

These tokens together form a Statement/instruction → (typed)

3. Operators

1. Arithmetic operators
+ - * / %

2. Relational/Comparison

>, <, >=, <=, !=

3. Logical operator

&&, ||, !

4. Bitwise operators

OR $\wedge, \neg, \sim, \wedge, \vee, \neg\wedge$ Left Shift

Right Shift

XOR $\wedge\vee, \neg\wedge\vee$

5. Ternary operators
?:

4. Literals/Constraints

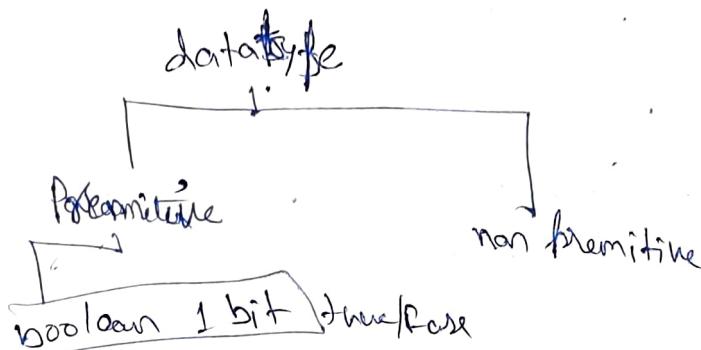
1. Integer
2. Floating point
3. Characters
4. String
5. Boolean

3 Variables →

It is an identifier which retrieves the values at runtime.

e.g. int a = 20;

Variables.



byte 1 byte \rightarrow -128 to 0 to 127

short 2 byte \rightarrow 2^{15} to 0 to $2^{15} - 1$

int 4 byte \rightarrow -2^{31} to 0 to $2^{31} - 1$

long 8 byte \rightarrow -2^{63} to 0 to $2^{63} - 1$

float 4 byte

double 8 byte

char - 2 byte (A) \rightarrow unicode \rightarrow 4 digit hexad.

$$4 \times 4 = 16$$

b - 48

i - 49

A - 65

B - 66

a - 97

! - 32

g - 57

z - 90

z - 122

2D

Identifiers → names for identification purpose
(variables, labels, methods, class)

Key word - Predefined words that has some meaning / functionality

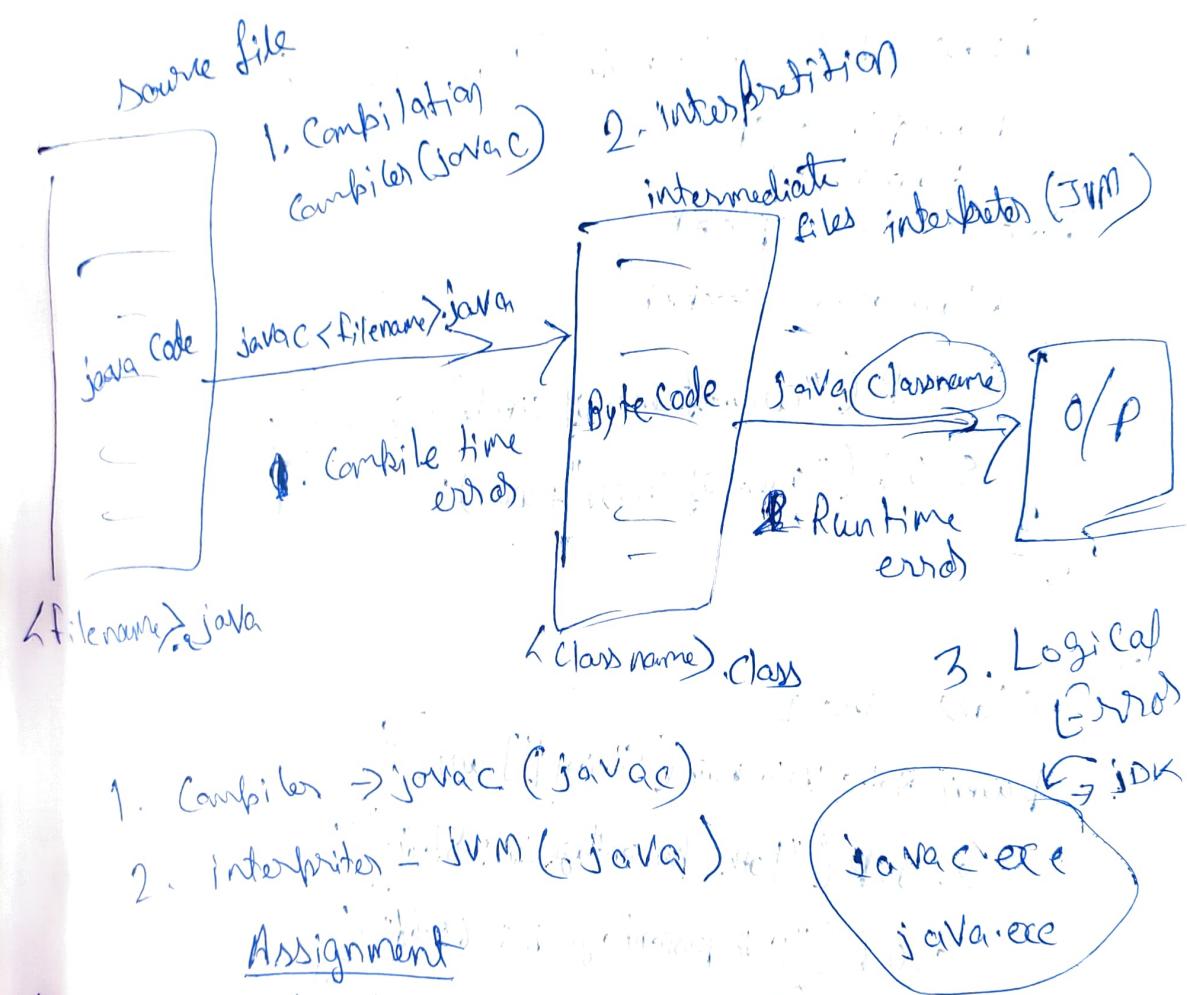
Total 53 - key words in Java.

