

# MY JAVA NOTES

---

FROM JSPIDER, HEBBAL

**Prepared By :**

***SAGAR SUNAR***

**Contact No. :** +91-7483666261

+977-9864453622(Whatsapp)

**E-mail :-** [sagarsunar202@gmail.com](mailto:sagarsunar202@gmail.com)

**Address :** Thirumenahalli, Yelanhanka, Bangalore-64.



*These notes are completely based on lectures given by respected sir, Mr. Subham K.S. , whose teachings always made new hope in learning programming concepts. Hence, it is for my personal educational purpose and should not be used for any commercial printing or selling purpose.*

*-Thank you!  
SAGAR*



# **MODULE-1(BASICS)**

- Introduction
- JDK/JRE/JVM
- Hello World Program
- Operator
- Control of Statements
- Methods
- Basics of String



**DATE :-** 14-12-2021

SHUBHAM K.S. SIR

### **Basics of Programming:-**

- ➔ A computer is an electronic device which is instructed by the user to do some tasks.
- ➔ An operating system is an interface between the user and computer. It will bring every component of the computer into the existence.  
e.g., Windows, MacOS, Ubuntu, Linux, etc.
- ➔ An application or software is set of programs written by the developers or programmer in order to perform  
Specific tasks.  
E.g., Mx player, skype, whatsapp, calculator, notepad, etc.

### **Types of Application:-**

1. Stand Alone Application(SAA)  
The application which will run without any internet are known as SAA.  
E.g., Calculator, Calendar, Notepad, etc.
2. Client Server Application  
The application which will run with the help of internet are known as Client Server Application.  
E.g., Instagram, Facebook, PUBG, etc.

### **Types of Language in Programming world:-**

1. **High Level Language(HLL):-**  
The language which is understood and readable by the developers or programmer are known as HLL.  
E.g., JAVA, SQL, C, C++, Python, etc.
2. **Assembly Level Language(ALL):-**  
It is also known as middle-level language.  
e.g.,  
move ax,bx  
add cx,bx
3. **Machine level Language(MLL) or Low Level Language(LL) :-**  
The language which is understood by the processor or the language which is in the form of 0's and 1's are called as MLL or LLL.

### **Translator :-**

Since the processor will understand only 0's and 1's , we make use of translator in order to convert high level language or assembly level language to machine level language or low level language.  
e.g., compiler and assembler.

Compilers are used to convert high level language into machine level language.

Assemblers are used to convert assembly level language into machine level language.

- ➔ .type(extension) :-  
Extensions are used to specify the type of file.  
e.g.,  
.txt→text file

.pdf→document  
.png→png  
.exe→c executable file  
.java→java source file  
.py→python source file  
.html→html source file  
.class→java executable file

### Introduction to JAVA :-

The initial name of JAVA was **OAK**. JAVA was introduced in the year 1995 by **James Gosling & Co.** who were working for an organization called **SunMicros System**. Later in the year **2010**, entire SunMicros System was taken over by **Oracle** and from then JAVA is the product of Oracle.

### Difference between C and JAVA:-

C	JAVA
It is platform dependent. It has pointer. It is not simple and secure. It has less libraries. It is procedure-oriented.	It is platform independent Pointer Concept was eliminated. It is simple and secure. It is rich in libraries. It is object-oriented.

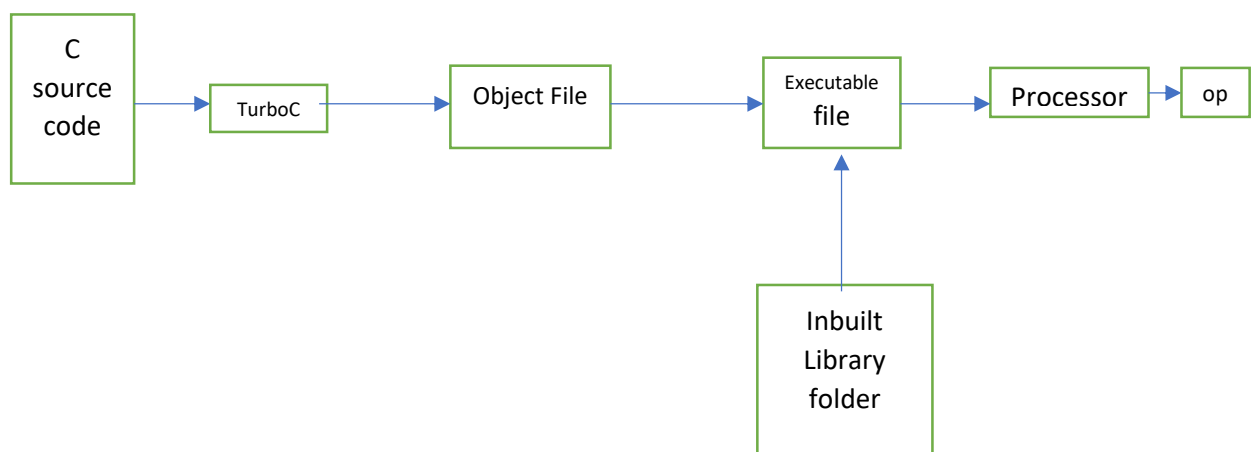
Date :- 15-12-2021

### Properties of JAVA or advantages of JAVA over C :-

1. JAVA is a platform independent programming language.

a. Compilation of C :

