

First Program "Hello world":-

```
class HelloWorld {  
    public static void main (String[] args) {  
        System.out.println("Hello, world");  
    }  
}
```

1) Main method:-

public :- So that JVM can execute the method from anywhere. (Access Modifier)

Static :- Main method is to be called without object

Void :- The main method doesn't return anything

main() :- Name Configured in the JVM.

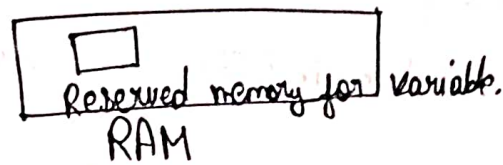
String[] :- The main method accepts a single argument :-
an array of elements of type string.

Data Types:-

Type	Default	Size (bit)	Range.
boolean	false	1	True, False
byte	0	8	-128 to 127
Char	\u0000	16	0 to 255
Short	0	16	-32,768 to 32,767
int	0	32	-2,147,483,648 to 2,147,483,647
long	0	64	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
float	0.0	32	upto 7 decimal digits.
double	0.0	64	upto 16 decimal digits

Variables in Java.

Type \swarrow
int count; \searrow name.



Types of Variables:-

- 1) Local variables
- 2) Instance variables
- 3) Static variables

Code:- { Data Types }

byte num = 25;

Short numb = 1256;

Char b = '#';

int a = 25648;

long c = 258964153;

float f = 2.14589;

double e = 2.1589632478;

Sysout (num);

Sysout (numb);

//

"

"

"

"

④

25

1256

#

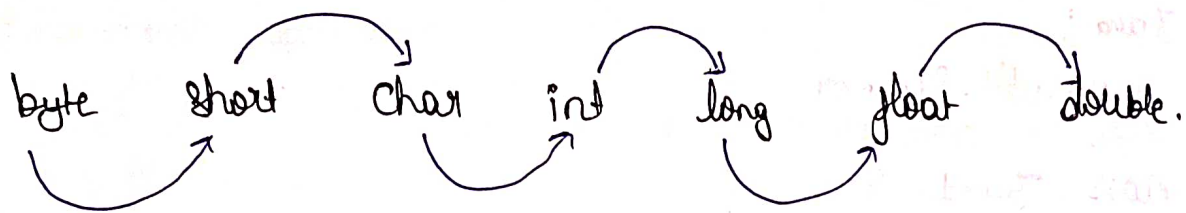
25648

2589 64153

2.14589

2.1589632478

#4 Type Casting & Taking user Input:



#1 Automatic typecasting:-

If we follow the above series from left to right automatic typecasting takes place.

```
int x = 25;
```

```
int y = x;  
System.out.println(y);
```

① 25.0

#2 Manual typecasting:-

```
double myDouble = 2.826;
```

```
int myInt = myDouble;
```

→ Error: Type mismatch: cannot convert from double to int.

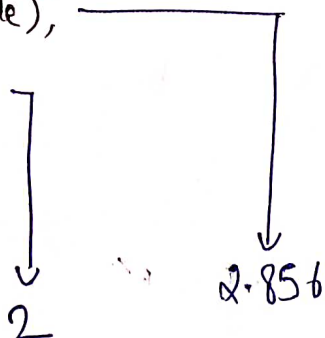
So,

```
double myDouble = 2.826;
```

```
int myInt = (int) myDouble;
```

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

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



★ Always print floor value.

Taking input:-

```
Package  
Class Java;  
import java.util.Scanner;
```

```
public class Java1 {
```

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

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

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

```
System.out(x);
```

② 25

25

Operators in Java:-

1) Arithmetic Operators:-

It perform simple arithmetic operations on primitive data types.

- * : Multiplication
- / : Division
- % : Modulo
(Returns Remainder)
- + : Addition
- - : Subtraction.

int a;

int b;

- ++ } Increment operator
- -- } Decrement operator

#1. int num = 5;

num++;

System.out.println(num);

Output: 6

int num = 25;

~~int num;~~

num--;

System.out.println(num);

Output: 24

Arithmetic

+ -

Unary

+ -

* / %

++ --

both

a

a + b

2.) Unary operators:-

```
int a = 10
```

```
a = -a
```

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

① -10

• Not operator (!)

3.) Assignment operators :-

used to assign values to a variable

a.) `int num;` ← Initialise

`num = 25;` ← assign.

b.) `" += "`

It add the current value of the variable on left to the value on the right.

```
int a = 12;
```

```
a += 5;
```

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

① 17

int a = 5

a *= 5
System.out.println(a);
25

{ a * 5 } = { 5 * 5 }

d. ~~do~~ "a = 25"

int a;

d.) "a = 25"

int a = 25

a -= 5;

System.out.println(a);

20

e.) ~~do~~ "a /= 7"

int a = 49

a /= 7;

System.out.println(a);

7

8.) Modulo operator :-

```
int a = 11;
```

```
a %= 3;
```

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

① 2

Relational Operator:-

1.) Equal to operator ($==$):-

```
int a = 25;
```

```
int b = 25;
```

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

① True.

