

Basic JAVA Programme: ① → જાણવા જાણુ Print કરા

public class

જે નામ દિધે તે નામ
File નામો Save કરાવે શરૂ
જે નામ દિધે Programme
દર્શાવે

{ ① } of ②

* Exam ને જે નામ
દર્શાવે તે નામ File
Save કરાવે

public static void main (String[] args) { ②

System.out.println ("Hello world");

} ①

} ②

JAVA Script :

console.log ("Hello World!");

* Comment kaise likhenge?

① `/* */` → ① Multiple comment

② `//` → ② Single line comment.

* Variable : → variable type = Value ;

	Name	Example	Example with code
①	int	without decimal 123 to (-123)	<code>int x = 5;</code> <code>print:</code> <code>System.out.println("x");</code>
②	float	with decimal 19.99 to (-19.99)	<code>float x = 5.99f;</code> <code>double x = 5.99d;</code>
③	char	A, B, c, a, b, c	<code>char ch = 'D';</code>
④	string	"Hellow"	<code>String text = "Hellow";</code>
⑤	Boolean	true/false	<code>boolean num = true;</code>

Note:

- 1) Print કરવા માટે જો $(+ something)$ હોય, તો તેના પાછળ add કરો.
- 2) All JAVA variable should have a unique name which will call the term named "identifier".

JAVA Type Casting: ① Widening Casting
② Narrowing casting

↓
is a way to convert a variable from one data type to another data type:

* Data જોડે લખવામાં આવેલું Print/Result નો
જો આપણે જાણીએ તો તે સરળતાથી સમજી શકાય છે.

①

byte
↓
short
↓
char
↓
int
↓
long
↓
float
↓
double

②

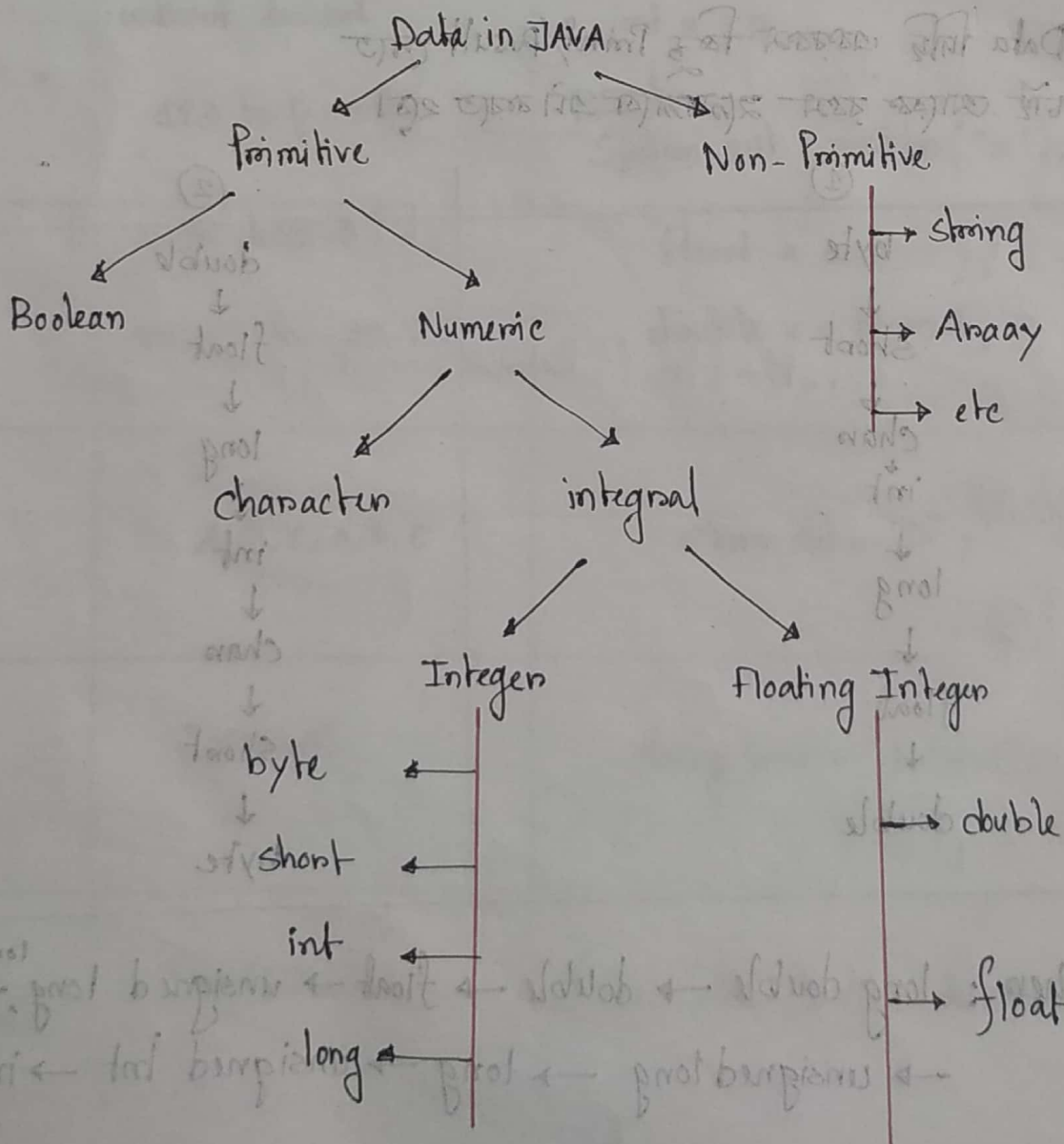
double
↓
float
↓
long
↓
int
↓
char
↓
short
↓
byte

