DSA

Module 2
Numbers and Loops

Author
Srinivas Dande

Java
Learning
Center

2. Control Statements

- Control Statements are used to control the execution flow of Java Program.

There are three types of control statements:

1) Conditional Control Statements
   a. if statement
   b. switch statement
2) Looping Control Statements
   a. for statement
   b. Enhanced for Statement (Added in Java 5)
   c. while statement
   d. do-while statement
3) Unconditional Control Statements
   a. break
   b. continue

2.1. Conditional Control Statements

- Conditional Control Statements are also called as decision-making statements or selection statements.
- It allows the program to select between alternative actions during the program execution.
- The execution will be based on the result of conditional expression.

2.1.1. if Statement

1) Simple If
```
if(<condition>){
  // Statements (IF BLOCK)
}
// Other Statements
```
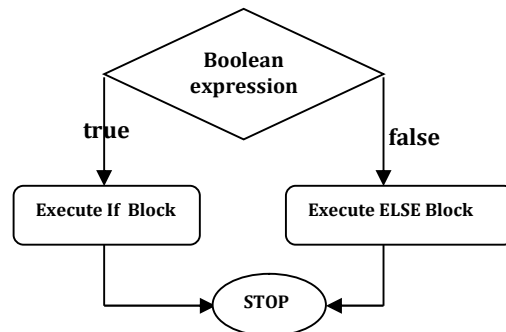- It indicates whether IF BLOCK Statements will be executed or not, as per the condition.

Boolean expression

true

false

Execute If Block

STOP

---
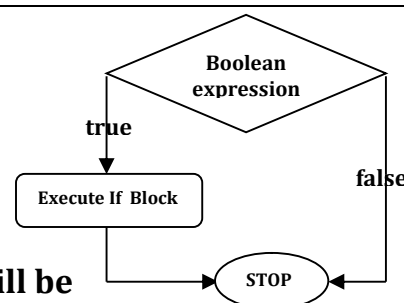
2) If-else Statement

```
if(<condition>){
  // Statements  (IF BLOCK)
}else{
  // Statements  (ELSE BLOCK)
}
```

- It indicates which block will be executed, as per the condition.
- if condition is true IF BLOCK Statements will be executed otherwise ELSE BLOCK Statements will be executed.

3) If Else-if statement

```
if(<condition1>){
  // Statements  (Block 1)
}else if(<condition2>){
  // Statements  (Block 2)
}else if(<condition3>){
  // Statements  (Block 3)
}
...
else{
  // Statements
}
```
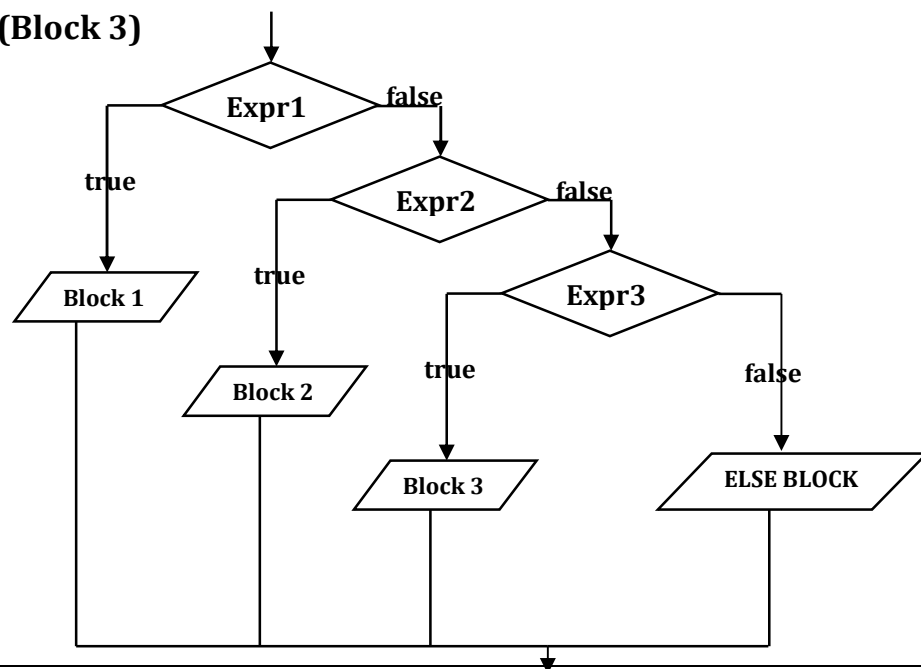
- ◆ **Condition can be any expression which returns boolean value.**
- ◆ **Block notation is mandatory when if block or else block have more than one statement.**
- ◆ **Block notation is optional when if block or else block have only one statement.**
- ◆ **When you declare any variable within if block or else block then that variable can be accessed within that block only. It can't be accessed from outside that block.**

---

1) Demo1.java

```
package com.jlcindia;

/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
**/
class Demo1{
public static void main(String as[]) {
int a=42;
if(a % 2==0)
        System.out.println("Number is EVEN");
else
        System.out.println("Number is ODD");
}
}
```

---

2) Demo2.java

```
package com.jlcindia;
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
**/
class Demo2{
public static void main(String as[]) {
int a=45;
if(a % 5==0)
        System.out.println("Number is Divisible by 5");
else
        System.out.println("Number is Not Divisible by 5");
}
}
```

3) Demo3.java

```
package com.jlcindia;
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
**/
class Demo3{
public static void main(String as[]) {
int a=10;
int b=20;
int max=0;
if(a>b)
        max=a;
else
        max = b;

System.out.println("Max is "+max);

}
}
```

4) Demo4.java

```
package com.jlcindia;

/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
**/
class Demo4{
public static void main(String as[]) {
int a=10;
int b=20;
int c=15;
int max=0;
if(a>b && a> c)
        max=a;
else if(b> a && b>c)
        max = b;
else
        max=c;
System.out.println("Max is "+max);
}
}
```

2.1.2 switch Statement

- Switch statement allows the program to select one action among multiple actions during the program execution.