Operators - are symbols that operates between two operands.

Variables or Values

4) Literals are types of data we specify to a variable.

5) Variables - There are the identifiers which identifies the address of memory location through which we can store, retrieve and modify the value and store in the same memory location.
 (Variable = name to a memory location)



- Q1. Write a Java program to print your name, mobile no & mail id.
- Q2. WAP to print area and circumference of a circle
- Q3. WAP to print 10, 12, degree percentage
- Q4. WAP to initialize two integer Value & performs all the arithmetic operation on those two integers.
- Q5. WAP calculate & print area & perimeter of a rectangle.

1. Class Test ?

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

```
    S.O.P("Name : SAURAV KUMAR JHA");
```

```
    S.O.P("Mobile no : 7992203938");
```

```
    S.O.P("Mail id : SauravJha62@gmail.com");
```

3
3

2. Class Area ?

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

```
    float pie = 3.14f;
```

```
    float r = 5;
```

```
    float Circum = 2 * pie * r;
```

```
    float area = pie * r * r;
```

```
    S.O.P ("Circumference of circle" + Circum);
```

```
    S.O.P ("area of circle" + area);
```

3
3

3. Class Test ?

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

```
    S.O.P ("10th percentage is: 81.6%");
```

```
    S.O.P ("12th percentage is: 63.6%");
```

```
    S.O.P ("Btech percentage is: 76%");
```

3
3

4. Class Abc ?

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

```
    int a = 15, b = 5;
```

```
    S.O.P ("a+b" + a + b);
```

```
    S.O.P ("a-b" + a - b);
```

```
    S.O.P ("a*b" + a * b);
```

```
    S.O.P ("a/b" + a / b);
```

```
    S.O.P ("a%b" + a % b);
```

5. class Rect {

```
int l = 10;  
int b = 5;  
int per = 2 * (l+b);  
int area = l * b;  
System.out.println("Perimeter of rectangle is:" + per);  
System.out.println("Area of rectangle is:" + area);  
}
```

}

Java

util

Scanner

boolean	nextBoolean()
byte	nextByte()
short	nextShort()
int	nextInt()
long	nextLong()
float	nextFloat()
double	nextDouble()
String	next()
String	nextLine() - Read the line

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

```
class Main
```

```
{
```

```
b.S.V.M() {  
    Scanner sc = new Scanner(System.in);  
    S.O.P("Enter the age.");
```

```
    int age = sc.nextInt();  
    Read integer value
```

```
    S.O.P("Entered age is " + age);
```

```
}
```

↳ Capitalize the value

Write a java program to read radius of a circle and compute area.

area = πr^2

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

```
class maincircle {
```

```
    mainmethod
```

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

```
        S.o.P ("Enter the radius");
```

```
        double r = sc.nextDouble();
```

```
        S.o.P ("Radius is "+r);
```

```
        double area = 3.142 * r * r;
```

```
        double cir = 2 * 3.142 * r;
```

```
        S.o.P ("Area is "+area);
```

```
        S.o.P ("Circle is "+cir);
```

- Q. WAP to read the side of a square and print area and perimeter of square.

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

```
        S.o.P ("Enter the side");
```

```
        double s = sc.nextDouble();
```

```
        S.o.P ("Side of square is "+s);
```

```
        double area = s * s;
```

~~double per = 4 * s;~~

```
        double per = 4 * s
```

```
        S.o.P ("Perimeter of square is "+per);
```

```
        S.o.P ("Area of square is "+area);
```

3rd

- Q. WAP to read the length and breadth. Print area and perimeter of rectangle.

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

```
        S.o.P ("Enter length");
```

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

```
        S.o.P ("Enter breadth");
```

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

```
        int area = l * b;
```

~~int per = 2 * (l + b) / 2 * (l + b);~~

S.O.P ("Perimeter of rectangle is " + per);
S.O.P ("Area of rectangle is " + area);

3

- Q. WAS & T read person name and his age and print what will be his age after 5 years. print what is his age.