2.) Not Equal to operator ($!=$):-

```
int a = 25;
```

```
int b = 25;
```

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

① False.

3.) Greater than operator ($>$)

4.) Smaller than operator ($<$)

5.) Greater than or equal to ($>=$)

6.) Smaller than or equal to ($<=$)

Logical operators :-

1.) 'Logical AND (&):-

```
int a = 5;
```

```
int b = 6;
```

```
int c = a & b;
```

```
System.out (c);
```

② 4

2.) 'Logical OR' (||) :-

 = a || b;

```
System.out (c);
```

15

Ternary operator :-

```
int a = 5;
```

```
int b = 65;
```

```
int result = 0;
```

```
result = a > b ? a : b;
```

```
System.out (result);
```

② 65

Also,

```
int a = 65, b = 101, c = 3;
```

```
int result = 0;
```

```
result = a > b ? a : b > c ? b : c;
```

② 101

If else, else if :-

If :-

```
int age = 11;  
if (age >= 18) {  
    System.out.println("you can vote now");  
}
```

① _____

2) If else class :-

```
if (condition)  
{  
    // Executes this block if  
    // condition is true  
}
```

}

else

```
{ // Executes this block if  
  // condition is false  
}
```

}

```
int no. = 5;  
if (no % 2 == 0) {  
    System.out.println("He is good");  
}
```

```
else  
{ System.out.println("Boo")  
}
```

Else-If:-

```
int number = sc.nextInt();
```

```
if (number <= 10) {
```

```
    System.out ("no. is less than or equal to 10");
```

```
} else if (number > 10 && number <= 20) {
```

```
    System.out ("no. is greater than 10's less than 20");
```

```
} else if (number > 20 && number <= 30) {
```

```
    System.out ("no is greater than 20's less than 30");
```

```
} else {
```

```
    System.out ("no is greater than 30");
```

8. Nested If-else:-

Logical operators:-

&&

And

||

OR

!

NOT

1.) int a = 15;

if (a >= 1 && number <= 100) {

System.out.println("number is in the range");

}

ⓐ number is in the range.

2.) if number = 15;

if (number >= 1 && number <= 100) {

System.out.println("number is in the range");

z) Same output-

|| → {checks both?}

Logical 'OR' :-

int grade = 10;

if (grade == 10 || grade == 12) {

System.out.println("you can give boards now");

Logical 'Not' :-

```
int grade = 11;
```

```
if (! (grade == 10 || grade == 12)) {
```

```
    System.out.println("you can't give boards now");
```

```
}
```

Nested If - Else :-

Max of 3 numbers:

```
int a = 36, b = 480, c = 599;
```

```
if (a > b) {
```

```
    if (b > c) {
```

```
        result = a;
```

```
    } else {
```

```
        result = c;
```

```
    }
```

```
    } else {
```

```
        if (b > c) {
```

```
            result = b;
```

```
        } else {
```

```
            result = c;
```

```
        }
```

```
    } System.out.println(result);
```

```
}
```

```
}
```

Nested Ternary operator:-

```
int a, b, c;
```

```
int result = 0;
```

```
result = a > b ? a > c ? a : c : b > c ? b : c;
```

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

#

```
result = a > b ? a : b > c ? b : c;
```

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

Switch Case:-

int dayofweek = 7.

Switch(dayofweek) {

Case 1:

Sysout ("At work");

break;

Case 2:

Sysout ("At home");

break

default:

Sysout ("I don't know what the date is");

}

Q Rating:-

int rating = 4; int

Switch (rating) {

Case 1:

Sysout ("At work");

int rating = sc.nextInt();

switch (rating) {

Case 1:

Case 2:

System.out.println("Bad");

break;

Case 3:

System.out.println("Average");

break;

Case 4:

Case 5:

System.out.println("Good");

break;

default: ?

System.out.println("Ratings are given out of 5");

Loops :-

3 Types

1.) For loop

2.) while loop

3.) do-while loops.

#1 For loop:-

```
1) for (int i = 0 ; i < 100 ; i = i + 1) {  
    System.out.println ("Hello World");
```

```
}
```

① Hello world

|
|

|

— 100 times.

```
2.) for (int i = 0 ; i < 100 ; i + 1) {  
    System.out.println (i) :
```

```
}
```

|

|

|

100

3.) Sum of 1st 5 numbers :-

```
int sum = 0;  
int n = 5;  
for (int i = 1; i <= n; i++) {  
    sum = sum + i;  
}  
System.out.println(sum);
```

Q 15

Sum of no. from 25 to 50 :-

```
int sum = 0;  
int n = 50;  
for (int i = 25; i <= n; i++) {  
    sum = sum + i;  
}  
System.out.println(sum);
```


4. Table

```
int table = Sc.nextInt();  
for (int i = 1 ; i <= 10 ; i++) {  
    System.out.println(table * i);  
}
```

}

}

5.) factorial;

```
int n = Sc.nextInt();  
int factorial = 1;  
for (int i = 1 ; i <= n , i++) {  
    factorial = factorial * i;  
}
```

