

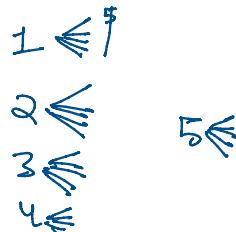
27th Oct → MCT-1

• Nested for loop

- If a loop exists inside the body of another loops. It's called as nested for loop.

for (int i=1 ; i≤ 5 ; i++) // outer for loop

 for (int j= 1; j≤ 5 ; j++) // inner for loop



```
int weeks = 3; ✓  
int days = 7; ✓
```

```
// outer loop prints weeks  
✓ for(int i = 1; i ≤ weeks ; i++) {  
    System.out.println("Week: " + i); ✓  
  
    // inner loop prints days  
    for(int j = 1; j ≤ days ; j++) {  
        System.out.println(" Day: " + j);  
    }  
}
```

Week: 1
Day : 1 ✓
Day : 2 ✓
Day : 3 ✓
Day : 4 ✓
Day : 5 ✓
Day : 6 ✓
Day : 7 ✓

Week: 2
Day : 1
Day : 2
Day : 3
Day : 4
Day : 5
Day : 6
Day : 7

Week: 3
Day : 1
Day : 2
Day : 3
Day : 4
Day : 5
Day : 6
Day : 7

i = ~~1 2 3~~
j = ~~1 2 3~~ ↪

• while () {

 for loop

}

• for loop () {

 while loop

}

• while () {
 while () {

}

```

int weeks = 3;
int days = 7;
int i = 1 ;
// outer loop prints weeks
while(i<= weeks){
    System.out.println("Week: " + i);

    // inner loop prints days
    for(int j =1; j<= days ; j++){
        System.out.println(" Day : " + j);
    }

    i++;
}

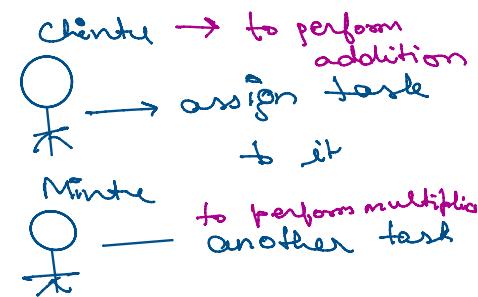
```

- Introduction to Method

- A method is a block of code that performs a specific task.

$4 + 3 \rightarrow$ addition

$4 * 3 \rightarrow$ multiplication



- Dividing a complex problem into smaller chunks make our program easy to understand

& reuse

return type Method name
~~void~~ printNumber()

System.out.println(34); ✓ → body

} → ball → returns

Chintu

public static void main(String[] args) {

° fielder → catch

Code will run first

• fielder → catch

• return type

returns

void

It will not return anything.

System.out.println();

int

Integer ✓

double

decimal ✓

boolean

true/false ✓

float

decimal ✓

string

String ✓

they have the
ability to return
value

```
public class Main {  
    static void printNumber(){  
        System.out.println(34);  
    }  
  
    public static void main(String[] args) {  
        printNumber();  
    }  
}
```

```

public class Main {
    // creating a method
    static void printNumber(int number){ // number -> parameter
        System.out.println(number);
    }
    // main() method
    public static void main(String[] args) {
        int n = 10;
        // calling a method
        printNumber(n); // n -> argument
    }
}

```

n = 10

specific

1st this will

Chiku

Add two numbers

```

// creating a method
static void addNumber(int a, int b){ // a, b -> parameter
    int sum = a+b;
    System.out.println(sum);
}
// main() method
public static void main(String[] args) {
    // calling a method
    addNumber(2,3); // n -> argument
}

```

addNumber
(2,3)
sum=5

there is no return in
the void return type

no. of arguments
≠ no. of parameters

Change the return type to int

```

public class Main {
    // Creating a method
    static int addNumber(int a, int b){ // number -> parameter
        int sum = a+b;
        return sum;
    }
    // main() method
    public static void main(String[] args) {
}

```

print x

2

we have to return
an integer value

```

    }
    // main() method
    public static void main(String[] args) {
        int result;
        // calling a method
        result = addNumber(4, 9); // n -> argument
        System.out.println(result);
    }
}

```

on my

Check Odd Even

```

// creating a method
static boolean oddEvenCheck(int a){ // a -> parameter
    if(a%2 == 0){ // 23
        return true; ✓
    }else{
        return false; ✓
    }
}
// main() method
public static void main(String[] args) {
    int number = 23; // sc.nextInt()
    // calling a method
    boolean result = oddEvenCheck(number); // number -> argument
    System.out.println(result);
}

```

Code Reusability

```

public class Main {
    static double addNumbers(double a, double b){
        double sum = a + b;
        return sum;
    }
}

public static void main(String[] args) {
    double result;
    result = addNumbers(4.5, 9.6); // 14.1
    System.out.println(result); // 14
}

result = addNumbers(-15.4, 10.0); // -5.4
System.out.println(result); // -5.4

```

2 parameter

parameter ✓

Ben 10

└ Omnitrix

aliens

└ Capabilities

└ ... L

```

result = addNumbers(-15.4 , 10.0); → 25.4
System.out.println(result);

result = addNumbers(-12.4 , 18.0);
System.out.println(result);
}
}

```

Code w/ ↴ data type

• Advantages of using Method :

- It makes our code reusable . We can call the same method again and again as per our needs .
- Our code is easier to understand and maintain .
- It allow us to divide the large program into smaller subprograms . So a larger project can be divided among teams members .



Person 1 → tyres .

Person 2 → engine ✓

Person 3 → light .