  Syntax:

  ```
  switch(<variable/expression/value>){
  case <val1>:
      // Statements    [<break>];
  case <val2>:
      // Statements    [<break>];
  . . .
  default:
      // Statements
  }
  ```

- First switch expression will be evaluated and then tries to find case value matching with resultant value of the expression.
- If matching case is found then the statements related to that case will be executed otherwise statements related to default case will be executed.
- Following types of values can be passed in switch as an argument up to Java 2:
  - byte
  - short
  - char
  - int
- enum can be passed in switch as an argument from Java 5.
- String can also be passed in switch as an argument from Java 7.

5) Demo5.java

```java
package com.jlcindia;
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
**/
class Demo5{
public static void main(String as[]) {
int a=3;
switch(a){
case 0:
        System.out.println("Sunday");
        break;
case 1:
        System.out.println("Monday");
        break;
case 2:
        System.out.println("Tuesday");
        break;
case 3:
        System.out.println("Wednesday");
        break;
case 4:
        System.out.println("Thursday");
        break;
case 5:
        System.out.println("Friday");
        break;
case 6:
        System.out.println("Saturday");
        break;
default:
        System.out.println("Wrong Input");
}

}
}
```

6) Demo6.java

```java
package com.jlcindia;
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
**/
class Demo5{
public static void main(String as[]) {

char ch='A';
if(ch>=65 && ch<=90){
switch(ch){
case 'A':
case 'E':
case 'I':
case 'O':
case 'U':
  System.out.println("Character is Vowel");
  break;
default:
   System.out.println("Character is Consonant");
}
}else{
  System.out.println("Invalid Alphabet");
}

}
}
```
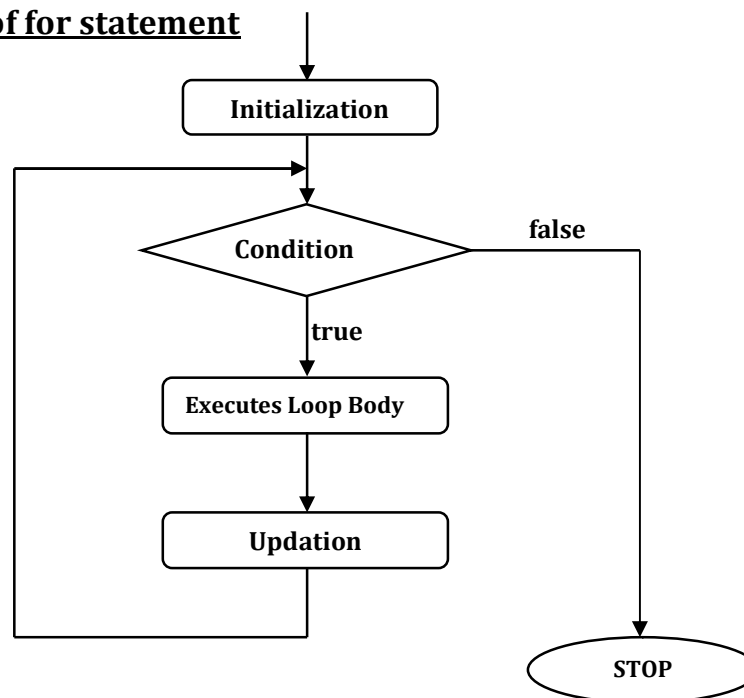
2.2. Looping Control Statements

- ❖ **Looping Control statements are also called as Iteration statements.**
- ❖ **These statements are used to execute a block of statements repeatedly.**
- ❖ **An expression is required for looping control statements which returns boolean value.**
- ❖ **Block notation is mandatory when looping block has more than one statement.**
- ❖ **Block notation is optional when looping block has only one statement.**
- ❖ **When you declare any variable within loop block then that variable can be accessed within that block only. It can't be accessed from outside that block.**

2.2.1 for Statement

Syntax:

> **for(<initialization> ; <condition> ; <updation>)**
>
> **{**
> **// Statements**
> **}**

Processing Flow of for statement

```
                    ┌──────────────────┐
                    │  Initialization  │
                    └──────────────────┘
                              │
          ┌───────────────────┤
          │              ◇ Condition ◇────── false ──┐
          │                   │                        │
          │                  true                      │
          │         ┌──────────────────┐               │
          │         │ Executes Loop Body│              │
          │         └──────────────────┘               │
          │                   │                         │
          │         ┌──────────────────┐               │
          │         │    Updation      │              ( STOP )
          │         └──────────────────┘
          └───────────────────┘
```

a) Initialization Statement will be executed only once first time.

b) Conditional & Updation statement can be executed more than once.

