Basic codes in java:
Hello World, Command line argument.
Taking user input, creating functions

Unit-1

① Hello World in java

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

- In java, all codes must be inside a class. So main fun is created inside demo class.
- Declaration of main fun is **

public static void main (String args[]).

1) public → it can be accessed outside class.
2) static → main is static, so that it can be called w/o creating any obj of class.

eg main fun is called as follows.

demo.main( )

//when OS runs the pgm, it calls the main( )

• for non-static fun, obj needs to be created

eg demo obj ; // create a ~~dem~~ obj of demo
obj. main ( ); & call the main ( ).

3) Void → return type of main must be void.

4) String args [ ] → This an array of strings.

• main fun ~~f~~ must have this parameter.

• Its purpose is to take command line arguments from user.

• Println fun in java.

Its use is :-

System.out. println ("Hello");

1) System :- is an inbuilt class.
It is inside java.lang package.
It is imported automatically.
It ~~H~~ provides fun to perform input/output.

2) Out :- ~~out~~ is an obj inside ~~Sy~~ System class.
It is an obj of PrintStream class.
It is a static member, so it can be accessed by using System class name.

3) println :- it is a fun to print
It also prints new line character.

· Different ways to print :

```
String s1 = "Hello";
  "    s2 = "World";
int  a = 10, b = 20;
```

· Sopln → System.out.println ( )

```
· Sopln ("Hello World");
  Sopln (s1 + s2);
  Sopln ( s1 + "World");
  Sopln ( s1 + a); // Hello10
```

it is converted to string.

To print multiple int, use mul times

```
Sopln (a);      // Sopln (a+b+c); will print the
Sopln (b);                             sum.
Sopln (c);
```

· Sopln ( ); // prints a new line

· Use print ( ) to print w/o new line

```
eg System.out.print ("Hello");    // Hello World.
    "     "    "   ("World");
```

$\boxed{\text{How to run a pgm in java}}$

Step 1:- Compile using $\boxed{\text{javac filename.java}}$

Step 2:- run using $\boxed{\text{java classname}}$

eg open terminal
Save code to desktop
go to desktop: cd desktop
javac $\underbrace{\text{hello.java}}_{\text{filename.}}$
java $\underbrace{\text{demo}}_{\text{Class name.}}$

$\boxed{\text{Command line Arguments}}$

• CLA is a way to take input from user when the pgm is run.

eg i/p can be provided when running

> $\underbrace{\text{java demo}}_{\text{cmd to run pgm}}$ $\underbrace{\text{Rahul 10}}_{\text{input} \rightarrow (\text{CLA})}$

• 2 i/p r provided - these r called CLA.

- These i/p (CLA) r available to the user in the array passed inside main fun.

eg class demo {
    public static void main (String args[ ]) {

        //CLA r stored in this array ↑
        System.out.println(args[0]); // Rahul.
          "   "   (args[1]); // 10.
    }
}.

- This pgm is compiled & run.

```
> javac hello.java.
> java demo Rahul 10
```

These r passed to the args[ ] inside main fun. Then the o/p is printed.

Taking i/p from user.

- Scanner class is used to take i/p from the user. It is inside the java.util package. It needs to imported using "import java.util.Scanner".

Steps to read user input:-

1) import java.util.Scanner;
2) Create an obj of Scanner class.

Scanner scobj = new Scanner($\underbrace{\text{System.in}}_{\text{represents the std. i/p. i.e. Keyboard.}}$);

3) Diff. fun r used to read input:-

scobj.nextInt ( ) → to read int
" .nextDouble() → " " double
" .next Float () → . . .
" .next Line () → to read a string
".next().CharAt (0); → to read Char.
$\underset{\text{zero}}{\downarrow}$

eg ~~to~~ Read str, char & int

```
String s;
Char c;
int a;
Scanner scobj = new Scanner(System.in);
s = scobj.nextLine();
Sopln(s);
c = scobj.next().CharAt(0);
Sopln(c);
a = scobj.nextInt();
```

Note **

- Special precaution needs to be taken when taking input in the order <u>int</u>, then <u>String</u>
- In this case, the nextLine() fun reads the stray '\n' in the buffer.
- To remove this '\n', ~~Et~~. nextLine() fun needs to be called 2 times.

<u>eg</u>
```
int a;
String s;
//when int is read by a string, use
nextLine() fun 2 times.
~~a~~ Scanner scobj = new Scanner(System.in);
a = scobj.nextInt();
sopln(a);
s = scobj.nextLine(); // to remove '\n'
                ~~after~~ from the buffer.

s = scobj.nextLine();
sopln(s);
```

Creating fun in java

- Any fun can be created/defined b4 or after main fun in java.
- eg add ( )
- This must be static, bcz it is called inside the main ( )
- Main() itself is static, so it can call other static fun. main() can't call other non static fun.

eg WAP to create add fun:..

```
Clas demo {
main   Public static void main (string args[]) {
()  →         int a=10; b=20;
              add (a, b);
       }
       public static void add (int x, int y) {
                    Soplm (a+b);
add() →  }      must be static.
}
```

eg WAP to print sq of a no.:-

```
class demo{
    public static void main(String args[]){
        int a=10;
        sq(a);
    }
    public static void sq (int x){
        Soplm (x*x);
    }
}
```

Call by value &
Call by reference

Q** Are fun called by value or
" " ref in java ?