Person 4 → Design

Add & Multiply two number ✓

```

static int addNumbers(int a , int b){
    int sum = a + b;
    return sum;
}

static int MultiplyNumbers(int a , int b){
    int prod = a * b;
    return prod;
}

public static void main(String[] args) {
    int sum , prod;
    sum = addNumbers(4 , 9); ✓ 13
    System.out.println(sum);

    prod = MultiplyNumbers(4 , 10); → 40
    System.out.println(prod);

    sum = addNumbers(4 , 18); → 22
    System.out.println(sum);
}

```

$\text{sum} = \text{addNumber}(4, 9);$
 $\text{product} = \text{multiplyNumber}(4, 10)$

Classwork Questions

Pattern 1 - Print Stars in same line

Problem	Submissions	Leaderboard	Discussions
---------	-------------	-------------	-------------

Take an integer n and print n stars in the same straight line.

$\text{int } n \rightarrow \text{input}$

$\leq n$

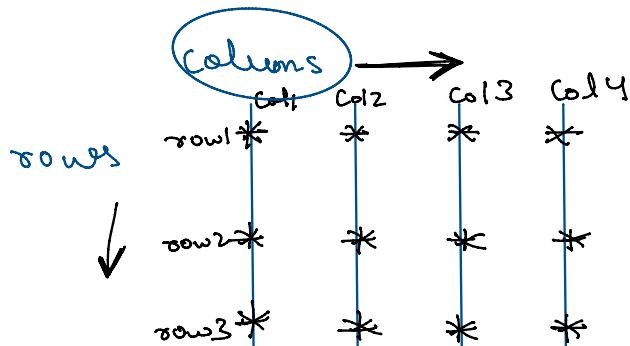
Generic



```

Scanner scn = new Scanner(System.in);
int n = scn.nextInt(); 5
for(int i = 1; i <=n ; i++){
    System.out.print("*"); → *****
}

```



Sample Input 0

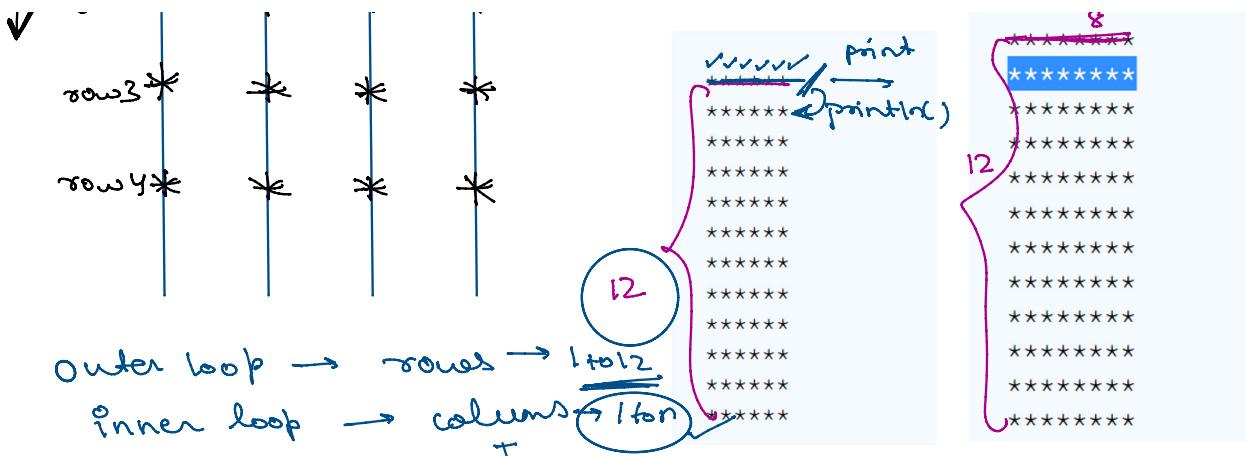


Sample Output 0



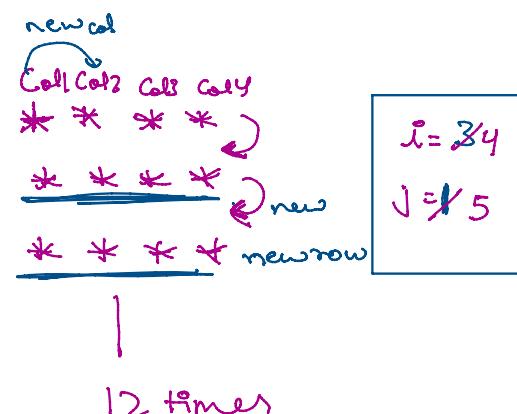
Sample Output 1





/* Enter your code here, read input from

```
Scanner scn = new Scanner(System.in);
int n = scn.nextInt(); 4
for(int i = 1 ; i <= 12 ; i++){ 5
    for(int j = 1; j <= n; j++) { 4
        System.out.print("*"); 3
    }
}
System.out.println(); 2
} 1
```



- outer loop → row
 - inner loop → col

```
static void printPattern(int n){  
    for(int i = 1 ; i <=12 ; i++){  
        for(int j = 1; j <=n ; j++){  
            System.out.print("*");  
        }  
        System.out.println();  
    }  
}
```

```
public static void main(String[] args) {  
    /* Enter your code here. Read input from STDIN  
    Scanner scn = new Scanner(System.in);  
    int n = scn.nextInt();  
    printPattern(n);  
}
```

```
Scanner scn = new Scanner(System.in);
int n = scn.nextInt();
int i = 1 ;
while( i <=12){
    int j = 1;
    while( j <=n ){
        System.out.print("*");
        j++;
    }
    System.out.println();
    i++;
}
```