7) Demo7.java

```java
package com.jlcindia;
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
**/
class Demo7{
public static void main(String as[]) {

int num=10;
for(int i=1;i<=num;i++){
System.out.println(i);
}


}
}
```

8) Demo8.java

```java
package com.jlcindia;
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
**/
class Demo8{
public static void main(String as[]) {

int num=25;
for(int i=1;i<=num;i++){
if(i % 2 == 0)
   System.out.println(i);
}
}
}
```

2.2.2 Enhanced For Statement

- ◆ **It is a new feature added in Java 5.**
- ◆ **It is also called as for-each loop.**
- ◆ **The enhanced for-loop is mainly used to access the elements from Collection/Array sequentially.**

Syntax:
for(<data Type> varName : array | Collection) {
// Statements (BLOCK)
}

9) Demo9.java

```
package com.jlcindia;

/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
**/
class Demo9{
public static void main(String as[]) {

int arr[]={99,88,77,66,55};

System.out.println("Length  :"+arr.length);

for(int element []: arr){
     System.out.println(element);

}
}
```
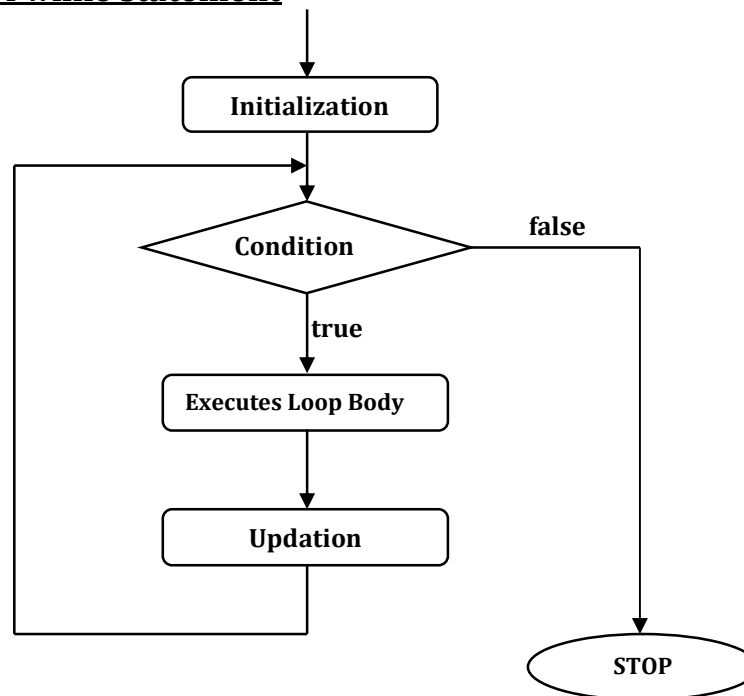
2.2.3 while Statement

Syntax:

<initialization>

while(<condition>){

// Statements (LOOP BODY)

<updation>

}

* condition of while statement is mandatory and must be boolean type.

Processing Flow of while statement

```
        ┌──────────────────┐
        │  Initialization  │
        └──────────────────┘
                 │
       ┌─────────▼─────────┐
       │     Condition     │────false────┐
       └───────────────────┘             │
                 │true                    │
        ┌────────▼─────────┐              │
        │ Executes Loop Body│             │
        └──────────────────┘             │
                 │                        │
        ┌────────▼─────────┐              │
        │     Updation     │          ┌───▼───┐
        └──────────────────┘          │ STOP  │
                                       └───────┘
```

10) Demo10.java

```
package com.jlcindia;
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
**/
class Demo10{
public static void main(String as[]) {

int i=1;
while(i<=10){
System.out.println(i);
i++;
}


}
}
```

2.2.4 do-while Statement

- **When you are using for statement or while statement then it verifies the condition before executing the block.**
- **So in the case of for statement and while statement, when first time condition is false then the block will not be executed.**
- **for and while statement are also called as Entry Controlled Loop.**
- **If you want to execute the block at least once then use do-while statement.**

Using do-while loop

- **In the case of do-while first block of statements will be executed and then condition will be verified.**
- **do-while statement is also called as Exit Controlled Loop.**

Processing Flow of while statement
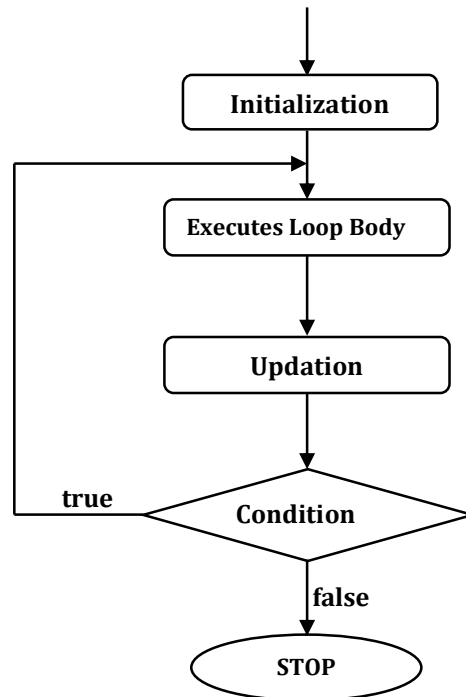

```
11) Demo11.java

package com.jlcindia;
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
**/
class Demo11{
public static void main(String as[]) {

int i=1;
do{
System.out.println(i);
i++;
}while(i<=10);


}
}
```

