

Day 9

Scanner class:

→ Input the values

```
public class Scanner
```

```
{
```

```
    public void nextInt() → Read
```

```
    {
```

```
    }
```

```
    public void next() →
```

```
    {
```

```
    }
```

OOP Concept:

class → logical entity

collection of variables/method

object → physical entity

Instance of class

class → logical entity

object → physical entity

Man → Human

Cat → Animal

Car → Vehicle

} class

No memory allocation for class

Memory allocation for objects

dog → class

german sheperd / pug / boxer → object

Scanner: we need obj for Scanner class

`Scanner objName = new Scanner();` → object creation
↓
constructor;
↓
same as class name

`Employee(System.in);`

→ `import scanner;`

`System.out.println("Enter your age");`

method call
object name

`Scanner sc = new Scanner();`

`int age = sc.nextInt();` → Reading

`System.out.println("Enter your name");`

`String name = sc.next();`

`System.out.println("Enter a char");`

`char c = sc.next().charAt(0);`
↓
index

with object method call.

①	import scanner
②	class scanner
③	object creation for scanner class.
④	Reading input
⑤	Display the INPUT
⑥	sc.close();

Answer:-

```
int a = 10;
int b = 20;
int c = 30;
```

} variable single value store

- it is set of elements contains same data types
- homogeneous data

→ pit & Nav'g = new 1st (2007).

```
int [ ] a;  
int a [ ];
```

① single dimensional array

array name
int [5] → array size

new keyword

↓
memory allocation

staining heavily

index value = 0-4

Base on \rightarrow

10		30				
0	1	2	3	4	5	

System (a [0]); \rightarrow Repetitive process.

int a [] = {10, 20, 30} ~~no need to give size~~

How to read/print from away

① $\sup(a[\omega])$;

②: $\text{fun } (i, j) = 0, 1, 2, \dots, i+j$
 $\text{sort } a[i]$

③ for $(1st\ i:a)$ send (i) ;

Q1. → Array declare. `int a[] = new int[5];`
 → `sysout ("enter num")`
 → scanner class
 → Read value using for loop. (i < 5)
 Read code
 `sc.nextInt(a[i]);`
 → for loop `i = a`
 `sysout(a[i]);`

if we don't know values:

`int[] marks = new int[120];` (we need to use input)

if we know the values:

`int[] marks = {1, 2, 3, 4, 6, 9, 10};`

Drawback:

- once created we cannot create another location in one Array
- once size is fixed it can't be changed.

1	2	3	4	5	6
0	1	2	3	4	5

Array index = 5

Array length = 5 + 1 = 6

Programs:

`int[] a = new int[5];`

`a[0] = 10;`

`a[1] = 20;`

`a[3] = -40;`

`sysout(a[0]);`