Scanner sc = new Scanner (System.in)

what is

S.O.P ("What is your name");

String name = sc.nextLine();

How old are you..;

S.O.P ("How old are you..");

int age = sc.nextInt();

int age = age + 5;

S.O.P ("Your name is " + name);

S.O.P ("Your age after 5 years ago will be " + age);

S.O.P ("Your age before 5 years was " + age);

3

- Q. WAS & T calculate the simple interest

Scanner sc = new Scanner (System.in);

S.O.P ("Enter Principle");

int P = sc.nextInt();

S.O.P ("Enter Rate");

int R = sc.nextInt();

S.O.P ("Enter Time ");

int T = sc.nextInt();

int SI = (P * R * T) / 100;

S.O.P ("Simple interest : " + SI);

3

a. WAIPT need two integer value from the user and perform all the arithmetic operation.

```
Scanner sc = new Scanner(System.in);
S.O.P("Enter the value of a");
int a = sc.nextInt();
S.O.P("Enter the value of b");
int b = sc.nextInt();
S.O.P("a+b = " + a+b);
S.O.P("a-b = " + a-b);
S.O.P("a*b = " + a*b);
S.O.P("a/b = " + a/b);
S.O.P("a%b = " + a%b);
```

b. WAIPT read person name and his age from the user and print he is eligible to vote or not.

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

```
Class MainVote
```

```
{ S U M (String arg[]) }
```

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

```
S.O.P("what is your Name?");
```

```
String name = sc.next();
```

```
S.O.P("How + How old are you?");
```

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

```
If (age >= 18) {
```

```
S.O.P(name + " eligible to vote");
```

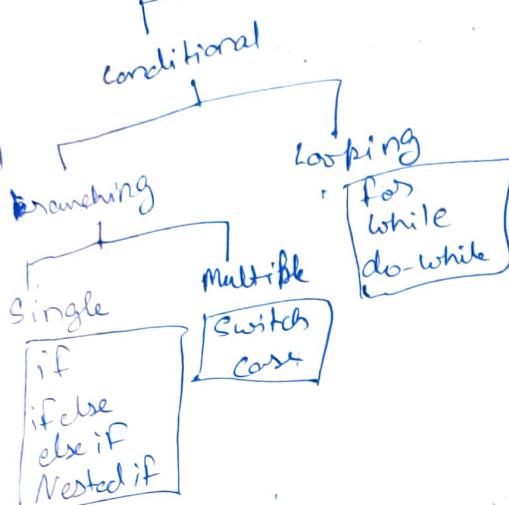
```
else {
```

```
S.O.P(name + " Not eligible to vote");
```

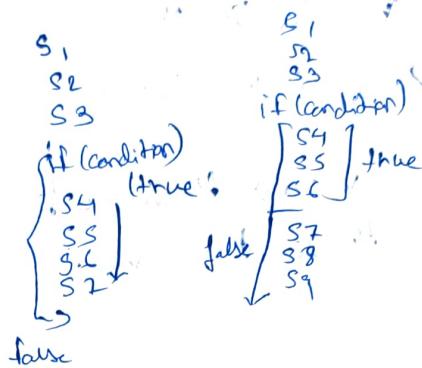
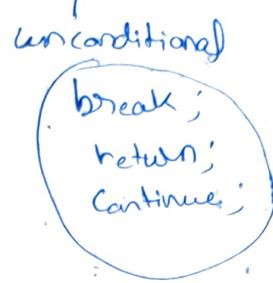
```
S.O.P("Thank You");
```

→ Parenthesis
() → brace
[] → work bracket

Control Statement



$= = \rightarrow$ equals to
 $= \rightarrow$ assignment operator



WAP to check user entered no is special two digit no or not.

Scanner sc = new Scanner(System.in); 5 9

S.O.P ("Enter two digits"); 5+9+5*9

int n = sc.nextInt();

14 + 45 = 59

int d1 = n/10;

int d2 = n%10;

int sum = d1+d2+d1*d2;

if (sum == n)

S.O.P ("It's a special two digit no.");

else

S.O.P ("It's not a special digit no.");

S.O.P ("Thank You");



WAPT read 1 integer value from user & print that no is negative or positive .

WAPT read one integer from user and print that no is even or odd .

WAPT read two integer value from user and print biggest among those two integer .

WAPT read one integer value from user & print it is Zero or non zero .

WAPT read three no from the user & print biggest among those three no .

WAPT read the month no from the user & print the corresponding month name .

WAPT read day no from the user & print corresponding day name .

WAPT 4 integers value from the user & print biggest among the 4 no .

WAPT read 3 floating point no from user & print smallest among those three no .

WAPT read 4 sub marks of a student & print his result either pass or fail ,

biggest no among 3 no.

int $x = \text{sc}.nextInt();$

int $y = \text{sc}.nextInt();$

int $z = \text{sc}.nextInt();$

If ($x > y$)

{ if ($x > z$)

 S.O.P ("x is biggest no");

 else

 S.O.P ("z is biggest no");

 else if ($y > z$)

 S.O.P ("y is biggest no");

 else

 S.O.P ("z is biggest no");

or

|| → logical And

|| → logical OR

if ($x > y$) || ($x > z$)

 S.O.P ("x is biggest no");

else if ($y > z$)

 S.O.P ("y is biggest no");

else

 S.O.P ("z is biggest no");

Q. WAS PT need to take marks of student and print his result fail or pass

int P = sc.nextInt();

int M = sc.nextInt();

int C = sc.nextInt();

int B = sc.nextInt();

String res = "Pass";

if ($P < 35$)

 res = "Fail";

```

if (c < 35);
    res2fail;
if (m < 35)
    res2fail;
if (b < 35)
    res2fail;
S.O.P ("Result is fail");

```

(a)

```

if (P < 35 || C < 35 || m < 35 || b < 35)
    S.O.P ("Result is fail");
else
    S.O.P ("Result is Pass");