2.3 Unconditional Control Statements

2.3.1 break Statement

- **break is a keyword.**
- **It can be used within switch or any looping statements.**
- **It is used to terminate the execution of the current looping/switch statement.**
- **break can be used in two ways:**
  - **break;**
  - **break <label>;**

---

12) Demo12.java

```java
package com.jlcindia;

/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
**/
class Demo12{
public static void main(String as[]) {

for(int i=0;i<10;i++){
System.out.println(i);
break;
System.out.println("---Inside Loop---");
}
System.out.println("---Outside Loop---");

}
}
```

2.3.2 continue Statement

- ◆ **continue is a keyword.**
- ◆ **It can be used within any looping statements.**
- ◆ **It is used to continue the execution of the current looping statement with next iteration.**
- ◆ **continue can be used in two ways:**
    - o **continue;**
    - o **continue <label>;**

---

13) Demo13.java

```java
package com.jlcindia;

/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
**/
class Demo13{
public static void main(String as[]) {

for(int i=0;i<10;i++){
System.out.println(i);
continue;
System.out.println("---Inside Loop---");
}
System.out.println("---Outside Loop---");

}
}
```

2.4. Programs on Numbers and Loops

1) Lab1.java

```java
package com.jlcindia;

// 1. WAP to Swap Two Numbers without third Variable.

/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */
class Lab1 {
public static void main(String as[]) {

        int a=10;
        int b=20;

        System.out.println("a = "+a+ " , "+ "b = "+b);

        a= a + b;
        b= a - b;
        a= a - b;

        System.out.println("a = "+a+ " , "+ "b = "+b);

}
}
```

2) Lab2.java

package com.jlcindia;

// 2. WAP to find Max of Two Numbers.

```
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */
class Lab2 {
public static void main(String as[]) {

            int a=20;
            int b=10;

            int max = 0;

            if(a!=b) {

            if(a>b) {
                    max= a;
            }
            else {
                    max=b;
            }
            System.out.println("Max :  "+max);

            }
            else {
                    System.out.println("Both are Same  ");
            }

}
}
```

4) Lab4.java

package com.jlcindia;

// 4. WAP to Print the Even Numbers from 1 to N.

```
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */
class Lab4 {
public static void main(String as[]) {

        int n = 10;

        for (int i = 1; i <= n; i++) {
            if (i % 2 == 0)
                    System.out.println(i);
        }

}
}
```

5) Lab5.java

package com.jlcindia;

// 5. WAP to Print the Numbers Divisible by 5 from 1 to N.

```
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
**/
class Lab5 {
public static void main(String as[]) {

        int n = 25;
        for (int i = 1; i <= n; i++) {
            if (i % 5 == 0)
                    System.out.println(i);
        }


}
}
```

6) Lab6.java

package com.jlcindia;

// 6. WAP to Print Divisibles of the given Number.

```
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
**/
class Lab6 {
public static void main(String as[]) {

            int n = 10;
            for (int i = 1; i <= n; i++) {
                    if (n % i == 0)
                            System.out.println(i);
            }


}
}
```

7) Lab7.java

package com.jlcindia;

// 7. WAP to Find the Number of Divisibles of the given Number.

```java
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
**/
class Lab7 {
public static void main(String as[]) {

int n = 10;
            int count = 0;

            for (int i = 1; i <= n; i++) {
                  if (n % i == 0)
                        count ++;
            }

            System.out.println("Count : "+ count);

}
}
```

8) Lab8.java

package com.jlcindia;

// 8. WAP to Find whether given Number is Prime or Not.

```java
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
**/
class Lab8 {
public static void main(String as[]) {


        int n = 10;
        int count = 0;

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

            if (n % i == 0) {
                count ++;
                break;
            }

        }

        if(count==0) {
            System.out.println(n + "  is Prime: ");
        }else {
            System.out.println(n + "  is Not Prime: ");
        }


}
}
```

9) Lab9.java

package com.jlcindia;

// 9) WAP to print the prime numbers between M to N.

```java
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */
class Lab9 {
public static void main(String as[]) {
        private static boolean isPrime(int num) {
                int count = 0;
                boolean primeFlag=false;
                for (int i = 2; i <= num / 2; i++) {
                        if (num % i == 0) {
                                count++;
                                break;
                        }
                }
                if (count == 0) {
                        primeFlag = true;
                }
              return primeFlag;
        }

        public static void main(String[] args) {
                int m= 1;
                int n = 25;
                for (int num = m; num <= n; num++) {
                                if (isPrime(num)) {
                                System.out.println(num);
                        }
                }
}
}
```

9A) Lab9A.java

package com.jlcindia;

// 9A) WAP to print the prime numbers between M to N.

```java
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */
class Lab9A {
public static void main(String as[]) {

        private static boolean isPrime(int num) {

                boolean primeFlag=true;
                for (int i = 2; i <= num / 2; i++) {
                        if (num % i == 0) {
                                primeFlag=false;
                                break;
                        }
                }
                return primeFlag;
        }

        public static void main(String[] args) {
                int m= 10;
                int n = 25;
                for (int num = m; num <= n; num++) {
                                if (isPrime(num)) {
                                System.out.println(num);
                        }
                }

}
}
```

9B) Lab9B.java

package com.jlcindia;

