

operator \Rightarrow ✓

\Rightarrow loops \Rightarrow conditionals (if-else) \Rightarrow pattern pg

{ loops + conditionals + operator \Rightarrow start to programming (X)

\Rightarrow `System.out.println();` or `System.out.print();`

\Rightarrow `System.out.println();` X

\Rightarrow loops:

\Rightarrow `for()`

\Rightarrow `while()`

\Rightarrow `do-while()`

\Rightarrow `for-each (enhanced for loop)` \Rightarrow arrays

`System.out.println();` 20 50
30

\Rightarrow for

```

graph TD
    Start([start]) --> Init[initialization]
    Init --> Check{condition check}
    Check -- No --> Stop([stop])
    Check -- Yes --> Body[body of loop]
    Body --> Update[update]
    Update --> Check
  
```

Learn | Screen | Share | Neun | Docu | YouT | Full S | Enter | Neun | Main | C# | Java | Enter | PS_J | Goog | Goog

keep.google.com/note/1066101260107.1361546462

Hyder Abbas

Handwritten notes on a grid background:

① ② ③ ④

for (inti=0; i<5; i++)

{

 s.o.p(" ");

}

0 1 2 3 4

i 0 1 2 3 4

i < 5

0 < 5

1 < 5

2 < 5

3 < 5

4 < 5

for (initiation; condition; update)

{

 Body

}

① ② ③ ④

initiation condition update

Body

⑤

* * * *

Learn | Screen | Share | Neun | Docu | YouT | Full S | Enter | Neun | Main | C# | Java | Enter | PS_J | Goog | Goog

keep.google.com/note/1066101260107.1361546462

Hyder Abbas

Handwritten notes on a grid background:

inti=0;

do

{

 s.o.p(" ");

 i++;

} while (i<n);

①

*

n 5

i 1 2 3 4 5

↓ ↓

if & only if condition true

once also not execute

Learn | Screen | Share | Neun | Docu | YouT | Full S | Enter | Neun | Main | C# | Java | Enter | PS_J | Goog | Goog

keep.google.com/note/1066101260107.1361546462

Hyder Abbas

Handwritten notes on a grid background:

⇒ while () ⇒

do while () ⇒

inti=5; inti=0; ① i=0 ✓

while (i<n) ②

{

 s.o.p(" ");

 i++;

}

③ ④

⇒ for ()

i 0 1 2 3 4 5

i < n

0 < 5

1 < 5

2 < 5

3 < 5

4 < 5

5 < 5

* * * * *

Handwritten notes on a grid background. At the top, there's a small diagram with a circle containing $n-1$ and arrows pointing to 3 and $n/2$. Below it, a larger diagram shows a circle with $n-1/2$ and arrows pointing to $i=0$ and $j=n-1$. To the right, there's a vertical sequence of asterisks with arrows pointing to them from the left. The text $i=0 \parallel j=n/2$ and $i=n-1 \parallel j=0$ is written in the center.

Handwritten notes on a grid background. A central diagram shows a square with dashed lines, labeled with 0 , $n/2$, and $n-1$ at its corners. To the left, there's a vertical sequence of asterisks with arrows pointing to them from the left. To the right, there's a vertical sequence of asterisks with arrows pointing to them from the left. The text $i=n/2$ and $j=n-1$ is written in the center.

Q>

Consider below code:

```

public class Test {
    public static void main(String[] args) {
        char c = 'Z';
        long l = 100_00L; //from JDK1.7 for a literal we can give _ also, if we give compiler will remove that _ in .class file
        int i = 9_2; //from JDK1.7 for a literal we can give _ also, if we give compiler will remove that _ in .class file
        float f = 2.02f;
        double d = 10_035d; //from JDK1.7 for a literal we can give _ also, if we give compiler will remove that _ in .class file

        l = c + i; //char + int = int int ----> long (implicit)
        f = c * l * i * f; //char * long * int * float = float
        f = l + i + c; //long + int + char = long long ----> float (implicit)
        i = (int)d; //double ----> int (explicit)
        f = (long)d; //double ----> long , long ----> float (implicit)
    }
}

```

Does above code compile successfully?

A. Yes
B. No

Answer : A


```

3
4 public static void main(String[] args) {
5     // TODO Auto-generated method stub
6
7     int n=10;
8     for(int i=0;i<n;i++)
9     {
10         for(int j=0;j<n;j++)
11         {
12

```

$\Rightarrow i=0 \text{ to } n-1$
 $j=0$
 $i=n-1 \text{ \& } j=n-1$
 $j=n-1 \text{ \& } i=0 \text{ to } n-1$

Q>

```

int x = 4; //line-n1
int y = 4++; //line-n2 whether it is post or pre-increment it can only be done on variables not on direct literals
System.out.println(x);
System.out.println(y);