```

(b)

```

if (P > 35 && C > 35 && m > 35 && b > 35)
    S.O.P ("Result is Pass");
else
    S.O.P ("Result is fail");

```

Q. WAPT read a integer value from the user and print that integer is digit or no

```

int n = sc.nextInt();
if (n > -10 && n < 10)
    S.O.P ("n is a digit");
else
    S.O.P (n + " is a number");

```

Q. WAPT read the month no from the user and print how many days in that month.

Q. WAPT read one integer value from user and print the integer is two digit or not.

Q. WAPT read the month no from the user and print the month no is valid or not

Q. WASP T read 3 floating point values from user and print the smaller no.

Q. WASP T read 3 distinct values from user and print middle value among these three no.

1. Scanner sc = new Scanner(System.in);
S. O. P ("Enter Month no.");
int n = sc.nextInt();
if (n == 1 || n == 3 || n == 5 || n == 7 || n == 8 ||
n == 10 || n == 12)
S. O. P ("n+ Having 31 day in month");
else if (n == 2)
S. O. P ("n+ Having 28 or 29 days");
else if (n == 4 || n == 6 || n == 9 || n == 11)
S. O. P ("n+ Having 30 days in month");
else
S. O. P ("n+ is not valid");

2. Scanner sc = new Scanner(System.in);
S. O. P ("Enter one digit no.");
int n = sc.nextInt();
if (n >= 0 && n <= 9) || (n < -9 && n > -100)
S. O. P ("n+ is Valid no");
else
S. O. P ("n+ is not Valid no").

3. Scanner Scanned Scannots (System.in).

S. O. P ("Enter month no");

int n = sc.nextInt();

if (n > 0 & n < 13)

S. O. P (n + " month no is Valid").
else

}

S. O. P (n + " month no is not
Valid").

4. Roberts Don.

Scanner Scanned Scanner (System.in)

S. O. P ("Enter 3 floating points").

float n1,

float n2,

float n3;

if (n1 < n2 & n1 < n3)

S. O. P (n1 + " first one is smallest")

elseif (n2 < n1 & n2 < n3)

S. O. P (n2 + " 2nd one is smallest")

else if (n3 < n1 & n3 < n2)

S. O. P (n3 + " third one is smallest")

else (" invalid no").

{
}

S. O. P (" invalid no").

```

int n1 = sc.nextInt();
int n2 = sc.nextInt();
int n3 = sc.nextInt();
if (n1 > n2 && n1 < n3 || (n1 < n2 && n1 > n3)) {
    S.O.P(n1 + " is middle");
} else if (n2 > n1 && n2 < n3 || n2 < n1 && n2 > n3) {
    S.O.P(n2 + " is middle");
} else {
    S.O.P(n3 + " is middle");
}

```

$\frac{3}{3}$
 W.A.T print Sanju if number is divisible by 3
 print Greeta if no is divisible by 5 , print
 Sanju and Greeta if no is divisible by both
 3 & 5 . Otherwise print breakup .

. W.A.T \rightarrow check user entered no is even
 or odd without using modulus operator .

$$(n/2 \times 2 = n)$$

$$\begin{array}{c}
 \overbrace{5/2}^1 \quad \overbrace{2 \times 2}^4 \quad 4 \\
 \overbrace{8/2}^2 \quad \overbrace{4 \times 2}^8 \quad 8 \\
 \overbrace{15/2}^3 \quad \overbrace{7 \times 2}^{14} \quad 14 \\
 \overbrace{18/2}^4 \quad \overbrace{5 \times 2}^{18} \quad 18
 \end{array}$$

W.A.T without using any control
 condition check no is even or
 odd

String st[] = { "Even", "odd" } ;

int n = sc.nextInt();
 S.O.P(n + " is " + st[n % 2]);

Another way without using Control Statement
by using ternary operator

```
int n = Sc.nextInt();
String st = (n % 2 == 0) ? "Even" : "Odd";
S. O. P (n + " is " + st);
```

```
int big = (x > y) ? x : y;
```

```
S. O. P (" Biggest no " + big)
```

Q1. WAP/T print the no is -ve or +ve by using conditional operators

Q2. WAP/T print the no is zero or non zero by using conditional operators.

Q3. WAP/T print biggest among 3 no by using conditional operators.

Q4. WAP/T check year is leap year or not.

Q5. WAP/T check smallest among 3 no. by using conditional operators.

1.

```
int n2 = sc.nextInt();
string st2 = (n > 0) ? "tve" : "-ve";
S.O. P(n + " is " + st2);
```

2.

```
int n2 = sc.nextInt();
string st2 = (n == 0) ? "Zero" : "nonzero";
S.O. P(n + " is " + st2);
```

3.

```
int n1 = sc.nextInt();
int n2 = " ";
int n3 = " ";
int st = (n1 > n2) ? n1 : n2;
string st2 = (n1 > n2) ? "n1" : "n2";
int b = (n3 > st) ? n3 : st;
S.O. P(b + " is bigger");
```

~~int big2 = (n1 > n2) ? n1 : n2;~~

4.

```
int n2 = sc.nextInt();
string st2 = ((n2 % 4 == 2) || (n2 % 4 == 0 & n2 / 100 == 0)) ?
    "Leap" : "Not leap";
S.O. P(n + " is " + st2);
```

