

Day 10

Java → disassemble byte code.

Jdb → To used to find & fix problem.

Reading Input From the user:

Reading an input from the user is a very important task in any programming language which makes the code dynamic.

There are several ways to fetch data from the user. Such as:

- ① JSP/Servlets (Frontend).
- ② Scanner class (Pascal case).
- ③ Console Class.
- ④ Command Line Arguments.
- ⑤ File Handling Class.

Scanner class:→

Scanner class in java is used to take input from the user at the command prompt ~~C++ CLI~~
(CLI - command Line Interface.)

Code:

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

```
class Demo
```

```
{
```

```
    main()
```

```
    {
```

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

```
        int num1 = sc.nextInt();
```

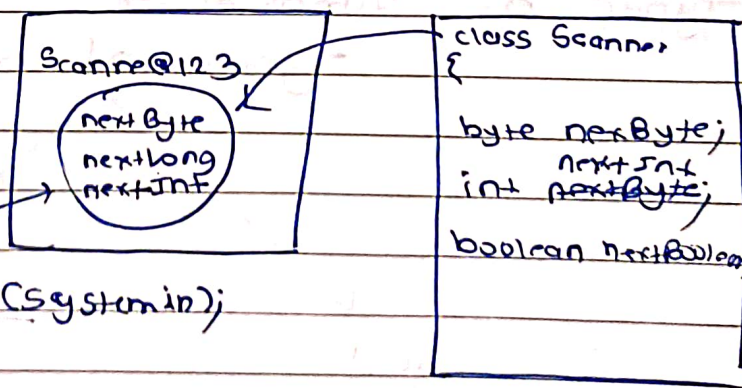
```
        int num2 = sc.nextInt();
```

```
    }
```

```
}
```

Map Area

Java.util. package.



Objects are stored inside JVM memory
(i.e. heap memory)

M	T	W	T	F	S	S
Page No.:						YOUVA
Date:						

Scanner is predefined built-in class which is defined inside java.util package.

Scanner enables the program to accept input from the user.

Java usage System.out as a standard output device and System.in as standard input device.

Console input is not directly supported in java. So we make use of Scanner class.

Syntax:

Scanner variable name = new Scanner(System.in);

- "new" keyword is used to create an object of Scanner, that Scanner object will be stored inside heap area.
- Scanner(System.in) is a constructor call which is used in object creation.
- Once the object creation process completes, the reference is stored inside the variable of Scanner.
- For invoking (call) the methods of Scanner class we use the reference variable.

Eg: import java.util.Scanner;

```
class Demo  
{
```

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

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

```
        int num1 = sc.nextInt();
```

```
        int num2 = sc.nextInt();
```

```
        int op = num1 + num2;
```

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

```
    }
```

```
}
```