C is platform dependent or OS dependent programming language. Code written on one OS cannot be executed on a different operating system



just by doing copy-paste of the executable file. If we want to run the same piece of code on different OS

we need to re-write the code, re-compile it and then execute it.

If we develop an application on Windows OS then that application will run only on Windows not any other OS like MacOS, Ubuntu, Linux, etc.



b. **Compilation of Java :-**

Java is a platform independent or OS independent programming language. The executable file or class file or bytecode file which is generated after successful compilation can be executed on any OS all we need to have the JVM(JAVA Virtual Machine).

**JVM(Java Virtual Machine) :-**

It will perform the task like a machine without having any physical existence. The task of the machine is to process a JAVA executable file to the processor because a java executable file cannot be given directly to the processor.

**JRE(Java Runtime Environment) :-**

It is responsible for providing necessary environment to the JVM so that it can perform its task without any interruption.

**JIT(Just In Time) & Interpreter :-**

Along with JavaC, these two are additional compilers which are present inside the **JVM**. They are responsible for line by line compilation of a java executable file and for changing the configuration of Java executable file.

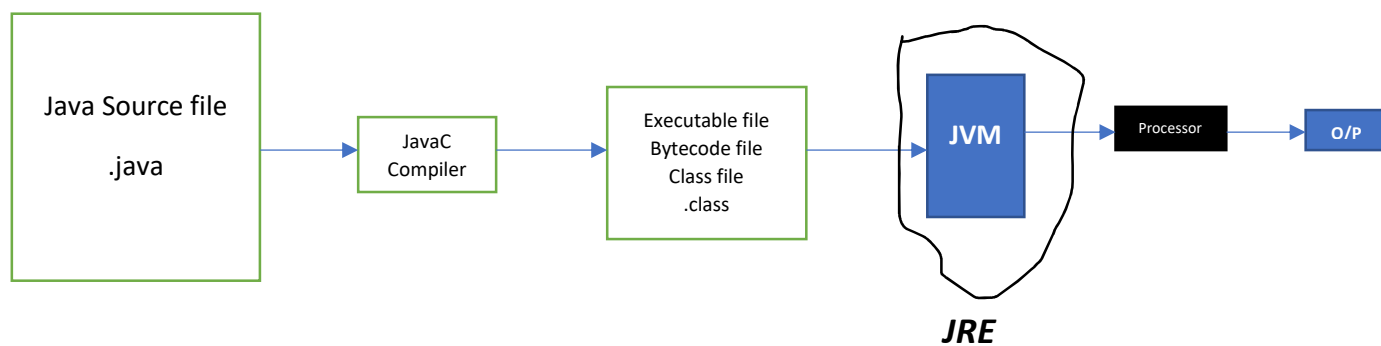
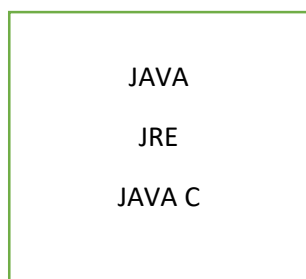


Fig. Compilation of Java

**JDK(Java Development Kit) :-**

It is responsible for providing all the necessary components that are required to compile and execute a java program.



**[Important note :-** In java, only the executable file or bytecode file or class file which is generated after successful compilation is platform independent whereas all the other components of Java like JavaC, JVM , JRE are platform dependent.]

2. In java, the pointer concept was eliminated from the user perspective but it was implemented implicitly with the help of **JVM**.

Using pointers, we can directly communicate with the processor. That's why, the performance of **C** is better when compared to any other programming language.

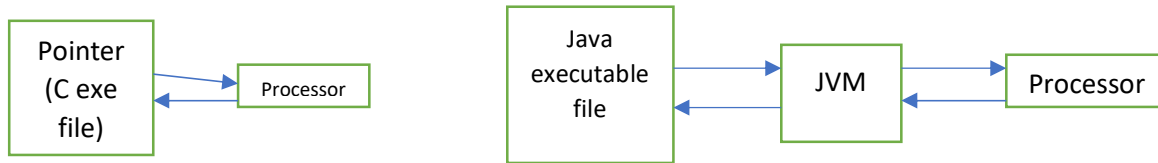


Fig. execution of **C & Java** file

3. Java is object-oriented programming language.

Any programming language that supports the concept of object, class, inheritance, encapsulation, polymorphism and abstraction are known as **object-oriented programming language**.