Another: long double → double → float → unsigned long → long long
→ unsigned long → long → unsigned int → int.

Example: `int i = 17;` `char c = 'A';` `int sum;`

`sum = i + c;` `System.out.println("Sum");`

memory to store data



Something add Programme: → (2)

```
import java.util. Scanner;
```

```
public class Something add {  
    public static void main (String [] arg) {
```

```
        int x;  
        int y;  
        int Sum;
```

```
        Scanner sc = new Scanner (System.in);
```

```
        System.out.println ("Enter the number x");
```

```
        x = sc.nextInt();
```

```
        System.out.println ("Enter the number y");
```

```
        y = sc.nextInt();
```

```
        sc.close();
```

```
        sum = x+y;
```

```
        System.out.println (+sum);
```

```
    }
```

```
}
```

JAVA Operator: → JAVA divides the operators into the following group:

- ① Arithmetic Operator
- ② Assignment Operators
- ③ Comparison Operators
- ④ Logical Operators
- ⑤ Bitwise Operators

① Arithmetic Operator → $(+, -, *, /, \%, ++, --)$
↑ Modulus = decrement
↓ increment

② Assignment Operators →

1) $=$	5) $/=$	
2) $+=$	6) $\%=$	9) $>>=$
3) $-=$	7) $ =$	10) $<<=$
4) $*=$	8) $\wedge=$	

↪ Operator ke value ko assign krta hai then us value ka output deta hai.

Ex: `int x = 10` output → (15)

`x += 5`

↪ $x = x + 5$

$10 + 5$

15

- ③ Comparison Operators →
- 1) $==$
 - 2) $!=$ → Not equal to
 - 3) $>$
 - 4) $<$
 - 5) $>=$
 - 6) $<=$ → equal to
 - 7) $===$

- ④ Logical Operators →
- 1) $\&\&$ → Logical AND
 - 2) $\|\|$ → Logical OR
 - 3) $!$ → Logical NOT
- Example: $if (condition) \{ \}$
 If condition is true, then output is printed.
 Example: $if (a > b) \{ \}$
 If $a > b$ is true, then output is printed.

- ⑤ Bitwise Operators →
- 1) $\&$ → Bitwise AND
 - 2) $\|$ → Bitwise OR
 - 3) \oplus → Bitwise XOR
 - 4) \sim → Bitwise NOT
 - 5) \ll → Bitwise Left Shift
 - 6) \gg → Bitwise Right Shift

Example: $a \& b$

Example: $a \ll 2$

Example: $a \gg 2$

JAVA String :

double quotes \leftarrow String \rightarrow are used for storing text
- use કરા રચ

Ex: String greeting = "Hello"

Ex: "abc" + 3 + 5

"abc3" + 5

"abc35"

String length :

string ના અક્ષર કે અંકો કે letter આદે કે Identify કરાત રહે Just Print કરા રચ

Ex:

`+(greeting).length()`

\downarrow
રજત આદે યેટ પાત્ર

String Methode :

