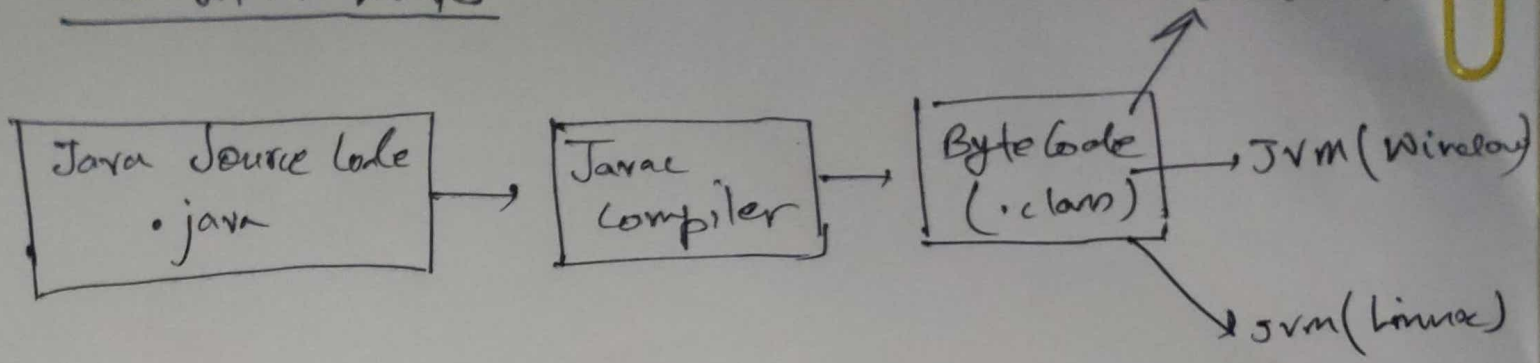
How JAVA works



→ Hello World
→ DT
→ Post/Past Increment
→ Type Casting.

JAVA RI

~~OOPS~~ TOPICS

Hello World

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

- No gap between class name
  - ⇓
  - Camel Case convention
- Also case sensitive
  - System ✓
  - String
  - println
  - System ✗
  - String
  - Println

Eg. rateOfInterest

(lowercase) small — big (uppercase).

Data Types

1. byte 8 bits
2. long 8 bytes
3. int 4 byte
4. short 2 byte
5. double
6. boolean
7. char
8. float

Post / Past Increment

```
b = 45
d = b++
System.out.println(b) = 46
                   (d) = 45
```

```
b = 45
d = ++b
System.out.println(b) = 46
                   (d) = 46
```

TYPE CASTING : Assign a value of one primitive data type to another.

Automatic Type Casting (ATC)

byte → short → char → int → long → float → double

being reverse : Manual Type Casting

Eg.

```
int a = 4
long b = a
print (b)
```

4

following ATC

But if

```
double p = 3.14
int q = p
print (a)
```

→ This line will give error

Reverse of ATC

∴ manual TC is required

Soln:

```
double p = 3.14
int q = (int) p
print(a)
```

3

Scanner function in Java.


```
import java.util.Scanner

public class SimpleInterest{
    public static void main (String[] g)
    { Scanner sc = new Scanner(System.in)

        int principal = 500      = sc.nextInt();
        float rate = 12.5f       = sc.nextFloat();
        int time = 12            = sc.nextInt();

        float SI = principal * rate * float /100;
        System.out.println("the SI in" + ____);
    }
}
```

(Pb) Take input of a string after a Number.

```
{ int num = sc.nextInt();
  sc.nextline();
  String str = sc.nextline();

System.out.println(num);
System.out.println(str);
}.
```

→ Operators 5 types.

1. Airthematic — +, −, ×, /, %, ++, −−
2. Bitwise — NOT, AND, OR, XOR — &=, |=
3. Assignment — =, +=, −=, *=, /=, %=, >>=, <<=,
4. Comparison — :=, !=, >, <, >=, <=
5. Logical — &&, ||, !

NOT
opposite

0	1
1	0

AND
Both 1
=1

0 0	0	0
0 1	0	1
1 0	0	1
1 1	1	0

OR
atleast 1

0 0	0
0 1	1
1 0	1
1 1	1

XOR
only one 1

0 0	0
0 1	1
1 0	1
1 1	0

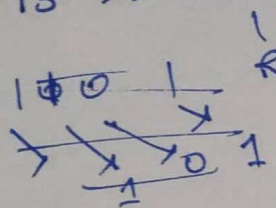
(6)

Right shift
÷2

1101 >>

a = 13

13 >> 1



1 1 0 1

0 1 1 0 = 6

int a = 13

b = 13 >> 1 $\frac{13}{2^1} = 6$

c = 13 >> 2 $\frac{13}{2^2} = 3$

d = 13 >> 3 $\frac{13}{2^3} = 1$

left shift
×2

a = 1 1 0 1 = 13

1 1 0 1 0 ⇒ 26

a = 13

b = 13 << 1 $13 \times 2^1 = 26$

c = 13 << 2 $13 \times 2^2 = 52$

d = 13 << 2 $13 \times 2^3 = 104$

13
8
104

Trick.

1101.0000 ... >> 1 ⇒ 110.0000 → 110 → 6

Right shift by x ⇒ Eliminate last x bits from last.
÷2^x : Dec form.