**C** is procedure-oriented programming language.

4. Java is simple programming language.

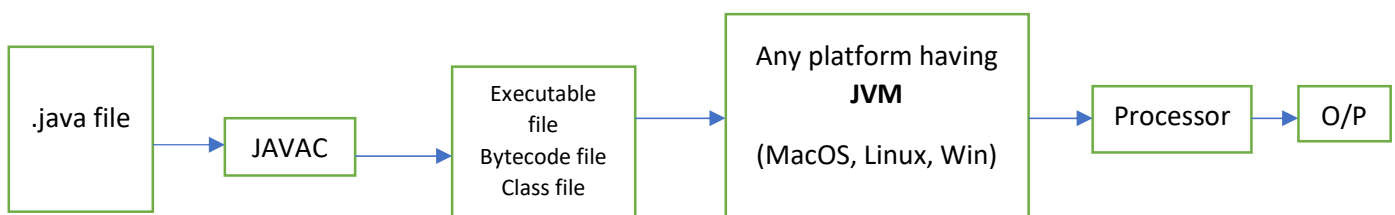
5. Java is secure programming language.

**Date :-** 17-12-2021

Friday

→ **WORA principle :-**

In JAVA, the executable file or bytecode file or class file(.class) which is generated after successful compilation can be executed on any operating system; all we need to have the **JVM**. This property of the java executable file to be generated only once and to be executed on any platform or OS for any number of times is known as **WORA(Write Once Run Anytime) principle**.



→ **'+' Operator :**

In java, the + operator has dual functionality.

- To perform addition.
- To perform concatenation.

Anything represented in single quote(' ') is known as character. In single quote, we can write only one character.

Anything represented in double quotes(" ") is known as string. In double quotes, we can write 0 or any number of characters.

The '+' operator will perform addition when,

- Both the values are number,
- Both values are characters,
- One value is number and other value is character.

The '+' operator will perform concatenation when any one of the values is of type string.

Every Character present in the keyboard has special number associated with it which are technically called as **ASCII** value. A given character will convert to its **ASCII** value only when the other value is of type: number or character.

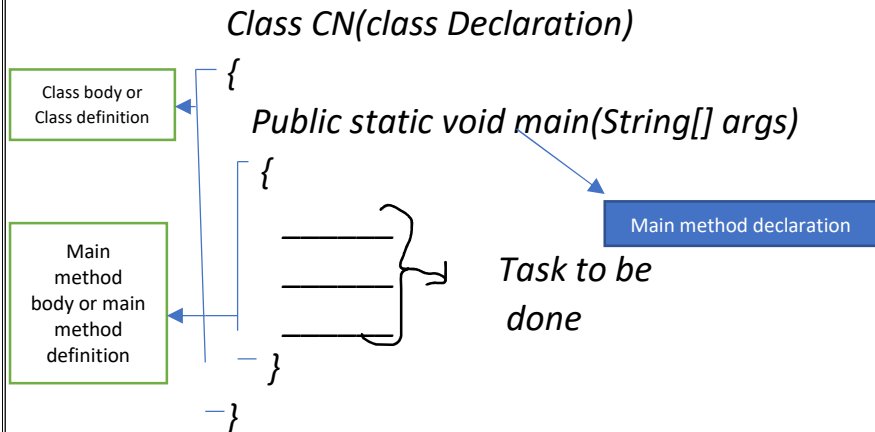
A character will not convert to its **ASCII**(American Standard Code for Information Interchange) value if the other value is of type string.

**A-B → 65-90    a-b → 97-122**

**0-9 → 48-57**

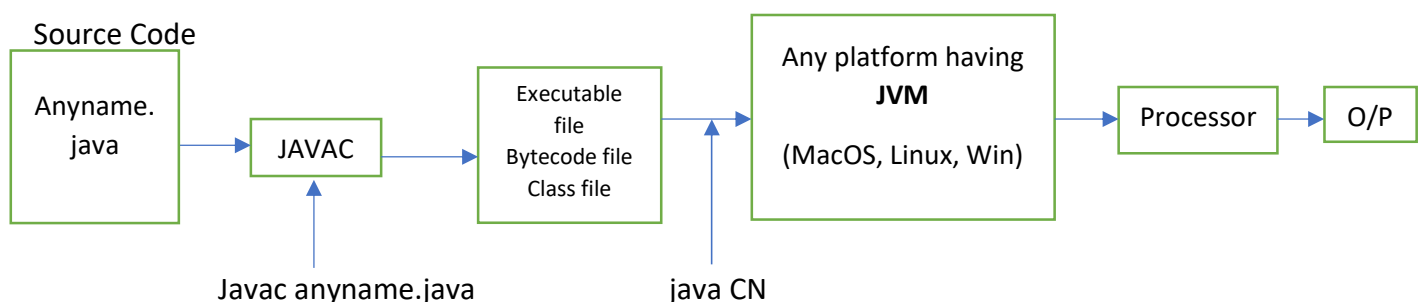
10+20→30	'A'+10 = 75	'2'+ 'A' = 115
30+70→100	10+ 'B' = 75	'1'+ "Hello" = 1Hello
10.1+10→20.1	'A'+ 'B'=131	'10' + 20 = error
"Hello"+10= Hello10	"Hello" + 'A' + 'B' = HelloAB	
"Hello" + "World"= HelloWorld	'A' + 'B' + "Hello" = 131Hello	
"Hello" + 'A' = HelloA	'AB'+10 = error	
"Hello" + 10+20 = Hello1020	1+10 = 11	
10+20+ "Hello" = 30Hello	'1' + 10 = 59	