// 9B) WAP to print the prime numbers between M to N.

```java
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
**/
class Lab9B {
public static void main(String as[]) {

        private static boolean isPrime(int num) {

                for (int i = 2; i <= num / 2; i++) {
                        if (num % i == 0) {
                                return false;
                        }

                }
                return true;
        }

        public static void main(String[] args) {

                int m= 10;
                int n = 25;
                for (int num = m; num <= n; num++) {
                                if (isPrime(num)) {
                                System.out.println(num);
                        }
                }

}
}
```

10) Lab10.java

package com.jlcindia;

```java
// 10) WAP to find sum of numbers from 1 to n.
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
**/
public class Lab10 {
    public static void main(String[] args) {
        int num = 5;
        int sum = 0;
        for (int i = 1; i <= num; i++) {
            sum = sum + i;
        }
        System.out.println("Sum : "+sum);
    }
}
```

10A) Lab10A.java

package com.jlcindia;

```java
// 10) WAP to find sum of numbers from m to n.
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
**/
public class Lab10A {
    public static void main(String[] args) {
        int m= 1;
        int n = 5;
        int sum = (m + n) * n/2 ;
        System.out.println("Sum : "+sum);
    }
}
```

11) Lab11.java

package com.jlcindia;

// 11) WAP to find sum of squares of numbers from 1 to n.

```
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */

public class Lab11 {

        public static void main(String[] args) {

                int num = 5;
                int sum = 0;
                for (int i = 1; i <= num; i++) {
                        sum = sum +  i * i ;   // i contains each Number
                }

                System.out.println("Sum : "+sum);
        }
}
```

12) Lab12.java

package com.jlcindia;

// 12) WAP to find sum of cubes of numbers from 1 to n.

```java
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */

public class Lab12 {

        public static void main(String[] args) {

                int num = 3;
                int sum = 0;
                for (int i = 1; i <= num; i++) {
                        sum = sum +  i * i * i ;   // i contains each Number
                }

                System.out.println("Sum : "+sum);
        }
}
```

13) Lab13.java

package com.jlcindia;

// 13) WAP to find sum of divisible of a given number

```
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */

public class Lab13 {

        public static void main(String[] args) {

                int n = 10; // Given Number
                int sum = 0;

                for (int i = 1; i <= n; i++) {
                        if (n % i == 0)   //i is having each number
                                sum = sum +i;
                }

                System.out.println("Sum : "+ sum);
        }

}
```

14) Lab14.java

package com.jlcindia;

// 14) WAP to find whether given number is perfect or not.

```java
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */

public class Lab14 {

        public static void main(String[] args) {

                int num = 6; // Given Number
                int sum = 0;

                for (int i = 1; i <= num /2 ; i++) {
                        if (num % i == 0) // i is having each number
                                sum = sum + i;
                }

                if (sum == num)
                        System.out.println(num + " is Perfect Number");
                else
                        System.out.println(num + " is Not Perfect Number");
        }

}
```

15) Lab15.java

```java
package com.jlcindia;

// 15) WAP to display the perfect numbers from m to n.

/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
**/
public class Lab15 {
        private static boolean isPerfect(int num) {
                int sum = 1;
                for (int i = 2; i <= num /2 ; i++) {
                        if (num % i == 0) // i is having each number
                                sum = sum + i;
                }

                if (sum == num)
                        return true;
                else
                        return false;
        }

        public static void main(String[] args) {
                int m= 1;
                int n = 50;
                for(int k=m;k<=n;k++) {
                        if(isPerfect(k)) {
                                System.out.println(k);
                        }
                }

        }
}
```

16) Lab16.java

package com.jlcindia;

// 16) WAP to Print the digits of given number?

```
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */

public class Lab16 {

        public static void main(String[] args) {

                int num = 123;

                while(num!=0) {
                        int rem = num %10;
                        System.out.println(rem);
                        num = num /10;
                }

        }
}
```

17) Lab17.java

package com.jlcindia;

// 17) WAP to find number of digits of given number?

```java
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */

public class Lab17 {

	public static void main(String[] args) {

		int num = 123;
		int count = 0;

		while(num!=0) {
			//int rem = num %10;
			count++;
			num = num /10;
		}

		System.out.println("No of Digits : "+ count);

	}
}
```

18) Lab18.java

package com.jlcindia;

// 18) WAP to find sum of individual digits of given number?

```
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */

public class Lab18 {

        public static void main(String[] args) {

                int num = 123;
                int sum = 0;

                while(num!=0) {
                        int digit = num %10;
                        sum = sum +digit;
                        num = num /10;
                }

                System.out.println("Sum : "+ sum);

        }
}
```

18A) Lab18A.java

package com.jlcindia;

// 18) WAP to find sum of individual digits of given number?

```java
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */

public class Lab18A {

        public static void main(String[] args) {

                int num = 123;
                int sum = 0;

                for(;num!=0;num = num /10) {
                        int rem = num %10;
                        sum = sum +rem;
                }

                System.out.println("Sum : "+ sum);

        }
}
```

19) Lab19java

package com.jlcindia;

// 19) WAP to find sum of Squares of individual digits of given number?

```java
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */

public class Lab19 {

        public static void main(String[] args) {

                int num = 123;
                int sum = 0;

                while(num!=0) {
                        int digit = num %10;
                        sum = sum + digit * digit ;
                        num = num /10;
                }

                System.out.println("Sum : "+ sum);

        }
}
```

20) Lab20.java

package com.jlcindia;