① to Upper class \rightarrow Hello

② to Lower class \rightarrow hellow

Ex: `greeting.toLowerCase()`

Finding character in a string : \rightarrow `-indexOf()`

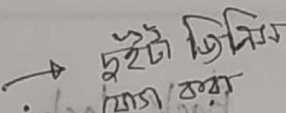
① greeting = Hello

\rightarrow એ Hello (એક H એ અક્ષર ન હોત તો)

Code:

```
String a = 'Hellow'
System.out.println("String" + a);
int index = a.indexOf
System.out.print("a" is index %d \n", index);
```

```
}
```

String Concatenation : 

- ① System.out.print (first Name + " " + Lastname);
- ② system.out print (first Name.concat (lastname));

1st Space जोड़ें



JAVA counts positions from zero → Like : 0, 1, 2, 3, 4

'0' is the first position in a string

'1' is " second

'2' " third

Special Characters :

'Hello' = a print

Escape character	Result	Description	Code
'\'	''	single Quote	We are the "Viking" -②
'\"'	""	double Quote	It's alright -①
'\\'	'\'	Back slash	This sign \ is called Backslash.

☐ 6 other's escape sequence Valid in JAVA :

Code	Result
'\n'	New line
'\r'	Carriage Return
'\t'	Tab
'\b'	Backspace
'\f'	from feed

JAVA Math :

- 1) max
- 2) min
- 3) sqrt
- 4) random \rightarrow (0 থেকে 1 এর মাঝামাঝি (Random Number))
- 5) abs \rightarrow পরমাণ
- 6) power $\rightarrow 2^{\square}$

Ex: `System.out.print (Math. C (value))`

Ex:

`Math. pow (2,3)`

output $2^3 = 8$

JAVA Boolean : \rightarrow =, >, <

1) Yes/No

2) True/False

3) ON/OFF

boolean `5 > 2 = true`

`System.out.println (is JAVA Fun)`

output: true

Induction : প্রশ্নের ডিট (প্রশ্নের প্রশ্ন)

JAVA Condition Statement :

Conditional Statement 27/8 & 28/8 : 1) if 2) else 3) else if 4) switch

if statement Format →

```
if ( )  
{ .....  
}
```

else statement →

```
if ( )  
{ .....  
else { .....  
}
```

JAVA Boolean :

```
{ = }  
{ > }
```

else if statement →

if () → Condition ①

```
{  
}
```

else if () → Condition ②

```
{  
}
```

else

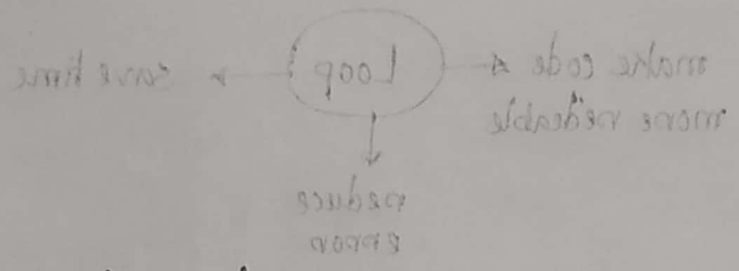
```
{  
}
```


JAVA Loop statements :

short hand If..else :

- 2) Do-while loop
- 3) for loop

Exits : Break / Continue statement :



JAVA Switch statements :

for loop with while and do-while

1) initialization

2) condition

3) increment / decrement

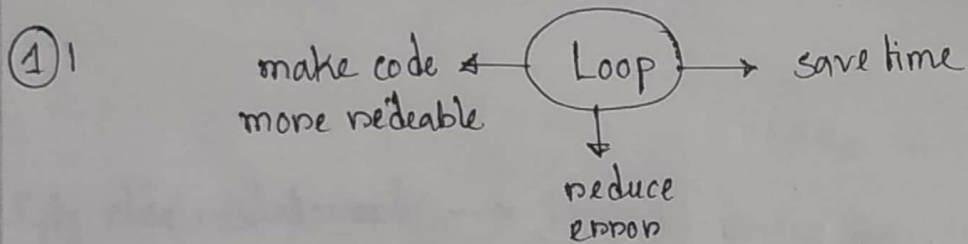
EX: $i = 1$, $i < 5$, $i++$
 initialization condition increment

output : 1 2 3 4 5

JAVA Loop statement :