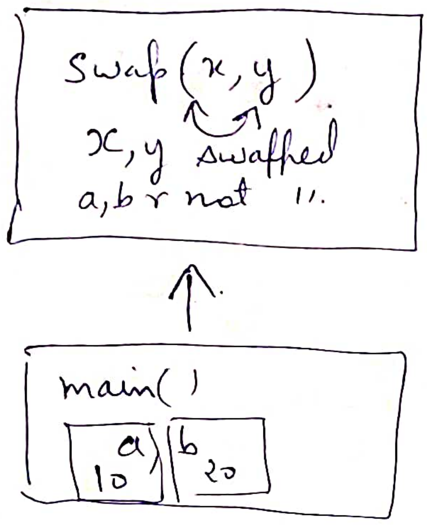
Ans 1) when primitive data types (int, char, float etc)
r passed, then fun is called by value.

2) when obj, arrays, strings r passed to fun
then fun is called by ref.

3) bcoz obj, arrays, strings ~~r implemented~~ variables
variables r reference

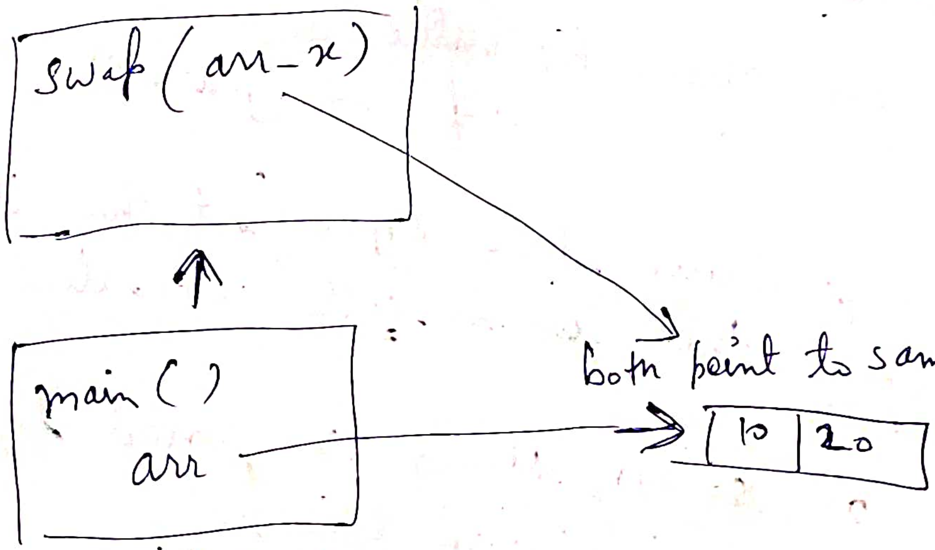
eg 1) A fun to swap 2 int var won't work bcoz int r passed by value.

2) but a fun to swap 2 elements of an array will work bcoz arr r passed by ref.



//A copy of ~~x, y~~ a, b is passed to x, y in swap
• x & y swapped.
a, & b r not swapped.

1- Swap of int



2- Swap of arr el

//when arr is passed to a fun, both arr & ar_x point to same array.

// So arr el swapped in swap fun will also be swapped for arr el in main fun.

eg. Pgm to demonstrate that arr r passed by ref & int r passed by value.

- Create swapint to swap 2 int.
- " swaparr to " el of an arr.

C D { psvm (--) {

means

Class demo {
public static void main (String args[ ]){

```
        int a=10, b=20;
        int ar[] = {10, 20 };
        swapint(a, b); // fun to swap int
        swapar (ar); // "    "    " ar.

        Sopln(a); //10 . a, b r not swapped.
        Sopln(b); //20  . swapint fun doesn't
                             work.

        Sopln(ar[0]); //20 . ar elements r swapped.
        Sopln(ar[1]); //10 . swapar fun works
                             bcoz it is called by ref.
}

public static void swapint(int x, int y){
        int t = x;
          x = y;
          y = t;
}
```

```
public static void swapar(int ar_x[]){
        int t = ar_x[0];
        ar_x[0] = ar_x[1];
        ~~ar~~ ar_x[1] = ~~ar_x~~ t;
    //swap ar[0] & ar[1];
  }
}// demo class ends.
```

Q) How to swap 2 int var?

A Swap w/o using fun.

eg
```
CD { psvm () {
        int a=10; b=20;
        int t=a;
          a=b;      // swap a,b
          b=t;      //  w/o using fun.
       Sopln(a);//20  } values r
       sopln(b);//10  }   swapped.
  }
}
```

Q) 1 Can we ~~useo~~ a fun to swap 2 char in java?

~~Q2) " " " " " " " " " String " " ?~~

Q1) " " " " " find factorial of a ~~A~~ int in java?

Q3) " " " " " to inc a No. by 10 in java?

Q4) " " " " " to find binary representation of a No. in java?

Ans-1 - No. Char r passed by value They won't be swapped.

Ans-2. Yes. No change is need to be made to var. So fun can be used.

3. No. It is passed by value. So it won't be incremented.

4. Yes. No change needs to be made - So fun can be used.