// 20) WAP to find sum of cubes of individual digits of given number?

```
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */

public class Lab20{

        public static void main(String[] args) {

                int num = 123;
                int sum = 0;

                while(num!=0) {
                        int digit = num %10;
                        sum = sum + digit * digit * digit;
                        num = num /10;
                }

                System.out.println("Sum : "+ sum);

        }
}

// Iterations : No of Digits times
```

21) Lab21.java

package com.jlcindia;

// 21) WAP to find the Factorial of given Number.

```
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
**/

public class Lab21{

        public static void main(String[] args) {

                int num = 1;
                int fact = 1;
                for(int i=2; i<= num;i++) {
                        fact = fact * i;
                }
                System.out.println("Factorial is : "+ fact);

        }
}
```

21A) Lab21A.java

package com.jlcindia;

import java.math.BigInteger;

// 21) WAP to find the Factorial of given Number.

```
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */

public class Lab21A{

        public static void main(String[] args) {

                int num = 200;
                BigInteger fact = BigInteger.valueOf(1);
                for(int i=2; i<= num;i++) {
                        fact = fact.multiply(BigInteger.valueOf(i)) ;
                }

                System.out.println("Factorial is : "+ fact);

        }
}

// For Large Numbers , use BigInteger class
```

22) Lab22.java

package com.jlcindia;

//22) WAP to find sum of Factorials of individual digits of given number?

```
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
**/
public class Lab22 {
        private static  int factorial(int num) {

                int  fact = 1;
                for (int i = 2; i <= num; i++) {
                        fact = fact * i;
                }
                return fact;
        }

        public static void main(String[] args) {

                int num = 1234;
                int sum = 0;

                while(num!=0) {
                        int digit = num %10; //Each Digit
                        sum = sum + factorial(digit);
                        num = num /10;
                }

                System.out.println("Sum : "+ sum);
        }
}
```

23) Lab23.java

package com.jlcindia;

// 23) WAP to find whether given number is strong or not.

```java
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
**/
public class Lab23 {
        private static int factorial(int num) {

                int fact = 1;
                for (int i = 2; i <= num; i++) {
                        fact = fact * i;
                }
                return fact;
        }

        public static void main(String[] args) {

                int num = 145;
                int sum = 0;
                int numCopy = num;
                while (num != 0) {
                        int digit = num % 10;
                        sum = sum + factorial(digit);
                        num = num / 10;
                }

                if (sum == numCopy)
                        System.out.println(numCopy + " is Strong Number ");
                else
                        System.out.println(numCopy + " is Not Strong Number ");
        }
}
```

24) Lab24.java

package com.jlcindia;

// 24) WAP to display the strong numbers from m to n.

```
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */

public class Lab24 {
        private static int factorial(int num) {

                int fact = 1;
                for (int i = 2; i <= num; i++) {
                        fact = fact * i;
                }
                return fact;
        }

        private static boolean isStrong(int num) {
                int sum = 0;
                int numCopy = num;
                while (num != 0) {
                        int digit = num % 10;
                        sum = sum + factorial(digit);
                        num = num / 10;
                }

                if (sum == numCopy)
                        return true;
                else
                        return false;
        }
```

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

        int m = 1;
        int n = 150;

        for (int i = m; i <= n; i++) {
            if (isStrong(i)) {
                System.out.println(i);
            }
        }

    }
}
```

25) Lab25.java

package com.jlcindia;

// 25) WAP to find whether given number is armstrong or not.

```java
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */

public class Lab25 {

        private static int cube(int num) {

                return num * num * num;
        }

        public static void main(String[] args) {

                int num = 153;
                int sum = 0;
                int numCopy = num;
                while (num != 0) {
                        int digit = num % 10;
                        sum = sum + cube(digit);
                         num = num / 10;  // Changing the Number
                }

                if (sum == numCopy)
                  System.out.println(numCopy + " is ArmStrong Number ");
                else
                   System.out.println(numCopy + " is Not ArmStrong Number ");
        }
}
```

26) Lab26.java

package com.jlcindia;

//26) WAP to display the Armstrong numbers from 1 to n.

```java
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
**/

public class Lab26 {

        private static int cube(int num) {
                return num * num * num;
        }

        private static boolean isArmStrong(int num) {

                int sum = 0;
                int numCopy = num;
                while (num != 0) {
                        int digit = num % 10;
                        sum = sum + cube(digit);
                        num = num / 10;
                }

                if (sum == numCopy)
                        return true;
                else
                        return false;
        }

        public static void main(String[] args) {

                int m = 1;
                int n = 1000;
```

```
        for (int i = m; i <= n; i++) {
                if (isArmStrong(i)) {
                        System.out.println(i);
                }
        }
    }
}
```

27) Lab27.java

package com.jlcindia;

// 27) WAP to print the Reverse of the given number.

```
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
**/
public class Lab27 {
        public static void main(String[] args) {

                int num = 1234;
                int sum = 0;

                while (num != 0) {
                        int digit = num % 10;
                        sum = sum * 10 + digit ;
                         num = num / 10;  // Changing the Number
                }
                        System.out.println(" Reverse  Number is "+ sum);
        }
}
```

28) Lab28.java

package com.jlcindia;

// 28) WAP to find whether given number is Palindrome or not.