JAVA loop statements ৩ প্রকার : 1) while loop
2) Do-while loop
3) For loop

* Extra : Break/Continue statement :



☐ while Loop করার জন্য আমরা ৩টি কাজ করা লাগে

① initialization

② Condition

③ increment/decrement

Ex: $(i = 1 , i \leq 5 , i ++)$ → increment/decrement
 ↓ ↓
 initialization condition

output : 1 2 3 4 5

1) while loop :

```
      initialization
while ( → condition
{
    increment
}
```

2) Do-while loop :

```
      initialization
do {
    print increment
}
while (condition)
```

4) For-loop :

```
      ↓      ↓      ↓
for ( initialization, condition, increment )
```

```
{
}
}
```

5) Nested loop : एक Loop में दूसरा Loop / एक Loop use करने के लिए Nested loop / Loop Nesting.

Break/Continue :

- ① break For loop এর ভিতরে যখন goal condition (শর্ত) পূর্ণ হয় তখন loop break করে।
- ② continue Do-while loop এর ভিতরে যখন goal condition (শর্ত) পূর্ণ হয় তখন loop continue করে।

Break	Continue
<pre>for () { if (==) { break; } }</pre>	<pre>for () { if (==) { continue; } }</pre>

Goal condition / Goal condition is not met : Goal condition is met (2) Nested loop : Goal condition is not met / Goal condition is met

JAVA Array Statement:

Array : ଓଡ଼ିଆରେ Variable ଅଟେ ଏବଂ ଏହା Variable ଓ ଏହାର ସୂଚକ, ଓ
Second Bracket/corli Bracket ଓ ଏହାର ସୂଚକ ଅଟେ ।

Ex: `int array [] = {1, 2, 3, 4, 5}`
`string array [] = {car, truck, bus}`

Array index start with 0 : [0] is the first element
[1] is the second element
[2] is the third element

: { {1, 2, 3}, {4, 5, 6} } = array [] []

Array Particular (ଆମେ ଏହା କିଛି କିଛି କରୁ)

Ex: `String [] cars = {"Volvo", "BMW", "Ford"};`
`System.out.println(cars[0]);`

change array Element:

Ex: `String [] cars = {"Volvo", "BMW"};`
`cars [0] = opel`
`System.out.println(cars[0]);`

Array length:

Ex: `String cars = { " ", " ", " " }`

`System.out.println(cars.length)`

`{ car, truck, bus } = cars`

Multi dimensional Array:

Ex: `int [][] myNumbers = { {1,2,3}, {4,5,6} };`

Ex: `int [][] myNumbers = { {1,2,3}, {5,6,7} };`

`int x = myNumbers[1][2];`

`System.out.println(x);`

change array element:

Ex: `String[] cars = { "Volvo", "BMW" };`

`cars[0] = "Volvo";`

`System.out.println(cars[0]);`

Loop Through an array :

① `String[] cars = { "Volvo", "car", "Ford" };`

`for (int i = 0; i < cars.length; i++)`

`{`
`System.out.println(cars[i]);`

`}`

② `String[] cars = { "Volvo", "car", "Ford" }`

`for (String i : cars)`

→ For Each

`{`

`System.out.println(i);`

`}`

Array

multiple loop with for :

```
Ex: { int [][] myNumbers = { {1,2,3}, {4,5,6} }  
    for (int i=0; i < myNumbers.length; ++i)  
        for (int j=0; j < myNumbers[i].length; ++j)  
            System.out.println(myNumbers[i][j]);  
    }
```

```
{ "boat", "car", "van" } = cars [ ] print
```

```
for (String i : cars) print i;
```

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


28 Different language in name :

① ALGOL

② ASPECT

APPLESCRIPT

ASSEMBLY Language

Bash (Unix shell)

Basic

C

C++

C#

Caml (Ocaml)

Clojure (clojurescript)

Cobol

Coffee script

Dart

DBase (Foxpro)

Delphi (object pascal)

eiffel

erlang

elixir

F#

Fortran

go

groovy (ruby)

haskell

IBM PRG

JAVA

JAVASCRIPT

Lisp

Logo

Lua

Machine code

mathematica (wolfram language)

matlab

mL

NODE.JS

OBJECTIVE - C

PASCAL

PERL

PHP

POWERShell

Python

R

RPGI

Ruby

Rust

Scala

Scheme

Scratch

Self

Swift

TCL

Typescript