

DSA with Java

First Video week(1st)

1. first java code

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

- a. psvm :- Public static void main
- b. sout :- System.out.Println()

2. Keyword : 100 + keyword in java.

3. Datatype :

3. Type conversion :-

- a. explicit conversion
- b. Implicit conversion

4. Comments :

- a. single line comments :- //
- b. MultiLine comments :- /* */

Second Video week(1st)

1. Binary number system.

- a. decimal to Binary.
- b. Binary to decimal.

1Byte -> 8 bits

MSB -> Most Significance Bits

LSB -> Least Significance Bits

- a. Binary addition.
- b. Binary Subtraction.

2. Operators in java

Types of Operators in Java

- 1. Arithmetic Operators
- 2. Assignment Operators
- 3. Relational Operators
- 4. Logical Operators
- 5. Unary Operators
- 6. Bitwise Operators

- a. Arithmetic Operator.

1. Arithmetic Operators

Operator	Operation
<code>+</code>	Addition
<code>-</code>	Subtraction
<code>*</code>	Multiplication
<code>/</code>	Division
<code>%</code>	Modulo Operation (Remainder after division)

B. Assignment operator

2. Assignment Operators

Operator	Example	Equivalent to
<code>=</code> ↙	↪ <code>a = b;</code>	<code>a = b;</code>
<code>+=</code>	→ <code>a += b;</code>	<code>a = a + b;</code>
<code>-=</code>	→ <code>a -= b;</code>	<code>a = a - b;</code>
<code>*=</code>	→ <code>a *= b;</code>	<code>a = a * b;</code>
<code>/=</code>	→ <code>a /= b;</code>	<code>a = a / b;</code>
<code>%=</code>	→ <code>a %= b;</code>	<code>a = a % b;</code>

C. Relational operator.

Relational operator always Return bool value

3. Relational Operators → always return boolean value ↳ true/false		
Operator	Description	Example
<u>==</u>	Is Equal To	3 == 5 returns false
!=	Not Equal To	3 != 5 returns true
>	Greater Than	3 > 5 returns false
<	Less Than	3 < 5 returns true
>=	Greater Than or Equal To	3 >= 5 returns false
<=	Less Than or Equal To	3 <= 5 returns true

D. Logical operator :

4. Logical Operators		
Operator	Example	Meaning
→ && (Logical AND)	expression1 && expression2	→ true only if both expression1 and expression2 are true
→ (Logical OR)	→ expression1 → expression2	→ true if either expression1 or expression2 is true
→ ! (Logical NOT)	→ !expression	→ true if expression is false and vice versa

E. Bitwise operator :

Read in future__

5. Bitwise Operators

Operator	Description
<code>~</code>	Bitwise Complement
<code><<</code>	Left Shift
<code>>></code>	Right Shift
<code>>>></code>	Unsigned Right Shift
<code>&</code>	Bitwise AND
<code>^</code>	Bitwise exclusive OR

5 → 101 >>
 <<
 &
 1

Bitwise operation
↳ Bit manipulation

3. Take Input from user.

```
/*taking input as string*/
Scanner sc = new Scanner(System.in);
System.out.println("Enter your age");
int age = sc.nextInt();
System.out.println("My age is : " + age);

System.out.println("Enter your float no.");
float myFloatNo = sc.nextFloat();
System.out.println("my float no is : " + myFloatNo);

System.out.println("Enter Your address : ");
String myString = sc.next();
System.out.println("My first time string is : " + myString);

sc.close();
```

Various Input Types using Scanner

↓ ↓ ↓ ↓

We can use nextLong(), nextFloat(), nextDouble(), and next() methods to get long, float, double, and string input respectively from the user.

Note: It is recommended to close the scanner object once the input is taken using the close() method