```

A. line-n1 CompileTimeError
B. x=4
y=5
C. x=5
y=5
D. line-n2 CompileTimeError

Answer: D

```

public static void main(String[] args)
{
    int a = 20; // a = 18
    int var = --a * a++ + a---a; // var = 19 * 19 + 20 - 18 = 363
    System.out.println("a = " + a); // a = 18
    System.out.println("var = " + var); // var = 363
}

```

A.
a = 18
var=363
B.
a = 363
var=363
C. Compilation Error
D.
a = 25
var= 363

answer : A

18_10_2022_mipets_classnotes - Notepad

File Edit View

Q>
class Test
{
 public static void main(String[] args)
 {
 int i = 5; // i = 5,6,7
 //5 < 6(true)
 if (i++ < 6)
 {
 System.out.println(i++); // System.out.println(6)
 }
 }
}
A. 5
B. 6
C. Program executes successfully but nothing is printed on to console
D. 7

Ln 65, Col 1

22°C Cloudy

18_10_2022_mipets_classnotes - Notepad

File Edit View

Q>
boolean b=true; //line -n 1
b++; //line-n2 Ans. increment and decrement is applicable only for integral, floating type and character type not for boolean type
System.out.println(b);

A. line-n1 Compile Time Error
B. line-n2 Compile Time Error
C. false
D. true
E. None of the above

Answer : B

Ln 171, Col 1

22°C Cloudy

18_10_2022_mipets_classnotes - Notepad

File Edit View

D. line-n2 CompileTimeError

Answer: D

Q>
24
int x = 4; //line-n1
int y = ++(++x); //line-n2 whether it is post or pre-increment it can only be done on variables not on direct literals
System.out.println(x);
System.out.println(y);

A. line-n1 CompileTimeError
B. x=4
y=5
C. x=5
y=5
D. line-n2 CompileTimeError

Ln 96, Col 10

22°C Cloudy

18_10_2022_mipets_classnotes - Notepad

File Edit View

6



Nitin Manjunath

A. yes
B. no

Q>

```
int a = b = c = d = 20; // b,c,d not declared but using so CompileTime Error
System.out.println(a);
```

Will the code compile?

A. yes
B. no



Nitin Manjunath

C. 10
20

D. 20
10

Answer : B

```
Q>
int a = 10, b = 20;
byte c = (a>b) ? 30 : 40;
byte d = (a<b) ? 30 : 40;
System.out.println(c);
System.out.println(d);
```

A. 30
40

B. 40
30



Nishin Manjunath

Answer : B

Q>

```
int b,c,d;//declaring the variables
int a= b = c = d=10;
```

Will the code compile?

A. yes
B. no

Answer: A



18_10_2022_snppt_glametes - Notepad

File Edit View

Answer : B

Q>

byte c = (10 > 20) ? 30 : 40; // literals are involved so compiler performs operation
byte c = 40;
byte d = (10 < 20) ? 30 : 40; // literals are involved so compiler performs operation
byte d = 30;
System.out.println(c); // 40
System.out.println(d); // 30

A. 30
40

B. 40
30

C. 10
20

D. 20
10

Ln 153, Col 11

100% Windows (CRLF) UTF-8

22°C Cloudy

18_10_2022_snppt_glametes - Notepad

File Edit View

Q>

int a = 10, b = 20; // type checking is valid no problem
byte c = (a > b) ? 30 : 40; // literals are not involved in operation, so compiler would just check type checking of result
byte d = (a < b) ? 30 : 40;
System.out.println(c);
System.out.println(d);

A. 30
40

B. 40
30

C. 10
20

D. 20
10



E. CompiletimeError

Answer: E

Ln 175, Col 10

100% Windows (CRLF) UTF-8

22°C Cloudy



18_10_2022_support_classmate - Notepad

File Edit View

Q>
int a = 10, b = 20; // type checking is valid no problem
byte c = (a > b) ? 30 : 40; // literals are not involved in operation, so compiler would just check type checking of result
Compiler will see 30, 40 type it knows is int, so the result should be of int type only.
if compiler only perform the operation it will try to map with casting chart otherwise it wants the exact type.

byte d = (a < b) ? 30 : 40;
System.out.println(c);
System.out.println(d);

A. 30
40

B. 40
30

C. 10
20

D. 20
10

24

22°C Satisfactory air
Ln 163, Col 22

Windows (CRLF) UTF-8 100% 22:42 18-10-2022

18_10_2022_support_classmate - Notepad

File Edit View

Q>
Consider below statements:
1. int x = 5_0; // Literal values can be with _ also but it should be in b/w not at the beginning or at the end
2. int y = _50; // invalid becoz starts with _
3. int z = 50_; // invalid becoz in ends with _
4. float f = 123.76_86f; // valid
5. double d = 1_2_3_4; // valid

How many statements are legal?
A. One statement only
B. Two statement only
C. Three statement only
D. Four statement only
E. All 5 statement only.

Answer : C

25

22°C Cloudy
Ln 197, Col 11

Windows (CRLF) UTF-8 100% 22:44 18-10-2022