1101.0000 ... << 1 ⇒ 11010.000 → 26

Left shift by x ⇒ Add x zeros in binary form
×2^x : Dec form

Right shift by x ⇒ /2^x in Decimal. (7)

eliminate last x bits in Binary

Left shift by x ⇒ x 2^x in decimal

append x bits of zeros in binary form

Q. Shortchand (Ternary op)

variable (cond) ? exp1 : exp2.

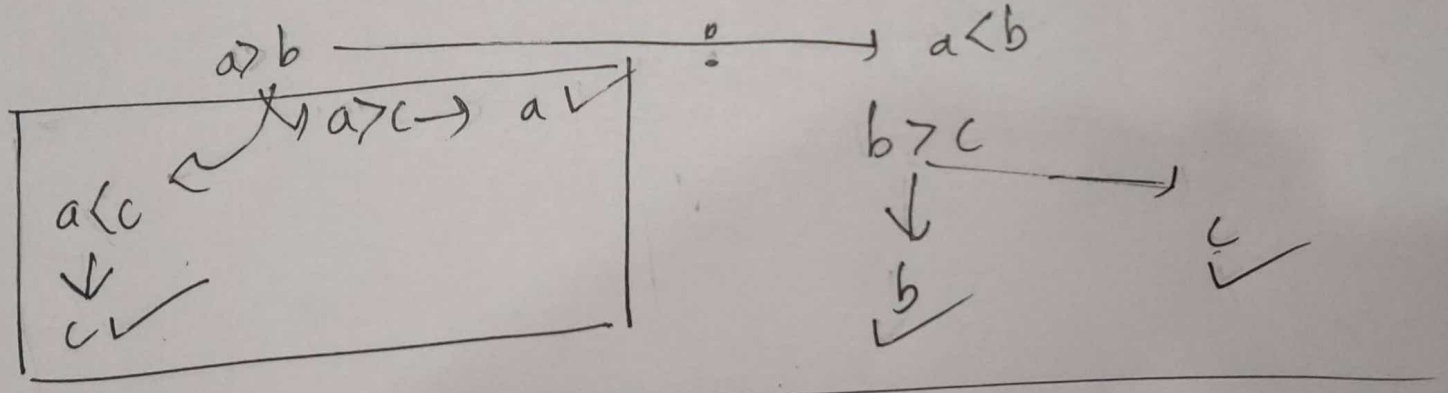
Greatest of three numbers.

→

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

Explanation.

a>b ? a>c ? a : c : b>c ? b : c

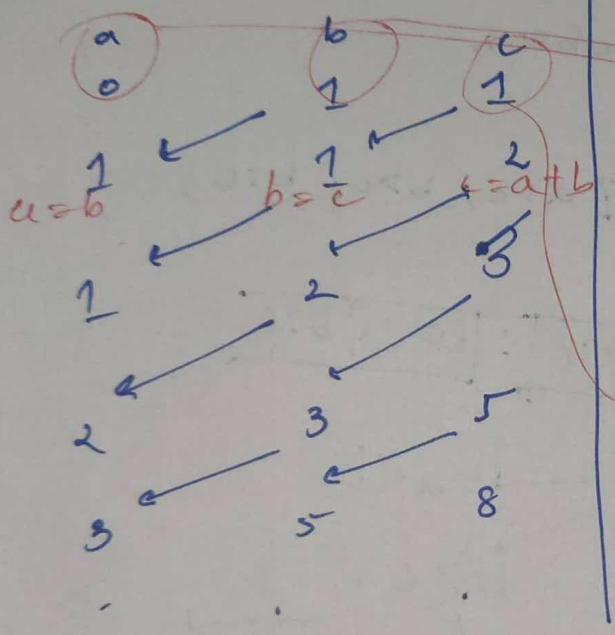


Switch Case

(P6) factorial:-

```
int n = 5                    5!
for(i=n; i>0; i--)
    { product = product * n };
print(product)
```

(P7) fiboraccii.




```
int n = 7.
int a = 0, b = 1
int c, i
for(i=0; i<n-2; i++)
{ c = a+b
  print(c)
  a = b
  b = c
}
```

0 1 1 2 3 5 8 ...

(P8) Prime OR NOT

~~If a num is prime, then its atleast on factor no~~

For any number (n), atleast one of its factor will be present in $(2, \sqrt{n}]$ i.e before $i = \sqrt{n}$ i.e $i*i = n$

```
bool p = ~~false~~ true;
for(i=2; ~~i<=√n~~ i*i<=n; i++)
  { if (n%i == 0)
      { ~~p=true;~~
        p = false;
        break; }
if(p) print("Num in prime");
else print("Num is not prime");
```

(P9)

Series Sum $1+\frac{1}{2}+\frac{1}{3}+\frac{1}{4}+ \cdots \frac{1}{n}$ Ans

Input n o/p = Ans.

```
int num n; sum = 0;
for (i = 1; i <= n; i++)
  { sum = sum + 1/i };
print (sum);
```

(P10) $1-\frac{1}{2}+\frac{1}{3}-\frac{1}{4}+\frac{1}{5}+ \cdots \frac{1}{n}$

o/p = Ans.

I/p = n

```
int n; sum = 0
for(i=1; i<=n; i++)
{ if (i%2 != 0)
    { sum = sum + 1/i; }
  else
    { sum = sum - 1/i; }
}
print (sum);
```

	If ∞ loop or not
for (i = 0; i<10; i++)	N
for (i = 0; ; i++)	Y
for (; ; i++)	N : i not declared, what to t
for (; ;)	Y