ที่เป็น true



# Statement และ Block

---

---

- Block คือกลุ่มของบรรทัดคำสั่ง ครอบด้วย { }

```
if (Boolean Expression)  
    a statement ;
```

```
if (Boolean Expression)  
    a statement ;  
else  
    a statement ;
```

```
if (Boolean Expression) {  
    statements ;  
}
```

```
if (Boolean Expression) {  
    statements ;  
} else {  
    statements ;  
}
```

if statement

if-else statement

# เขียนได้หลายแบบ

---

---

<pre>if (x &lt; 0) {     x = -x; }</pre>	<pre>if (x &lt; 0)     x = -x;</pre>	<pre>if (x &lt; 0) x = -x;</pre>
<pre>if (x &lt; 0) {     done = true; } else {     done = false; }</pre>	<pre>if (x &lt; 0)     done = true; else     done = false;</pre>	<pre>done = (x &lt; 0);</pre>

# โปรแกรมแสดงชื่อเดือน

```
import jlab.JLabIO;

public class Month {
    public static void main(String [] args) {
        String month = "Illegal month number";
        int m = JLabIO.readInt("Enter month number : ");
        if (m == 1) month = "JANUARY";
        if (m == 2) month = "FEBRUARY";
        if (m == 3) month = "MARCH";
        if (m == 4) month = "APRIL";
        if (m == 5) month = "MAY";
        if (m == 6) month = "JUNE";
        if (m == 7) month = "JULY";
        if (m == 8) month = "AUGUST";
        if (m == 9) month = "SEPTEMBER";
        if (m == 10) month = "OCTOBER";
        if (m == 11) month = "NOVEMBER";
        if (m == 12) month = "DECEMBER";
        System.out.println(month);
    }
}
```

# โปรแกรมแสดงชื่อเดือน

```
import jlab.JLabIO;

public class Month {
    public static void main(String [] args) {
        String month = "Illegal month number";
        int m = JLabIO.readInt("Enter month number : ");
        if (m == 1) month = "JANUARY";
        else if (m == 2) month = "FEBRUARY";
        else if (m == 3) month = "MARCH";
        else if (m == 4) month = "APRIL";
        else if (m == 5) month = "MAY";
        else if (m == 6) month = "JUNE";
        else if (m == 7) month = "JULY";
        else if (m == 8) month = "AUGUST";
        else if (m == 9) month = "SEPTEMBER";
        else if (m == 10) month = "OCTOBER";
        else if (m == 11) month = "NOVEMBER";
        else if (m == 12) month = "DECEMBER";
        System.out.println(month);
    }
}
```

# switch - case

```
import jlab.JLabIO;
public class Month {
    public static void main(String [] args) {
        int m = JLabIO.readInt("Enter month number : ");
        switch (m) {
            case 1 : System.out.println("JANUARY"); break;
            case 2 : System.out.println("FEBRUARY"); break;
            case 3 : System.out.println("MARCH"); break;
            case 4 : System.out.println("APRIL"); break;
            case 5 : System.out.println("MAY"); break;
            case 6 : System.out.println("JUNE"); break;
            case 7 : System.out.println("JULY"); break;
            case 8 : System.out.println("AUGUST"); break;
            case 9 : System.out.println("SEPTEMBER"); break;
            case 10 : System.out.println("OCTOBER"); break;
            case 11 : System.out.println("NOVEMBER"); break;
            case 12 : System.out.println("DECEMBER"); break;
            default : System.out.println("Illegal month number");
        }
    }
}
```