### → Structure of Java Program :-



### → Execution steps of the java programming :-

1. SAVE :- anyname.java
2. COMPILE :- javac anyname.java
3. EXECUTE :- java CN(ClassName)



[ **Important Note :-** A **java source file** can be with any name but the executable file which is generated after successful compilation will have the same name as that of the **class name**. That's why, in order to avoid confusion, it's recommended to save a java source file with the same name as that of **Class name**. ]

**Date :-** 21-12-2021

- **Different print statements in java :-**

1. `System.out.print();` #print and stay on same line

e.g.

`System.out.print("Hello");`

2. `System.out.println();` #print and move to next line

e.g.,

`System.out.println("Hello");`

**NOTE :-**

1. `\n` :- newline/nextline
2. `\t` :- 1 tab(4 spaces)
3. `//` :- Single line command
4. `/*`  
---- → Multi Line command  
---  
`*/`

→ **Types of Error in JAVA :-**

Anything that stops the user from executing the program is known as **error**.

In java, we have two types of error:

- a) Compile Time Error
- b) Run Time Error

If we get any error during compilation of the program, it is called as **Compile Time Error(CTE)**.

Compilation Step:

`Javac sourcefilename.java`

e.g., ; missing  
} missing

If we get any error during execution of the program, it is called as **Run Time Error(RTE)**.

Execution Step:

`Java classname`

e.g., classfile not found  
main method not found

"JAVAC" is responsible for compile time error and "JVM" is responsible for run time error.

## ➔ Example Programs :-

### Example1:-

Source code:-

*Class Demo1*

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

C:\Users\SAGAR> B:

B:\>cd JavaLearning\

B:\JavaLearning\>cd Basics\

B:\JavaLearning\Basics\>

### Example2:-

Source Code:

*Class Demo2*

```
{  
    public static void main(String[] args)  
    {  
        System.out.println("Hello" + "World");  
        System.out.println("Hello"+10+20);  
        System.out.println(10+20+ "Hello");  
        System.out.println("Hello" + 'A');  
        System.out.println('A'+10);  
        System.out.println('A' + 'B');  
        System.out.println("A" + 10);  
    }  
}
```

### Example3:-

Source Code:

*Class Sample1*

```
{  
    Public static void main(String[] args)  
    {  
        System.out.println("Hello");  
        System.out.print("welcome");  
        System.out.println("to");  
        System.out.print("java");  
        System.out.print("classes by");  
        System.out.println("Dinga Kumar!!");  
    }  
}
```

### Execution part:-

Javac Sample1.java

Java Sample1

### **OUTPUT:-**

Hello  
Welcometo  
Javaclasses byDinga Kumar!!

[Note:- cd .. → one folder back]

**Date :- 22-12-2021**

### **→ KEYWORDS, IDENTIFIER & VARIABLES :-**

**Keywords** are set of predefined words or reserved words for which already some meaning is defined by the

Java developers.

In java, there are 52 keywords.

Every keyword must be represented in lowercase.

e.g., class, public, void, static, etc.

**Identifiers** are used to identify the elements of java(class & method) . In other words, the names that we

give to a class or a method in order to identify them uniquely are known as identifier.

**Rules to be followed while defining an identifier..**

1. We can use alphanumeric(only alphabets or combination of both alphabets & numbers) for identifier.
2. We can use both uppercase and lowercase alphabets for identifiers.
3. An identifier should always begin with an alphabet and later it can have any digits of number.
4. Space is not allowed for identifier.
5. The only special character allowed for identifiers are '\_' and '\$'.

### **Bit, nibble & byte :-**

1. 0/1 → 1 bit
2. 4 bits → 1 nibble
3. 8 bits→2 nibble→1 byte
4. 1024 byte(B) → 1 kB
5. 1024 kB →1 MB
6. 1024 MB →1 GB
7. 1024 GB→ 1 TB
8. 1024 TB→1 PB(peta byte)

**e.g.,**

Store 39 in 1 byte

[0|0|1|0|0|1|1|1|] -1 byte

39→100111

Store 'D' in 3 nibbles

[0|0|0|0|0|1|0|0|0|1|0|] 0|

D→68→1000100

Store 43 in 1 byte

|0|0|1|0|1|0|1|1|

43→101011

Store 'c' in 2 byte

|0|0|0|0|0|0|0|0|0|0|1|0|0|0|1|1|

c→99→100011

## Variables :-

Variables are identifiers in which we store some value.