5.

```
int n1 = sc.nextInt();
int n2 = " ";
int n3 = " ";
int st2 = (n1 < n2) ? n1 : n2;
int b = (n3 < st2) ? n3 : st2;
S.O. P(b + " is smallest");
```

Bigest among three no.

int big2 ($x > y$) ? $x : y$;

big2 (big $> z$) ? big : z;

S. O. P ("biggest value" + big);

or

int big2 ($x > y$) ? ($x > z$) ? $x : z$: ($y > z$) ? $y : z$;

or

int big2 ($x > y \& x > z$) ? x : ($y > z$) ? $y : z$;

Leap Year

int Y = sc.nextInt();

if ($Y \% 400 \geq 0$)

S. O. P (Y + "is leap year").

else if ($Y \% 4 \geq 0 \& Y \% 100 \neq 0$)

S. O. P (Y + "is leap year").

else

S. O. P (Y + "is not a leap year").

or

if ($Y \% 400 \geq 0 \mid Y \% 4 \geq 0 \& Y \% 100 \neq 0$)
S. O. P ("leap year").

else

S. O. P ("Not a leap year").

looping Control Statements

while,

D1

D2

D3

while (condition)

{

D4

D5

D6

D7, D8

D9

D10

int i=10;

while(i>0)

{ S.O.P("Raghav");

S.O.P("Java");

i--;

} S.O.P("Thank You");

do while

D1

D2

D3

do {

D4

D5

D6

} while (condition);

D7

D8

D9

true

false

int i=10

do {

S.O.P("Raghav");

S.O.P("Java");

i--;

} while (i>0);

S.O.P("Thank You");

for :-

D1

D2

D3

for(D1; D2; D3)

{

D4

D5

D6

true

D7

D8

D9

for(int i=10; i>0; i--)

{

S.O.P("Raghav");

S.O.P("Java");

}

S.O.P("Thank You");

1. W.A.S.P To calculate sum of first n natural no.

2. W.A.S.P To calculate the factorial of N,

3. W.A.S.P Print even no with in N,

4. W.A.S.P Print odd no with in N,

5. W.A.S.P print sum of even nos and odd nos
within in N,

```
1. int n = Sc.nextInt();
   int i = 1;
   while(i <= n) {
       S.O.P(i);
       i++;
   }
```

```
int n = Sc.nextInt();
for(int i = 1; i <= n; i++) {
    S.O.P(i);
}
```

```
2. int n = Sc.nextInt();
   int fact = 1;
   int i = 1;
   while(i <= n) {
       fact = fact * i;
       i++;
   }
   S.O.P(fact);
```

```
int n = Sc.nextInt();
int fact = 1;
for(int i = 1; i <= n; i++) {
    fact = fact * i;
}
S.O.P(fact);
```

```
3. int n = Sc.nextInt();
   int i = 1;
   while(i <= n) {
       if(i % 2 == 0) {
           S.O.P(2 * i);
       } else {
           S.O.P(i);
       }
       i++;
   }
```

$O(n)$

```
int n = Sc.nextInt();
for(int i = 1; i <= n; i++) {
    S.O.P(2 * i);
    if(i % 2 == 0) {
        S.O.P(i);
    }
}
```

```

4. int n = sc.nextInt();
   int i = 1;
   while (i <= n) {
      if (i % 2 == 1) {
         S.O.P(i);
      }
      i++;
   }
}

```

```

5. int n = sc.nextInt();
   int i = 1;
   int oddSum = 0;
   int evenSum = 0;
   while (i <= n) {
      if (i % 2 == 0) {
         evenSum += evenSum + i;
      } else {
         oddSum += oddSum + i;
      }
      S.O.P("EvennoSum" + evenSum);
      S.O.P("OddnoSum" + oddSum);
   }
}

```

```

int n = sc.nextInt();
for (int i = 1; i <= n; i++) {
   if (i % 2 == 1) {
      S.O.P(i);
   }
}
int n = sc.nextInt();
int oddSum = 0;
int evenSum = 0;
for (int i = 1; i <= n; i++) {
   if (i % 2 == 0) {
      evenSum += evenSum + i;
   } else {
      oddSum += oddSum + i;
   }
   S.O.P("EvennoSum" + evenSum);
   S.O.P("OddnoSum" + oddSum);
}

```

$O(n/2)$

```

int n = sc.nextInt();
for (int i = 2; i <= n; i += 2) {
   S.O.P(i);
}

```

WAP JPT check user entered no is perfect no or not.

```
import java.util.Scanner;  
class Perf Perfecto {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        int n = sc.nextInt();  
        int sum = 0;  
        for (int i = 1; i <= n / 2; i++) {  
            if (n % i == 0) {  
                sum = sum + i;  
            }  
        }  
        if (sum == n) {  
            System.out.println(n + " is perfect no");  
        } else {  
            System.out.println(n + " is not perfect no");  
        }  
    }  
}
```

fact of
6 → 1, 2, 3
1+2+3=6
6 → 1, 2, 3
1+2+3=6
So perfect no

WAP JPT check the no is prime no or not.

```
public static void main(String args[]) {  
    Scanner sc = new Scanner(System.in);  
    System.out.print("Enter the no ");  
    int n = sc.nextInt();  
    for (int i = 2; i <= n / 2; i++) {  
        if (n % i == 0) {  
            System.out.println(n + " is not a prime no");  
            return;  
        }  
    }  
    System.out.println(n + " is prime no");  
}
```

w AJP T print the multiplication for user entered no.

```
n = dc.nextInt();
for (int i = 1; i <= 10; i++)
    S.O.P( n + " " + "x" + " " + i + " = " + n * i);
    or (n + "x" + i + " = " + n * i);
```

w AJP T ~~check~~ print how many digits present in user entered no

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



```
} For number 20
do {
    dc *= 10;
    n = n / 10;
} while (n != 0);
```

```
S.O.P(No of digit " " + dc);
```

w AJP T calculate ~~knowing~~, the sum of digit in the no.

w AJP T calculate the sum of square of its digit

w AJP T check user entered no is palindrome or not

w AJP T Count how many even digit & odd digit present in the no.

w AJP T ~~check~~ user entered no is Armstrong or not.

