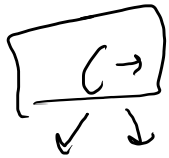
Module - 01 → Basic of Java:

- Syntax :-
- Operators
- Loop
- functions
- Break

Java → high level and object Oriented programming language

↳ Classes and object

↳ platform independent
↳ robust
↳ secure
↳ multi threading



→ software
→ web development
→ mobile development

Printing in Java

↳ System.out.println(" ");

My name is ABC.
I am working in NavGurukul.

or

↳ System.out.print(" ");

↳ My name is ABCI am working in NavGurukul.

Using "System.out.print"

```
public static void main(String[] args) {

    System.out.print("My name is ABC");
    System.out.print("I am working in NavGurukul");

}
```

Your Output

```
My name is ABCI am working in NavGurukul
```

Using "System.out.println()"

```
public static void main(String[] args) {

    System.out.println("My name is ABC");
    System.out.println("I am working in NavGurukul");

}
```

Your Output

```
My name is ABC
I am working in NavGurukul
```

Q→ Achieve functionality of System.out.println with the help of System.out.print

Ans → "\n"

```
System.out.print("My name is ABC \n");
System.out.print("I am working in NavGurukul");
```

My name is ABC
I am working in NavGurukul

```
public static void main(String[] args) {

    System.out.print("My name is ABC \n" );
    System.out.print("I am working in NavGurukul");

}
```

\n → next line

\n → next line

General statement :-

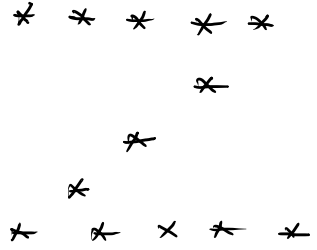
System.out.print() + "\n" = System.out.println()

```
System.out.println("I love Java");
System.out.print("I hate Python");
System.out.println("Hello Everyone");
```

I love Java
I hate PythonHello Everyone

Print the following patterns

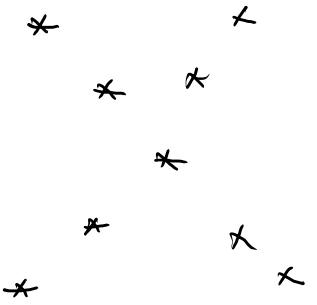
1



Z pattern

5 min

2



X pattern

5 min

```
System.out.println("*****");
System.out.println("    *");
System.out.println("   *");
System.out.println("  *");
System.out.println("*****");
```

Your Output

```
*****
    *
   *
  *
*****
```

```
System.out.print("*    *\n");
System.out.print(" * *\n");
System.out.print("  *\n");
System.out.print(" * *\n");
System.out.print("*    *");
```

Your Output

```
*    *
 * *
  *
 * *
*    *
```

5 minutes break

9:13 pm

Operators →

↳ symbols which are used to perform operations

↳ 5 types of operators.

(a) Arithmetic Operators.

(b) Assignment Operator.

(c) Logical Operators

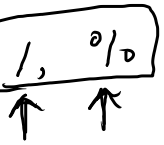
(d) Comparison Operator.

(e) Bitwise Operator.

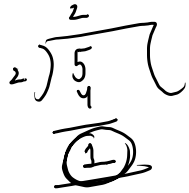
(1) Arithmetic Operator :-

↳ Operators used to perform basic mathematic operation.

↳ +, −, *, /, %,



→ Ex → 5 % 2 = 1



integer → 5 / 2 = 2

% → remainder

/ → quotient

(2) Assignment Operator :-

↳ is used to assign some value to a variable

↳ "=" ← assignment operator

Ex:- a = 10

a value '10' is assign to a variable "a"

+= Ex: a = a + 5 ⇓ a += 5

a = a - 5 ⇓ a -= 5

a = a / 5 ⇓ a /= 5

a = a % 5 ⇓ a %= 5

a = a * 5 ⇓ a *= 5

(3) Logical Operator :-

- are used to perform / evaluate logical expression
- && → AND operator.
- || → OR operator.
- ! → NOT operator.

a) AND operator (&&) :- gives correct o/p when all the inputs are correct. (True value)

boolean a = True

b = True

c = True / False

a && b && c

T && T && T → T

a && b && c

T && T && F → F

a	b	AND
F	F	F
F	T	F
T	F	F
T	T	T

(b) OR operator

↳ gives true o/p when any of the I/p is true

(atleast one is true)

a = T
b = f
c = f

a || b || c
↓
T || F || F
⇒ T

a = f
b = f
c = f

a || b || c
↓
F || F || F
⇒ F

a	b	OR
F	F	F
F	T	T
T	F	T
T	T	T

(c) NOT operator

↳ I/P → F → O/P → T

↳ I/P → T → O/P → F

a	NOT
F	T
T	F

a = False → !a = True

a = True → !a = False

AND Operator

```
boolean a = true;
boolean b = true;
boolean c = true;

System.out.println(a&&b&&c);
```

Your Output

```
true
```

```
boolean a = true;
boolean b = false;
boolean c = true;

System.out.println(a&&b&&c);
```

Your Output

```
false
```

```
// System.out.print( " " );

boolean a = true;
boolean b = false;
boolean c = true;

System.out.println(a||b||c);
```

Your Output

```
true
```

```
boolean a = false;
boolean b = false;
boolean c = false;

System.out.println(a||b||c);
```

Your Output

```
false
```

NOT operator

```
boolean a = false;
boolean b = false;
boolean c = false;

System.out.println(!a);
```

Your Output

```
true
```

```
boolean a = true;
boolean b = false;
boolean c = false;

System.out.println(!a);
```

Your Output

```
false
```

(4) Comparison Operator

↳ are used to do some comparison

↳ >, <, >=, <=, !=, ==

a = 5

b = 10

= → assignment operator

== → equal operator

a != b

↙ ↘

true false

a > b → greater than.

a < b → less than

a >= b → greater than / equal to

a = 10

b = 10

a == b → true

↳ false

a <= b → less than equal to.

a != b → not equal

```java
int a = 10, b = 15;

System.out.println(a>b);
System.out.println(a<b);
System.out.println(a == b);
System.out.println(a != b);
System.out.println(a>=b);
System.out.println(a<=b);
```

→ F
→ T
→ F
→ T
→ F
→ T

Your Output

```
false
true
false
true
false
true
```

→ Sunday → 10 – 12 – morning

→ Google form → for attendance

→ Github