

24

FEBRUARY
SATURDAY
DAYS 055-311

S	M	T	W	T	F	S
1	2	3				
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29		

FEBRUARY '24

Symbol

OPERATOR

Associativity

Precedence

()

Function call

L to R

Hig 14

[]

Array subscript

*

Multiplication

•

Dot (Member of structure)

/

Division

->

Arrow

%

Modulus (Remainder)

!

Logical NOT

+

Addition

°

-

Subtraction

-

One's complement

<<

Left-shift

++

>>

Right-shift

--

Change minus

<

Less than

&

Increment

<=

Less than or equal to

&

Decrement

>

Greater than

()

Address of

>=

Greater than or equal to

(Sunday)

Cast

==

Equal to

Size of

!=

Not equal to

2024

R 2 L

13

26

FEBRUARY
MONDAY
057-309 DAYS

S	M	T	W	T	F	S
31			1	2		
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

MARCH '24

12

2024

27

FEBRUARY
TUESDAY
DAYS 058-308

S	M	T	W	T	F	S
					1	2
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29		

FEBRUARY '24

Operators

+	Bitwise AND	L2R	8
-	Bitwise XOR	L2R	7
!	Bitwise OR	L2R	6
	Logical OR	L2R	5
==	Conditional L AND	L2R	4
?:	Conditional	R2L	3
=, +=	Ass D -		2
+=, *=.	Assignment operator	R to 2	I
,	Comma	L2R	O

S	M	T	W	T	F	S
31			1	2		
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

MARCH 24

FEBRUARY
WEDNESDAY
059-307 DAYS

28

Resulting data type after arithmetic operation.

$$R = b + s \rightarrow \text{int}$$

$$R = s + l \rightarrow \text{int}$$

$$R = d + f \rightarrow \text{float}$$

$$R = d + f \rightarrow \text{float}$$

$$R = c + q \rightarrow \text{int}$$

$$R = c + s \rightarrow \text{int}$$

$$R = l + d \rightarrow \text{double}$$

$$R = f + d \rightarrow \text{double}$$

$$b + s \rightarrow \text{int}$$

$\left\{ \begin{array}{l} B \rightarrow \text{byte} \\ F \rightarrow \text{float} \\ S \rightarrow \text{short} \\ T \rightarrow \text{int} \\ C \rightarrow \text{charach} \\ L \rightarrow \text{long} \end{array} \right.$

29

FEBRUARY
THURSDAY
DAYS 060-306

S	M	T	W	T	F	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29		

FEBRUARY '24

Explicit casting :-

Convert one data to another data.

Explicit Casting defined by programmer.

In Explicit there is data loss.

Explicit tells the compiler that you are ~~not~~ aware of this and accept the potential loss of data.

Eg 1 public class Explicit_Casting
 {
 public static void main
 (String [] args) {
 float a=5.2f;

 int n=5;

 int b = (int) (y+n);

 System.out.println(b);

S	M	T	W	T	F	S
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

APRIL '24

MARCH
FRIDAY
061-305 DAYS

01

Increment & Decrement

Increment operator is used to increment a value by 1. There.

are two varieties of increment operator.

Post increment & Pre-increment

Post increment : Value is first for computing the result then increment.

Pre-increment : Value is incremented first and then the result is computed

Eg: $\text{int } a = 10;$

$\text{int } b = \text{post increment } a++;$

Output = 10.

$\text{int } a = 10;$

$\text{int } b = ++a;$

Output is 11.

2024

02

MARCH
SATURDAY
DAYS 062-304

S	M	T	W	T	F	S
31				1	2	
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

MARCH 24

APRIL '24

SUNDAY

$$5++ \rightarrow 6$$

$$6 - 7$$

$$\begin{matrix} 6 \\ 6 \\ 7 \end{matrix}$$

$$qnt = a = 10 !$$

$$b = 10 \# 11$$

$$b = 10 ++$$

// Decrement operator if $I+$ is used for decrement the value by 1

There are two varieties of decrement operator

- Post & Pre-Decrement.

Note :-

public class AFGP

```
public static void
main (String [] str)
```

{

$$\text{Int } a = 10 !$$

$$\text{Int } b = ++a !$$

03 SUNDAY

2024

S	M	T	W	T	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

APRIL 24

MARCH
MONDAY
064-302 DAYS

04

$b = 10++;$ //error.

System.out.println(b);

3

3

error ~~in~~ Ps $b = 10++;$

Because increment & decrement operators can only be applied to variable, Not in Literals or constants

Literals (Right side hand value).

~~int a = a + t;~~ } correct.

~~int t = t + a;~~ }

$10++;$ } wrong.

$++10$ ~~at;~~ } wrong.

Literals : are fixed values directly written in the code. They represent value that cannot be changed during the execution of the program.

Constant : are variables whose values cannot be changed once assigned.

In Java, Constants are typically declared using the 'final keyword' which makes variable immutable.

For eg let us assume

height is constant

age is constant.