1. WAPT calculate the sum of digit in the no.

```
int n = sc.nextInt();
int sum = 0;
int lastd;
while (n != 0) {
    lastd = n % 10;
    sum = sum + lastd;
    n = n / 10;
}
s.o.p(sum);
```

2. WAPT calculate the sum of squares of its digit

```
int n = sc.nextInt();
int sum = 0;
int lastd;
while (n != 0) {
    lastd = n % 10;
    sum = sum + lastd * lastd;
    n = n / 10;
}
s.o.p(sum);
```

3. WAPT check user entered no is pallindrome or not

```
int n = sc.nextInt();
int num;
int sum = 0;
int lastd;
while (n != 0) {
    lastd = n % 10;
    sum = sum + 10 * lastd;
    n = n / 10;
}
s.o.p(sum);
if (sum == num) {
    s.o.p("Pallindrome");
} else {
    s.o.p("Not Pallindrome");
}
```

4. WAP/T Count how many even digit & odd digit present in the no.

```
int n = sc.nextInt();
int sum = 0;
int lastd;
int Even-digit = 0;
int Odd-digit = 0;
while (n != 0) {
    lastd = n % 10;
    sum = sum + lastd;
    if (sum % 2 == 0) {
        Even-digit++;
    } else {
        Odd-digit++;
    }
    n = n / 10;
}
```

S. O. P ("Even Digit" + Even-digit),

S. O. P ("Odd Digit" + Odd-digit),

5. WAP/T check user entered no is Armstrong no

```
int n = sc.nextInt();
int no = n;
int sum = 0;
int lastd;
while (n != 0) {
    lastd = n % 10;
    sum = sum + lastd * lastd * lastd;
    n = n / 10;
}
if (sum == no) {
    S. O. P (not "is Armstrong no");
} else {
    S. O. P (not "is not Armstrong no");
}
```

Java Main Program

V.I.V.m (String arr[]) // calling method

int a=25, b=17, c=20;

boolean rs = IsPrimeNumber(a); // Calling Statement

S.O.P (a + " is Prime : " + rs);

C.O.P (b + " is Prime : " + IsPrimeNumber(b));

boolean rs1 = IsPrimeNumber(c);

if (rs1) {

S.O.P (c + " is a Prime no");

else {

S.O.P (c + " is not a Prime no");

Formal Parameter

Static boolean IsPrimeNumber (int n)

for (int i=2; i<n/2; i++)

if (n % i == 0)

return false;

return true;

Called Method
Signature / Declaration

Method Body
Body

Skeleton / Temp

WAP define a method to return sum of digits in the no.

class Sumdigit
p.s.v.m (String arr[])

{
 s.o.p (sumofdigits(385));
 s.o.p (sumofdigits(403));
 s.o.p (sumofdigits(38));
 s.o.p (sumofdigits(486));

}

static int sumofdigits (int n) {

 int sum = 0;
 do {
 int r = n % 10;
 sum = sum + r;
 n = n / 10;
 } while (n != 0);

 return sum;

}

I/P \rightarrow 385
403
38
486
O/P \rightarrow 16
7
11
18

1. WAP define a method to return true if the no is perfect otherwise return false.

WAP define a method to return product of digits

3. WAP define a method to return true if the no is special two digit otherwise return false

WAP define a method to return biggest among three no

WAP define a method to return factorial of n.

1. WAPPT define a method to return true if the no is perfect otherwise return false.

```
import java.util.Scanner;
public class Perfectno {
    public static void main (String [] args) {
        Scanner sc = new Scanner (System.in);
        int s = sc.nextInt();
        int n = Perfect(s);
        if (n == 1) {
            System.out.println (s + " is Perfectno");
        } else {
            System.out.println (s + " is not perfectno");
        }
    }
    static int Perfect (int n) {
        int sum = 0;
        for (int i = 1; i <= n / 2; i++) {
            if (n % i == 0) {
                sum = sum + i;
            }
        }
        if (sum == n) {
            return 1;
        } else {
            return 0;
        }
    }
}
```

2. WAPPT define a method to return product of digits.

```
import java.util.Scanner;
public class Product {
    public static void main (String [] args) {
        Scanner sc = new Scanner (System.in);
        int n = sc.nextInt();
        int k = Product1(n);
        System.out.println ("Product of " + n + " is " + k);
    }
    static int Product1 (int n) {
        int product = 1;
        int lastd;
        while (n != 0) {
            lastd = n % 10;
            product = product * lastd;
            n = n / 10;
        }
        return product;
    }
}
```

lastd = n % 10;

Product = Product * lastd;

n = n / 10;