# switch-case

---

---

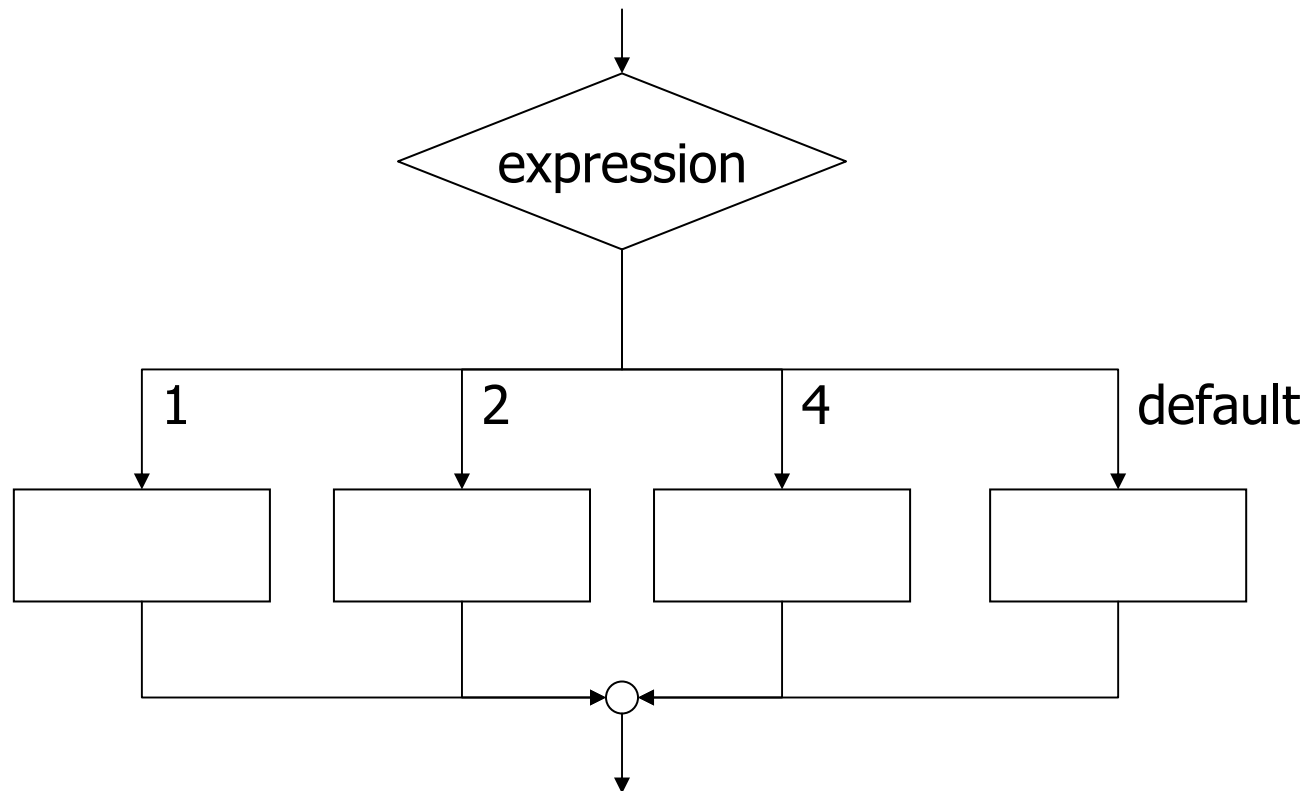
- ในวงเล็บหลัง **switch** ต้องเป็น **expr** ที่ได้ผลเป็นจำนวนเต็ม
- ตามหลัง **case** ต้องเป็นค่าคงตัวแบบจำนวนเต็ม
- ท้ายสุดของ **case** ถ้ามี **break** จะกระโดดออกนอก block
- ถ้าท้ายสุดของ **case** ไม่มี **break** การทำงานจะไหลต่อไปทำงานใน **case** ถัดลงไป
- ถ้าไม่ตรง **case** ใดเลย การทำงานจะไปที่ **default**

```
switch (m) {  
    case 1 :  
    case 2 :  
        System.out.println("m = 1 or 2");  
        break;  
    case 4 :  
        System.out.println("m = 4");  
        break;  
    default :  
        System.out.println("m != 1, 2, and 4");  
}
```