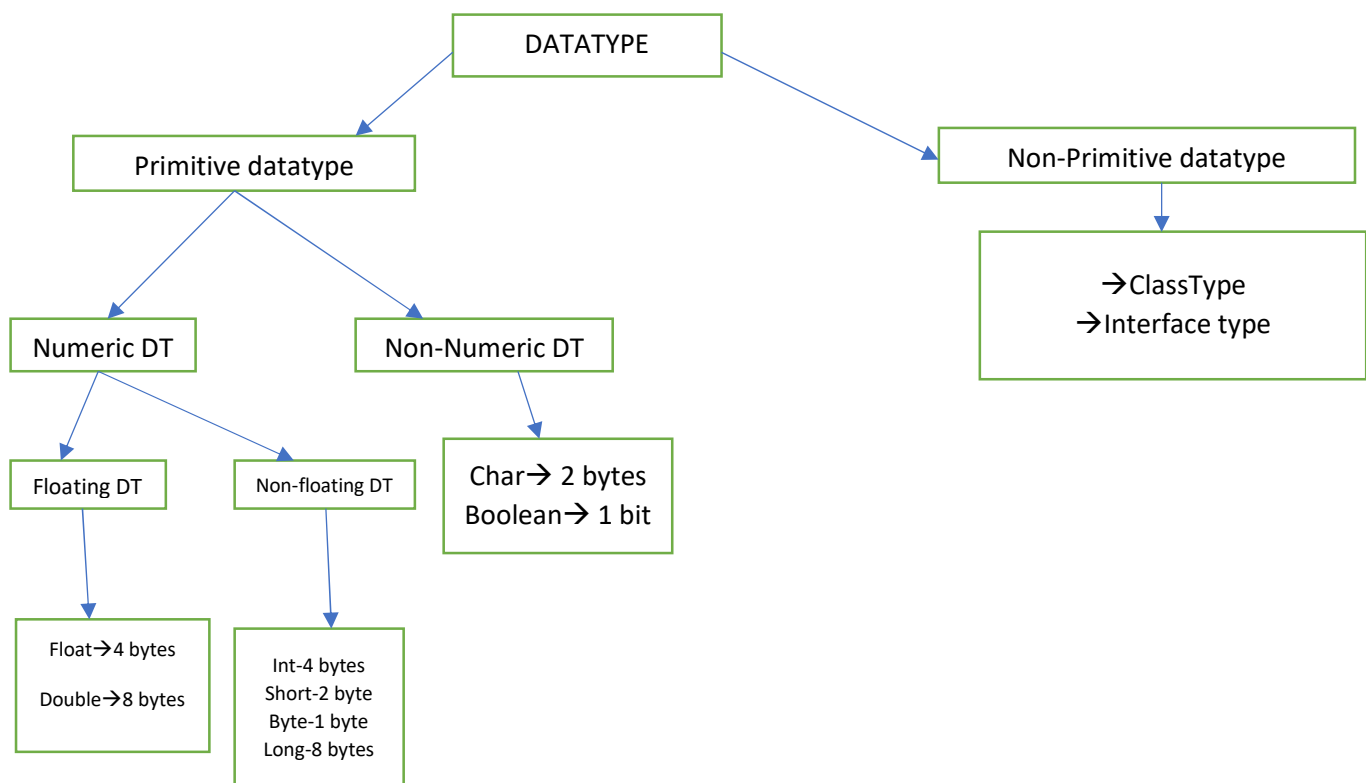
The type of the value & the length of the value to be stored inside a variable will be decided by **data types**.

Datatypes will be used along with the variable name during variable declaration process.

[\*Important notes :- All the datatypes are keywords.

In a Boolean variable, we can store either true or false not 1 or 0.

In java, string is not a datatype rather it is a predefined class which is available in in-built Library folder.]



## → Syntax for variable declaration :-

**//Variable declaration**

Syntax:

Datatype variable name;

e.g.;

int x;

**//Variable declaration & initialization**

Syntax:-

datatype variablename=value;

e.g.,

int x=123;



double num;  
Boolean var;

double num=10.12;  
Boolean var=true;

### //Variable initialization

Syntax :-

Variablename=value;

e.g.,

x=123;

num=10.12;

var=true;

**DATE** :- 23-12-2021

### **Example 1:**

```
class Program4
{
    public static void main(String[] args)
    {
        int a = 100;
        System.out.println("Hello World!");
        System.out.println(a+10);
        System.out.println("a="+a);
        System.out.println('a'+a);
        System.out.println(a+"a");
        System.out.println("Hello"+a);
        System.out.println(a+a);
    }
}
```

### **Output:**

Hello World!