```
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */

public class Lab28 {

        public static void main(String[] args) {

                int num = 121;
                int numCopy = num;
                int rev = 0;

                while (num != 0) {
                        int digit = num % 10;
                        rev = rev * 10 + digit ;
                         num = num / 10;  // Changing the Number
                }

                if(numCopy == rev)
                        System.out.println( numCopy + " is Palindrome");
                else
                        System.out.println( numCopy + " is NOT Palindrome");


        }
}
```

29) Lab29.java

package com.jlcindia;

```
/*
 29) WAP to print the fibonocii series.
  0  1 1 2 3 5 8 13
 */


/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* **/

public class Lab29 {
       public static void main(String[] args) {

               int n = 2; // Number of values Required in Series
               // Validation Code
               int firstNum = 0;
               int secondNum = 1;

               System.out.print(firstNum + "\t" + secondNum);

               int nextNum = 0;

                   for (int i = 3; i <= n; i++) {
                           nextNum = firstNum + secondNum;
                           System.out.print("\t" + nextNum);
                           firstNum = secondNum;
                           secondNum = nextNum;
                   }
       }
}
```

29A) Lab29A.java

package com.jlcindia;

```java
/*
 29) WAP to print the fibonocii series.
  0  1 1 2 3 5 8 13
 */


/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */
public class Lab29A {
        public static void main(String[] args) {

                int n = 5; // Number of values Required in Series
                if(n<2) {
                        System.out.println("Invalid Input");
                }else {

                int firstNum = 0;
                int secondNum = 1;
                System.out.print(firstNum + "\t" + secondNum);


                int nextNum = 0;
                for (int i = 1; i <= n-2; i++) {
                                nextNum = firstNum + secondNum;
                                System.out.print("\t" + nextNum);
                                firstNum = secondNum;
                                secondNum = nextNum;
                    }
                }

        }
}
```

30) Lab30.java

```java
package com.jlcindia;

/*
 30) WAP to print the following series.
1 1 6 4 11 9 16 16 21 25
 */

/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
**/

public class Lab30 {
       public static void main(String[] args) {

               int n= 5;  // Number of values Required in Series

               //Input - Only Even Numbers

               int a = 1;
               int b = 1;

               for (int i = 1; i <= n/2; i++) {
                       System.out.print(a+"\t"+b+"\t");
                       a = a+5;
                       b = (i+1) * (i+1);
               }

       }
}
```

31) Lab31.java

package com.jlcindia;

```java
/*
 31) WAP to print the following series.
1 1 3 8 6 27 9 64 12 125
 */


/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */

public class Lab31 {
        public static void main(String[] args) {

                int n= 5;  // Number of values Required in Series

                //Input - Only Even Numbers

                int a = 1;
                int b = 1;

                for (int i = 1; i <= n/2; i++) {
                        System.out.print(a+"\t"+b+"\t");
                        a = 3 * i;
                        b = i+1;
                        b = b * b * b;
                }


        }
}
```

32) Lab32.java

```
package com.jlcindia;

/*
32) WAP to evaluate following series.
1+2/x+3/x2+ 4/x3+ 5/x4+

 x = 1 => 1+2/x+3/x2 => 1 + 2 + 3 => 6
 x = 2 => 1+2/x+3/x2 => 1 + 1 + 3/4 => 2.75
 */


/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */
public class Lab32 {
        private static double power( int n , int p) {
                double power = 1;
                for(int i=1;i<=p;i++) {
                        power = power * n;
                }
                return power;
        }

        public static void main(String[] args) {
                int n= 3;  // Number of values Required in Series
                int x = 2;
                double sum= 0.0 ;

                for (int i = 1; i <= n; i++) {
                        sum =  sum +  i / power(x,i-1);
                }

                System.out.println("Sum : "+sum);
        }
}
```

33) Lab33.java

```java
package com.jlcindia;
/*
33) WAP to evaluate following series.
1-2/x+3/x2- 4/x3+ 5/x4-6/x5

 x = 1 => 1-2/x+3/x2 => 1 - 2 + 3 => 2
 x = 2 => 1-2/x+3/x2 => 1 - 1 + 3/4 => 0.75
 */

/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */
public class Lab33 {
	private static double power( int n , int p) {
		double power = 1;
		for(int i=1;i<=p;i++) {
			power = power * n;
		}
		return power;
	}
	public static void main(String[] args) {
		int n= 3;  // Number of values Required in Series
		int x = 1;
		double sum= 0.0 ;
		int sign = 1;

		for (int i = 1; i <= n; i++) {
			sum =  sum  + sign * (  i / power(x,i-1) );
			sign = sign * -1;
		}
		System.out.println("Sum : "+sum);
	}
}
```

34) Lab34.java

```java
package com.jlcindia;
/*
34) WAP to evaluate following series.
1/(x-1)! + x/x! + x2/(x+1)! + x3/(x+2)! + x4/(x+3)!


 x = 1 => 1+1+1/2 => 2.5
 x = 2 => 1+1+4/6= > 2.66
 */


/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */
public class Lab34 {

	private static double power( int n , int p) {
		double power = 1;
		for(int i=1;i<=p;i++) {
			power = power * n;
		}
		return power;
	}

	private static long factorial( int n ) {
		long fact = 1;
		for(int i=1;i<=n;i++) {
			fact = fact * i;
		}
		return fact;
	}

	public static void main(String[] args) {

		int n= 3;  // Number of values Required in Series
		int x = 2;
		int k= -1;
```

```
                double sum= 0.0 ;

                for (int i = 1; i <= n; i++) {
                        sum =  sum  +   power(x,i-1) / factorial (x+k) ;
                        k++;
                }

