variables → container

It stores data temporarily.

every variable has a type
- integer :) 1, 5, 7, 9
- float :) 1.2, 98.9, 5.0009
- String :) name, message

DATA TYPES

<datatype> <variable_name>

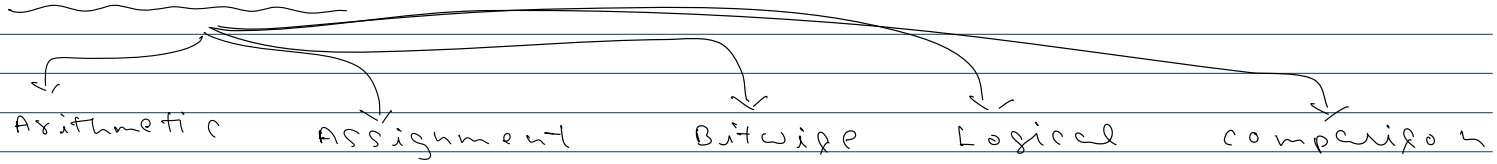
```
class Test{
    public static void main(String[] args){
        // code
        System.out.println("Hello Akarsh");
        System.out.print("Bye Akarsh");
        System.out.println("Okay Akarsh");

        // variable declaration
//          int a;
//          float b;
//          String name;

//          System.out.println(a);
//          System.out.println(b);
//          System.out.println(name);

        // variable declaration + intialisation
//          int a = 5;
//          float b = 2.78f;
//          String name = "Akarsh";

//          System.out.println(a);
//          System.out.println(b);
//          System.out.println(name);

        // variable declaration and variable assignment
        int a;
        float b;
        String name;

        a = 5;

        System.out.println(a);
        // System.out.println(b);
        // System.out.println(name);

        a = 100;
        System.out.println(a);

    }
}
```

variable naming convention

variable naming convention

starting char → - $, or any alphabet
cannot use reserved keywords.

Java Operators



Arithmetic Operators

They are used to perform mathematical operations.

- + add
- - sub
- * multiply
- / divide
- % modulus :- remainder

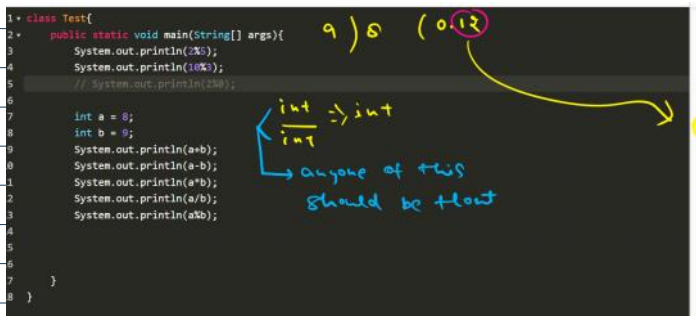
10 % 3 = 1

```
   3 ) 10 ( 3
     - 9
     ---
       1
```

2 % 5 = 2

```
   5 ) 2 (
```

10 6 % 0 : → Exception




```
class Test{
    public static void main(String[] args){
        System.out.println(2%5);
        System.out.println(10%3);
        // System.out.println(2%0);

        int a = 8;
```

```
        int b = 9;

        a = 90;
        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(5/2.0f);

        System.out.println(4/2.0f);

        System.out.println(2.0f);


    }
}
```

Assignment operators

They are used to assign values to variables.

```
a = 9 ;
```

↳ assignment

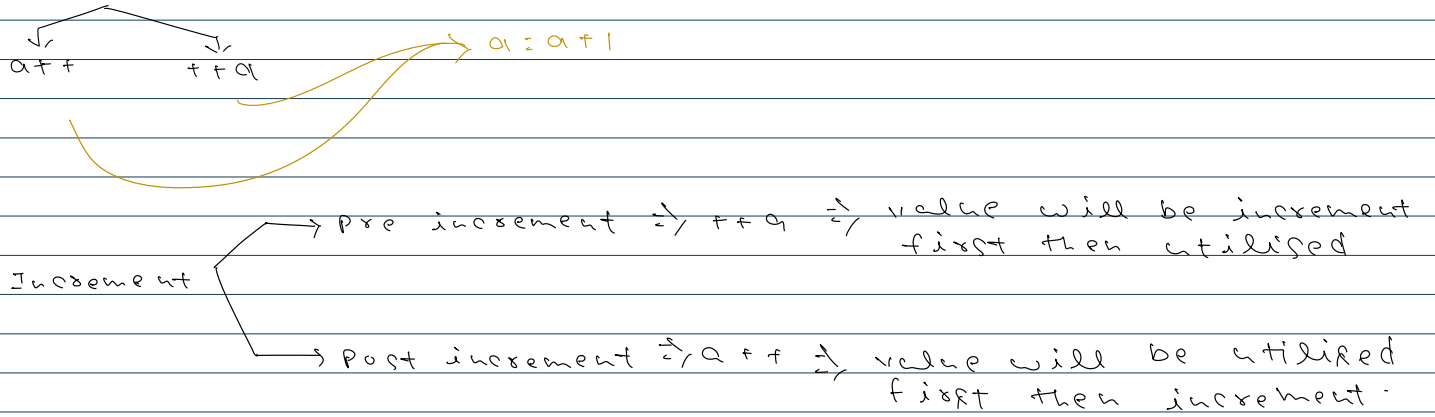
```
a = a + 1   =>   a += 1 ;
b = b - 1   =>   b -= 1 ;
a = a * 5   =>   a *= 5 ;
```

```
class Test{
    public static void main(String[] args){
        int a = 6;

        a = 7+1;

        System.out.println(a);

        int b = 7;

        a = a + b;
        System.out.println(a);

        a = a + 1;
        System.out.println(a);

        a += 1;
        System.out.println(a);

        a -= 5;
```

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


    }
}
```

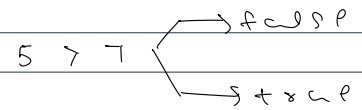
Increment / Decrement Operator

Increment → it will incr the value by 1.




```
class Test{
    public static void main(String[] args){
        int a = 6;

        a++; // a = a + 1;

        System.out.println(a); // a = 7

        System.out.println(a++);

        System.out.println(a); // a = 8

        System.out.println(++a); // a = 9


        int x = 8;
        int c = x++ - ++x;
        System.out.println(c);

        System.out.println(x);
        c = x++ - ++x + --x + x--;
        System.out.println(c);

        System.out.println(x);
        // c = x++ - ( ++x + --x) + x--;
        // System.out.println(c);

        c = x++ - (x + x) + x--;
        System.out.println(c);

    }
}
```

Comparison Operators



equal to ==
not equal to !=
greater than >
less than <
greater than equal to >=
less than equal to <=

```
class Test{
    public static void main(String[] args){
        int a = 5;
        int b = 7;

        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);

    }
}
```