110

a=100

197

100a

Hello100

200

### **Example 2:**

```
class Program5
{
    public static void main(String[] args)
    {
        char x='A';
        System.out.println("Hello"+x);
        System.out.println("x="+x);
        System.out.println(x+'x');
        System.out.println(x+"x");
        System.out.println(x+x);
        System.out.println(x+10);
    }
}
```



```
}  
}
```

**Output:**

```
HelloA  
x=A  
185  
Ax  
130  
75
```

**Example 3:**

```
class Program6  
{  
    public static void main(String[] args)  
    {  
        //variable declaration  
        int var1;  
        double var2;  
        char var3;  
        boolean var4;  
  
        //variable initialization  
        var1=1234;  
        var2=120.35;  
        var3='S';  
        var4=false;  
        System.out.println("var1="+var1);  
        System.out.println("var2="+var2);  
        System.out.println("var3="+var3);  
        System.out.println("var4="+var4);  
    }  
}
```

**OUTPUT:-**

```
var1=1234  
var2=120.35  
var3=S  
var4=false
```

➔ **Important Conclusions on Variable:-**

1. **Conclusion 1 :-**

Within the main method, we can have two variables of the same type but we cannot have two variables with the same name. If at all, we try to declare two variables with the same name, then while declaring the second variable, we will get **compile time error** as shown below:

```
Class A  
{  
    P S V main(-)  
    {  
        int a =100;  
        int b=200;  
        sop(a);  
        sop(b);  
    }  
}
```

```
Class B  
{  
    P S V main(-)  
    {  
        int a=100;  
        int a=200;  
        sop(a);  
    }  
}
```

```
Class C  
{  
    P S V main(-)  
    {  
        int a =100;  
        double a=10.12;  
        sop(a);  
    }  
}
```

## 2. Conclusion 2 :-

A variable declared inside a main method cannot be used without being initialized. If at all, we try to use a variable without assigning any values to it, we will get compile time error as shown below:

*Class Demo3*

```
{
    Public static void main(String[] args)
    {
        Int x;
        System.out.println("x="+x);
    }
}
```

### **Error message:-**

*Demo3.java:7: error: variable x might not have been initialized*

*System.out.println("x="+x);*

*1 error*

## 3. Conclusion 3 :-

In order to differentiate between a **double** value & **float** value, it is mandatory to write "**f**" beside the float value.

### **Example:**

a. *float a = 10.12f;*  
*sop(a); //10.12*

c. *double x = 10.12;*  
*sop(x); //10.12*

b. *float b = 123.12f;*  
*sop(b); //123.12*

d. *double y = 123.12;*  
*sop(y); //123.12*

## 4. Conclusion 4 :-

Whenever we are declaring multiple of the same type instead of having multiple declaration statement, we can declare all the variables in a single statement as shown in the below example:

### **Example1:**

-----

```
int x;
int y; → int x,y,z;
int z;
```

### **Example2:**

-----

```
char x;
char y; → char x,y,z;
char z;
```

## 5. Conclusion 5 :-

Assigning values to a variable which is already having some value is known as **re-initialization of a variable**.

Re-initializing a variable will not affect the previous statement rather it will affect the upcoming statements.

```

class A
{
    public static void main(String [] args)
    {
        int a = 10;
        sop(a); //10
        sop(a+10); //20
        sop(a-10); //0
        a = 100; //re-initialization
        sop(a); //100
        sop(a+10); //110
        sop(a-10); //90
    }
}

```

**Date :- 24-12-2021**

#### 6. Conclusion 6 :-

If we declare a variable as final, we cannot re-initialize it or re-assign values to it. A final variable can be assigned value only once through its lifetime. If at all, we try to re-initialize a final variable, then we will get compile time error as shown below:

```

class Program11
{
    public static void main(String[] args)
    {
        final int x=100; // variable declaration and initialization
        System.out.println("x="+x);
        x=200; //re-initialization
        System.out.println("x="+x);
    }
}

```

#### Command Prompt:

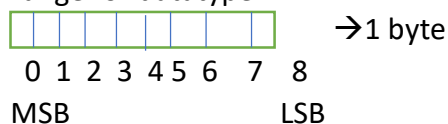
```

Program11.java:7: error: cannot assign a value to final variable x
    x=200; //re-initialization
    ^
1 error

```

#### 7. Conclusion 7 :-

Range for datatype:



MSB-Most Significant Bit      LSB- Least Significant bit

Byte → 1 byte →  $-2^7 \dots 2^7 - 1 = -128$  to  $127$

Short → 2 bytes →  $-2^{15} \dots 2^{15} - 1 = -32768$  to  $32767$

Int → 4 bytes →  $-2^{31} \dots 2^{31} - 1$

Long → 8 bytes →  $-2^{63} \dots 2^{63} - 1$

#### → Interview Questions :-

**Write a program to swap two numbers using a temporary variable or a dummy variable.**

```

class Program12
{
    public static void main(String[] args)

```

```

{
    int x=200, y=300;
    int temp;
    System.out.println("Before Swapping");
    System.out.println("-----");
    System.out.println("x="+x); // x= 200
    System.out.println("y="+y); // y=300

    //logic
    temp=x;
    x = y;
    y = temp;

    System.out.println("After Swapping");
    System.out.println("-----");
    System.out.println("x="+x); // x=300
    System.out.println("y="+y); // y=200
}
}

```

### **Output:-**

Before Swapping

-----

x=200

y=300

After Swapping

-----

x=300

y=200

### **//Logic:**

100  
→temp=x;

200  
→x=y;

100  
→y=x;

**Write a program to swap 2 number without using a temporary number.**

```

class Program12
{
    public static void main(String[] args)
    {
        int x=200, y=300;
        System.out.println("Before Swapping");
        System.out.println("-----");
        System.out.println("x="+x); // x= 200
        System.out.println("y="+y); // y=300

        //logic
        x= x+y;
        y=x-y;
        x=x-y;

        System.out.println("After Swapping");
    }
}

```

```

        System.out.println("-----");
        System.out.println("x="+x); // x=300
        System.out.println("y="+y); // y=200
    }
}

```

### **OUTPUT :-**

Before Swapping

-----

x=200

y=300

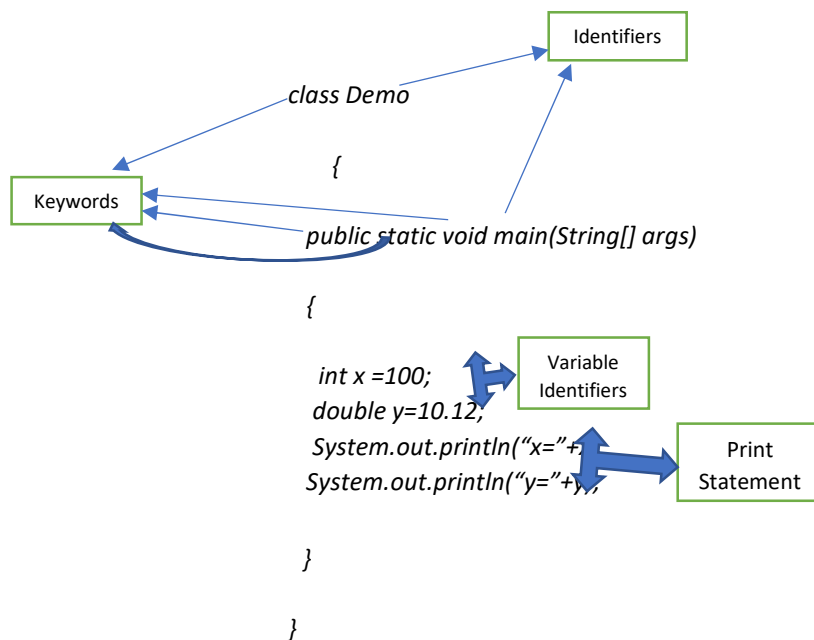
After Swapping

-----

x=300

y=200

### **Important Notes:**



➔ **Conversion from number to binary & binary to number :-**

**Number to Binary**

2|43  
2|21-1  
2|10-1  
2|5-0  
2|2-1  
1-0

$$43_{10} = 101011_2$$

**Binary to Number**

$$\begin{aligned} &101011 \\ &= 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 \\ &= 43 \end{aligned}$$

$$\begin{aligned} 47 &= 101111 \\ 86 &= 1010110 \\ 103 &= 1100111 \end{aligned}$$

**DATE :- 27-12-2021**

➔ **OPERATOR :-**

1. Operators are used to perform operations on **operands**.
2. The values on which we perform the operations are technically known as **operands**.

**Types of Operators :-**

- i. Arithmetic Operator( +, -, \*, /, %)
- ii. Relational Operator (>, <, <=, >=, ==, !=)
- iii. Logical Operator (&&, ||)
- iv. Bitwise Operator (&, |)
- v. Unary Operator (++ , --)
- vi. Shifting operator (>>, <<)

**i. Arithmetic operator :-**

They are used to perform mathematical operations on the given values.

If multiple operators are there in a single expression , the JVM will follow BODMAS rule.