} return Product;

}
}

WAPPT define a method to return true if the no is special
two digit no otherwise return false.

import java.util.*;

public class SpecialTwoDigit {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int n = sc.nextInt();

boolean k = Special(n);

S.O.P("n is special digit no : " + k);

}

static boolean Special(int n) {

int n1 = n / 10;

int n2 = n % 10;

int sum;

sum = ~~n1 * n2 + n1 + n2~~;

if (sum == n && n > 9 && n < 100) {

 return true;

} else {

 return false;

}
}

WAPPT define a method to return biggest among three no.

import java.util.Scanner;

public class BiggestAmongThree {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int n1 = sc.nextInt();

int n2 = sc.nextInt();

int n3 = sc.nextInt();

int k = Special(n1, n2, n3);

S.O.P("k is Biggest among these no");

static int Special(int n1, int n2, int n3) {

if (n1 > n2 && n1 > n3) {

```

    }  

    else if (n2 > n1 & & n2 > n3){  

        return n2;  

    }  

    else {  

        return n3;  

    }  

}

```

5. WAP to define a method to return factorial of n

```

import java.util.Scanner;  

public class Factorial {  

    public static void main (String [] args) {  

        Scanner sc = new Scanner (System.in);  

        int n = sc.nextInt();  

        int r = special (n);  

        System.out.println (r + " is the factorial of " + n);  

    }  

    static int special (int n) {  

        int fact = 1;  

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

            fact *= i;  

        }  

        return fact;  

    }  

}

```

class MainPerfect {
 D. D. V. M (String args)

class Perfect

```

Scanner sc = new Scanner  

sc.nextLine();  

int n = sc.nextInt();  

Perfect p = new Perfect();  

if (p.isPerfectNo (n)) {  

    System.out.println ("Perfect No");  

} else {  

    System.out.println ("Not Perfect");  

}

```

```

boolean isPerfectNo (int n)  

{  

    int sum = 0;  

    for (int i = 1; i <= n / 2; i++) {  

        if (n % i == 0) {  

            sum += i;  

        }  

    }  

    return sum == n;
}

```

Define a method to return true if the no is pallindrome.

return false;
Scanner sc = new Scanner(System.in);
Scanner sc2 = new Scanner(System.in);
sc. nextLine("Enter the no");
int n = sc.nextInt();
int n2 = sc.nextInt();
P2 new Pallindrome(?)
Boolean rs2 = P2.isPalindrome(n);
if (rs2 == true)
 S.O.P("n is Pallindrome");
else
 S.O.P("n is not Pallindrome").

3
Boolean isPalindrome(int n)

int rev = 0, t = n;
do {
 int r = n % 10;
 rev = rev * 10 + r;
 n = n / 10;
} while (n != 0);

return rev == t;

3

3

Define a method to return true if the no is Armstrong no.

otherwise return false.
D.S.V.M(String args[]){
 Scanner sc = new Scanner(System.in);
 Scanner sc2 = new Scanner(System.in);
 sc.nextLine("Enter the no");
 int n = sc.nextInt();
 Boolean rs2 = isArmstrong(n);
 if (rs2 == true)
 S.O.P("n is Armstrong");
 else
 S.O.P("n is not Armstrong");}

static boolean isArmstrong(int n){
 int t = n;
 int sum = 0; rem;
 while (n != 0){
 int rem = n % 10;
 sum = sum + rem * rem * rem;
 n = n / 10;
 }
 return sum == t;

WAPPT check user entered no. is Nevin no or not

1729

$$1+7+2+9 = 19$$

$$1 \cdot 9 \times 9 = 81$$

$$1729 \neq 81$$

S cannot Sc.nextInt

S.O.P("Enter the no"),
int n = Sc.nextInt()

boolean rs = isNiven(n),
if (rs == true)

S.O.P("The no is Niven")

else

S.O.P("The no is not
Niven")

By

Static boolean isNiven(int n) {

int t = n;

sum = 0;

while (n > 0) {

int r = n % 10;

sum = sum + r;

n = n / 10;

}

if (t == sum)

return true;

else

return false;

return t != sum;

1. WAPPT define a method to return n^P.

2. WAPPT define a method to check the no is
Disarium no or not

3. WAPPT define a method to check user entered
no is Strong no or not

4. WAPPT define a method to check user entered no
is Happy no or not

5. WAPPT define a method to return n Fibonacci
no.

1. WAPPT define a method to return n^P.

Scanner Sc = new Scanner(System.in);

int n = Sc.nextInt();

int P = Sc.nextInt();

int k = Power1(n, P);

S.O.P(n + " " + Power1(n, P) + " " + k);

Static int Power1(int n, int P) {

int Power = 1;

while (P != 0) {

Power = Power * n;

P--;

} return Power;

3

Q. WAP to define a method to check the no is Armstrong or not.

Armstrong or not

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

```
class Armstrong
```

```
{ p. D. V. m (String args[])
```

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

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

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

```
boolean r1 = isArmstrong (n);
```

```
if (r1 == true)
```

```
s. o. p ("n is Armstrong no");
```

```
else
```

```
s. o. p ("n is not Armstrong no");
```

3

```
static boolean isArmstrong (int n)
```

```
{ int sum = 0, t = n;
```

```
int dc = CountDigits (n);
```

```
do {
```

```
int r = n % 10;
```

```
sum = sum + Pow (r, dc);
```

```
n = n / 10;
```

```
} while (n != 0);
```

```
return sum == t;
```