# switch-case

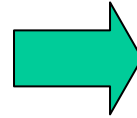
---

---



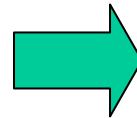
# สิ่งที่พบว่าผิดกันบ่อย

```
if (0 <= x <= 10)
if x == 0
if (x >= 0 && <= 10)
if (x == 0 || == 1)
if (x = 0)
if (x => 0)
```



```
if (0 <= x && x <= 10)
if (x == 0)
if (x >= 0 && x <= 10)
if (x == 0 || x == 1)
if (x == 0)
if (x >= 0)
```

```
if (x == 0)
    if (y == 0) z = 0;
else if (a == 1)
    z = 1;
```



```
if (x == 0) {
    if (y == 0) z = 0;
} else if (a == 1)
    z = 1;
```

เขียนไปไว้ค่า

```
if (x != 0 || x != 5)
if (x < 3 && x > 9)
if (done && !done)
if (done || !done)
```



# อย่าใช้ == เพื่อเปรียบเทียบ floating point

- การเก็บจำนวนจริงด้วย `float` และ `double` มีข้อจำกัดเรื่องความแม่นยำ
- การคำนวณจึงมักทำให้ผลลัพธ์มีความคลาดเคลื่อน
- ไม่ควรนำจำนวนจริงมาทดสอบว่าเท่ากันหรือไม่

```
public class Imprecision {  
    public static void main(String [] args) {  
        double a = 49.0;  
  
        System.out.println( 1.0 == (1.0/a * a) );  
        System.out.println( 1.0 - (1.0/a * a) );  
    }  
}
```

```
JLab>java Imprecision  
false  
1.1102230246251565E-16  
JLab>
```

## การเปรียบเทียบจำนวนจริง $x == y$ ?

- ตั้งค่า  $\epsilon$  ไว้ที่  $10^{-14}$  ถ้าผ่านการทดสอบข้างขวานี้ ก็แสดงว่า  $x$  และ  $y$  มีค่าใกล้เคียงกันมาก

$$\frac{|x - y|}{\max(|x|, |y|)} \leq \epsilon$$

```
public class Epsilon {
    public static void main(String [] args) {
        double EPSILON = 1.0E-14;
        double a = 49.0;
        double x = 1.0, y = (1.0/a * a);
        boolean eq = (Math.abs(x-y) /
                     Math.max(Math.abs(x), Math.abs(y)))
                     <= EPSILON;
        System.out.println( eq );
    }
}
```

```
JLab>java Epsilon
true
JLab>
```

## การใช้ == < > เพื่อเปรียบเทียบ char

---

---

- ใช้ == < > เพื่อเปรียบเทียบ char ได้ตามลำดับตัวอักษร (lexicographic ordering)
- การเปรียบเทียบอักขระคือการเปรียบเทียบตามรหัส Unicode ที่ใช้แทนอักขระนั้น

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
002		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
003	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
004	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
005	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
006	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
007	p	q	r	s	t	u	v	w	x	y	z	{		}	~	

## อย่าใช้ `==` `<` `>` เพื่อเปรียบเทียบ **String**

---

---

- **String** ไม่ใช่ประเภทพื้นฐาน
- ต้องใช้ methods `equals` และ `compareTo` ในการเปรียบเทียบ

`s1.equals(s2)` =  $\begin{cases} \text{true} & \text{ถ้า } s1 \text{ เหมือนกับ } s2 \\ \text{false} & \text{ถ้า } s1 \text{ ไม่เหมือนกับ } s2 \end{cases}$

`s1.compareTo(s2)`  $\begin{cases} < 0 & \text{ถ้า } s1 \text{ มาก่อน } s2 \text{ ตามลำดับตัวอักษร} \\ = 0 & \text{ถ้า } s1 \text{ เหมือนกับ } s2 \\ > 0 & \text{ถ้า } s1 \text{ มาหลัง } s2 \text{ ตามลำดับตัวอักษร} \end{cases}$

# การเปรียบเทียบ **String**

---

---

```
public class StringCompare {
    public static void main(String [] args) {
        System.out.println("Computer".equals("Computer"));
        System.out.println("Computer".compareTo("Computer"));
        System.out.println("Computer".compareTo("Engineer"));
        System.out.println("Computer".compareTo("Comp"));
        System.out.println("Computer".compareTo("Computing"));
    }
}
```

```
JLab>java StringCompare
```

```
true
```

```
0
```

```
-2
```

```
4
```

```
-4
```

```
JLab>
```