                System.out.println("Sum : "+sum);

        }
}
```

35) Lab35.java

```
package com.jlcindia;
/*
35) WAP to evaluate following series.
1/(x-1)! - x/x! + x2/(x+1)! - x3/(x+2)! + x4/(x+3)!

 x = 1 => 1-1+1/2 => 0.5
 x = 2 => 1-1+4/6= > 0.66
 */

/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */
public class Lab35 {
        private static double power( int n , int p) {
                double power = 1;
                for(int i=1;i<=p;i++) {
                        power = power * n;
                }
                return power;
        }
```

```java
        private static long factorial( int n ) {
                long fact = 1;
                for(int i=1;i<=n;i++) {
                        fact = fact * i;
                }
                return fact;
        }

        public static void main(String[] args) {

                int n= 3;  // Number of values Required in Series
                int x = 1;
                int k= -1;
                double sum= 0.0 ;
                int sign = 1;

                for (int i = 1; i <= n; i++) {
                        sum =  sum  +  sign * ( power(x,i-1) / factorial (x+k) );
                        k = k + 1;
                        sign = sign * -1;
                }

                System.out.println("Sum : "+sum);

        }
}
```

36) Lab36.java

```java
package com.jlcindia;
/*
 36) Print the following Pattern
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *


 */

/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
**/
public class Lab36{
        public static void main(String[] args) {

          int stars = 5;
         int rows= 5;

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

         //Task 1: Print Stars
             for (int j = 1; j <= stars; j++) {
                     System.out.print(" * ");
             }

          //Task 2: Print Newline
             System.out.println("");

        }
}
}
```

37) Lab37.java

```java
package com.jlcindia;
/*
37) Print the following Pattern
      *
      * *
      * * *
      * * * *
      * * * * *


*/


/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
**/
public class Lab37{
      public static void main(String[] args) {

        int stars = 1;
        int rows= 5;

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

         //Task 1: Print Stars
            for (int j = 1; j <= stars; j++) {
                    System.out.print(" * " );
                    num++;
            }

         //Task 2: Print Newline
            System.out.println("");

      }
}
}
```

38) Lab38.java

```java
package com.jlcindia;
/*
 38) Print the following Pattern
      *
      * * *
      * * * * *
      * * * * * * *


 */

/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
**/
public class Lab38{
      public static void main(String[] args) {

              int rows= 6;  //Even and Odd Numbers
              int stars = 1;

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

          //Task 1: Print Stars
              for (int k = 1; k<= stars; k++) {
                      System.out.print(" * ");
              }

          //Task 2: Print Newline
              System.out.println("");

              stars = stars + 2;
          }
  }
}
```

39) Lab39.java

```java
package com.jlcindia;
/*
 39) Print the following Pattern
      *
      * * *
      * * * * *
      * * * * * * *
      * * * * *
      * * *
      *


 */

/*
 * @Author : Srinivas Dande
 * @Company: Java Learning Center
 **/
public class Lab39{
      public static void main(String[] args) {

          int stars = 1;
          int rows= 7;  //Odd Numbers

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

          //Task 1: Print Stars
              for (int j = 1; j <= stars; j++) {
                      System.out.print(" * ");
              }

          //Task 2: Print Newline
              System.out.println("");

              stars++;
      }
```

```
        stars = rows/2;

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

                //Task 1: Print Stars
                    for (int j = 1; j <= stars; j++) {
                            System.out.print(" * ");
                    }

                //Task 2: Print Newline
                    System.out.println("");

                    stars--;
            }

}
}
```

40) Lab40.java

```
package com.jlcindia;
/*
 40) Print the following Pattern
            *
          * * *
        * * * * *
       * * * * * * *


 */


/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
**/
public class Lab40{

```

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

            int rows= 6;  //Even and Odd Numbers
            int stars = 1;
            int spaces = 15;

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

            //Task 1: Print Spaces
                for (int j = 1; j <= spaces; j++) {
                        System.out.print("  ");
                }

      //Task 2: Print Stars
            for (int k = 1; k<= stars; k++) {
                    System.out.print(" * ");
            }

            //Task 3: Print Newline
            System.out.println("");

            spaces = spaces - 2;
            stars = stars + 2;

      }

}
}
```

41) Lab41.java

```java
package com.jlcindia;
/*
 41) Print the following Pattern
          *
        * * *
       * * * * *
      * * * * * * *
       * * * * *
        * * *
          *


 */

/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
**/
public class Lab41{

        public static void main(String[] args) {

                int rows= 6;  //Even and Odd Numbers
                int stars = 1;
                int spaces = 25;

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

            //Task 1: Print Spaces
                for (int j = 1; j <= spaces; j++) {
                        System.out.print("  ");
                        }

            //Task 2: Print Stars
                for (int k = 1; k<= stars; k++) {
                        System.out.print(" * ");
                        }
```

```
    //Task 3: Print Newline
        System.out.println("");

        spaces = spaces - 2;
        stars = stars + 2;

    }

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

    //Task 1: Print Spaces
        for (int j = 1; j <= spaces; j++) {
            System.out.print("  ");
            }

    //Task 2: Print Stars
        for (int k = 1; k<= stars; k++) {
            System.out.print(" * ");
        }

    //Task 3: Print Newline
        System.out.println("");

        spaces = spaces + 2;
        stars = stars - 2;

    }

}
}
```