3

```
static CountDigit (int a)
```

```
int count = 0;
```

```
do {
```

```
Count++;
```

```
a = a / 10;
```

```
} while (a != 0);
```

```
return count;
```

3

```
static int Pow (int n, int p)
```

```
int Pow = 1;
```

```
while (P > 0)
```

```
Pow = Pow * n;
```

```
P --;
```

```
} return Pow;
```

3

B.D.V.M (String arr[])

{

Scanner sc2 new Scanner (System.in);

S.O.P ("Enter the no"),

int n2 sc.nextInt();

boolean rD2 isDisarium(n),

if (rD2 == true)

S.O.P (n + "is Disarium");

else

S.O.P (n + "is not Disarium");

}

Static boolean isDisarium (int n)

{

int sum = 0, t = n;

int dc = CountDigits(n);

do {

int r = n % 10;

sum = sum + Pow(r, dc);

dc--;

n = n / 10;

while (n > 0);

return sum == t;

}

Static int CountDigit(int a)

{
int count = 0;
do {

Count++;

a = a / 10;

} while (a > 0);

} return count;

Static int Pow(int a, int b)

{
int Pow = 1;

while (b > 0)

{
Pow = Pow * a;

b = b - 1;

} return Pow;

}

Happy no

b. D.V.M()

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

Scanner sc = new Scanner("Enter the no");

int n = sc.nextInt();

boolean rs2 ishappyno(n);

if (rs2 = true)

s.o.p(n + " is happy no");

else s.o.p(n + " is not happy no");

3

static boolean ishappyno(int n)

{

while (n > 9) :

{ int sum = 0;

~~while~~ do {

int r2 n/10;

sum = sum + r2 * 8;

$\exists n = n/10$

\exists while (n != 0);

$\exists n = sum$;

return n == 1 || n == 7;

3

Strong no

Scanner sc = new Scanner (System.in);

int n = sc.nextInt();

boolean rs = isStrong(n);

if (rs == true) {

s.o.p("Strong no");

} else {

s.o.p("not Strong no");

}

static boolean isStrong(int a) {

int sum = 0, t = a;

do {

int r = a % 10;

sum = sum + fact(r);

a = a / 10;

} while (a != 0);

return sum == t;

}

static int fact (int a) {

int sum = 1;

while (a != 0) {

sum = sum * a;

a = a - 1;

} return sum;

}

```

String str = "";
for(int i=1; i<10; i++)
{
    str = str + i;
}
S. O. P(str);

```

1 2 3 4 5 6 7 8 9

```

String str = "";
for(int i=1; i<10; i++)
{
    str = str + i;
}
S. O. P(str);

```

1 + "1"
2 + "1"
3 + "21"
4 + "321"
5 + "4321"
6 + "54321"
7 + "654321"
8 + "7654321"
9 + "87654321"

class Conversion {

```
    string decToBin(int dec)
```

```
{
    string bin = "";
    do {
        int r = dec % 2;
        bin = r + bin;
        dec = dec / 2;
    } while (dec != 0);
    return bin;
}
```

```
    string decToOct(int dec)
```

```
{
    string oct = "";
    do {
        int r = dec % 8;
        oct = r + oct;
        dec = dec / 8;
    } while (dec != 0);
    return oct;
}
```

```
    string decToHex(int dec)
```

```
{
    string hex = "";
    do {
        int r = dec % 16;
        if (r > 9)
            hex = r + hex;
        else
            hex = char(r + 55) + hex;
        dec = dec / 16;
    } while (dec != 0);
    return hex;
}
```

16 | 78
16 | 4 - 14
8 | 9 - 6
8 | 1 - 1
8 | 0 - 1
8 | 10 + SSA
11 | 55 B
12 | 55 C
13 | 55 D
14 | 55 E

Ex "E"
47 "E"
100110

Q) Write a program to convert decimal to binary.

Ans: conversion of

int binToDec(int bin)

{
int dec = 0, p = 1;
do {

int r = bin % 10;

dec = dec + r * p;

p = p * 2;

bin = bin / 10;

} while (bin != 0);

} return dec;

int outToDec (int out)

{
int dec = 0, p = 1;

do {

int r = out % 10;

dec = dec + r * p;

p = p * 8;

out = out / 10;

} while (out != 0);

} return dec;

- Q) all the
1. WAP to print the prime no within N ~~100~~
2. WAP to print all 3 digit pallindrome no.
3. WAP to print the strong numbers m to n.
4. WAP to print the first ⁿ fibonacci no
5. WAP to print all odd no from n to 1.

1110101

1×2^0	1
0×2^1	2
1×2^2	4
0×2^3	8
1×2^4	16
1×2^5	32
1×2^6	64

256.

6×8^0	1
5×8^1	8
2×8^2	64

Pow(20)
Pow(21)
Pow(22)

Scanner sc = new Scanner (System.in);

1. int n = sc.nextInt();
int count = 0;
for (int i = 2; i <= n; i++) {
 if (i % 2 == 0 || i == 0)
 continue;
 for (int j = 2; j <= i - 1; j++) {
 if (i % j == 0) {
 count = count + 1;
 break;
 }
 }
 if (count == 0)
 System.out.println ("prime no");
 else
 count = 0;
}

2.