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

}

```
② for (int i = n ; i >= 1 ; i--) {  
    factorial = factorial * i;  
    System.out.println(factorial);  
}
```

Fibonacci :-

Scanner class.

```
int n = sc.nextInt();
```

```
int a = 0;
```

```
int b = 1;
```

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

```
System.out.print(b);
```

```
for (int i = 0; i < n-2; i++) {
```

```
    int c = a+b;
```

```
    System.out.print(c);
```

```
    a = b;
```

```
    b = c;
```

Power:-

```
int base = sc.nextInt();
```

```
int power = sc.nextInt();
```

```
int result = 1;
```

```
for (int i = 0; i < power; i++) {
```

```
    result *= base;
```

```
}
```

```
System.out.print(result);
```

```
}
```

```
}
```

#12

Break statement:-

```
for (int i = 0; i <= 100; i++) {  
    System.out.println(i);  
}
```

=> 1 - - - - 100.

But,

```
for (int i = 1; i <= 100; i++) {  
    if (i == 35) {  
        break;  
    }  
}
```

3
System.out.println(i);

=> 1 - - - - 34

#

```
for (;;) {
```

```
    int n = Sc.nextInt();
```

```
    if (n < 0)
```

```
        break;
```

```
}
```

① 1, 2, 33, 4, -3 → out from loop after -ve

++ continue :-

Skips a point and then Continue

```
for (int i = 1 ; i <= 100 ; i++) {
```

```
    if (i >= 40 && i <= 50 ) {
```

```
        Continue;
```

```
    }
```

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

```
}
```

```
}
```

2.) for (int i = 1 ; i <= 15 ; i++) {

```
    if (i == 10 || i == 13) {
```

```
        Continue;
```

```
    }
```

```
    Sysout(i);
```

Nested for loops:-

```
for (int j = 1; j <= 5; j++) { // No of Times Code run
    for (int i = 1; i <= 10; i++) { // Content that runs.
        System.out.print(i + " ");
    }
    System.out.println();
}
```

Patterns:-

①

Scanner class;

int n = sc.nextInt();

```
for (int i = 1; i <= n; i++) {
    for (int j = 1; j <= n; j++) {
        System.out.print("* - ");
    }
}
```

3

System.out.println();

3

Input - 3

Output -

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


② Scanner class:-

```
int n = sc.nextInt();
```

```
for (int i = 1; i <= n; i++) {
```

```
for (int j = 1; j <= i; j++) {
```

```
    System.out.print(j);
```

```
    }
```

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

③ Input - 3

Output -

x

x x

x x x

③ Pattern 3:-

```
int n = sc.nextInt();
```

```
for (int i = 1; i <= n; i++) {
```

```
for (int j = 1; j <= n - i + 1; j++) {
```

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

```
    }
```

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

3.

④ 3

* * *

* *

*

#15 While loops & Do-while loops:-

Q-1 Write a program to find the sum of the digits of a number.

Eg. 532 \rightarrow digit sum = $5+3+2 = \underline{\underline{10}}$.

$$n = 435.$$

$n \% 10$ { Gives us one's place digit }.

$$435 \% 10 = \underline{\underline{5}}$$

$n / 10$ { Tens }

$$435 \div 10 = 43 \uparrow$$

$$43 / 10 = 4.$$

Code:-

```
int n = Sc.nextInt();
```

```
int temp = n;
```

```
int sum = 0;
```

```
while (temp > 0) {
```

```
    int lastDigit = temp % 10;
```

```
    temp
```

```
    temp /= 10;
```

```
    sum += lastDigit;
```

5
System.out.println(sum);

Q-2 Find no. of digits of a number:-

int n = Sc.nextInt();

int noOfDigits = (int) Math.log10(n) + 1;

System.out.println(noOfDigits);

Palindrome no.:-

Write a program to check whether the given number is palindrome or not.

12321 $\xrightarrow{\text{Reverse}}$ 12321 {same}

n = 535
r = 535

$$5 \times 10 + 3 = 53$$

$$53 \times 10 + 5 = 535 = r \text{ {Reversed No.}}$$

n = 471

$$1 \times 10 + 7 = 17$$

$$17 \times 10 + 4 = 174 \longrightarrow$$

Code:-

```
int n = SC.nextInt();
```

```
int temp = n;
```

```
int reversedNum = 0;
```

```
while (temp > 0) {
```

```
    int lastDigit = temp % 10;
```

```
    reversedNum = reversedNum * 10 + lastDigit;
```

```
    temp /= 10;
```

```
}
```

```
if (reversedNum == n) {
```

```
    System.out.println(n + " is palindrome");
```

```
} else {
```

```
    System.out.println(n + " is not palindrome");
```

Do-while loops:-

do {

// Condition .

} while (Condition);

1.) int n = a

do {

System.out.println("N is n");

} while (n != 0);

2.) int n = 0;

do {

n = SC.nextInt();

System.out.println("Number" + n);

} while (n != 1);