B → bracket { }

O → of()

D → division

M → Multiplication

A → Addition

S → Subtraction

**Types:-**

+ → Sum

- → Difference  
\* → Product  
/ → Quotient  
% → Remainder

1.  $2+2/2 = 3$
2.  $2*2+2 = 6$
3.  $2/2\%2 = 1$
4.  $2-2*2/2 = 0$
5. "Hello" + 10 + 20 = Hello1020
6. "Hello" + (10+20) = Hello30

#### Example Program :-

```
class Arithmetic
{
    public static void main(String[] args)
    {
        int a = 100, b = 25;
        System.out.println("Sum = " + (a+b));
        System.out.println("Difference = " + (a-b));
        System.out.println("Product = " + (a*b));
        System.out.println("Quotient = " + (a/b));
        System.out.println("Remainder = " + (a%b));
    }
}
```

#### Output :-

Sum = 125  
Difference = 75  
Product = 2500  
Quotient = 4  
Remainder = 0

#### ii. Relational Operator :-

They are used to compare two values.

Every relational operator will return either true or false based on the comparison.

##### Types:

> = greater than  
< = less than  
>= greater than or equal to  
<= less than or equal to  
== equal to  
!= not equal to

##### Example 1 :

10 > 10 => false  
10 <= 10 => true  
10 < 10 => false  
10 >= 10 => true  
10 != 10 => false

Int a=10,b=20,c=10;  
Sop(a>b); //false  
Sop(b<=c); //true  
Sop(a==c); //true  
Sop(a!=b); //true  
Sop(a!=c); //false

#### iii. Logical Operator :-

They are used to compare two conditions.

Every logical operator will return either true or false based on the comparison.

##### Types :-

&&

## ||

### Working of “&&”

C1	&&	C2	O/P
T		T	T
T		F	F
F		T	F
F		F	F

*Output will true only if all the conditions are true.*

### Working of ||

C1		C2	O/P
T		T	T
T		F	T
F		T	T
F		F	F

*Output will true if any conditions are true.*



### Example Program :-

```
class Logical
{
    public static void main(String[] args)
    {
        System.out.println(10>10); //true
        System.out.println(10>100); //false

        System.out.println(10==100 && 10<=11); //false
        System.out.println(10>=10 && 10!=100); //true
        System.out.println(100>=10 || 10!=10); //true
        System.out.println(10>10 || 100!=100); // false

        int a =10,b=20,c=10; //variable initialization and declaration
        System.out.println(a=c); //true
        System.out.println(a>=b); //false
        System.out.println(a>=c && c<b); //true
        System.out.println(a==b && b==c); //false
        System.out.println(a==c || b==c); //false
    }
}
```

### OUTPUT :-

```
false
false
false
true
true
false
10
false
true
false
true
```

#### iv. Bitwise Operator (Imp) :-

##### Types :-

& and |

##### Example:-

Int a =49;

Int b = 47;

System.out.println(a & b); **// 33**

System.out.println( a | b); **// 63**

a = 49 → 110001

b = 47 → 101111

1 1 0 0 0 1

& & & & &

1 0 1 1 1 1

---

**100001 → 33**

```

1 1 0 0 0 1
| | | | |
1 0 1 1 1 1
-----
1 1 1 1 1 1 → 63

```

### → ASSIGNMENT :-

1. `int a = 37;`  
`Int b = 39;`  
`Sop(a&b); //37`  
`Sop(a|b); //39`
2. `int x= 96;`  
`Int y = 99;`  
`Sop(x&y); //96`  
`Sop(x|y); //99`
3. `char x= 'B';`  
`Char y = ' Z' ;`  
`Sop(x&y); //66`  
`Sop(x|y); //90`
4. `int a = 79;`  
`Int b = 81;`  
`Sop(a&b); //65`  
`Sop(a|b); //95`
5. `char X = 'a';`  
`Char Y = 'c';`  
`Sop(x&y); //97`  
`Sop(x|y); //99`
6. `char a = '2';`  
`Char b = '9';`  
`Sop(a&b); //48`  
`Sop(a|y); //59`

**Date :- 28-12-2021**

### → Unary Operator :-

**Types:**

`++(INCREMENT)`

→ `PRE++`

→ `POST++`

`--(DECREMENT)`

→ `PRE--`

→ `POST--`

```
1. int a = 10;
   int b=a;
   Sop(a); //10
   Sop(b); //10
```

```
2. POST increment(++):
   int a = 10;
   int b=a++;
   Sop(a); //11
   Sop(b); //10
```

```
3. PRE increment(++):-
   int a = 10;
   int b=++a;
   Sop(a); // 11
   Sop(b); //11
```

```
4. Post Increment(--):-
   int a = 10;
   int b=a--;
   Sop(a); //9
   Sop(b); //10
```

```
5. Pre Increment(--):-
   int a = 10;
   int b=--a;
   Sop(a); //9
   Sop(b); //9
```

### ***Interview Questions on Unary Operation :-***

```
int x=20;
   Sop(x++ + ++x - x-- --x); //0
      20  22 - 22 - 20
int      x = 10;
System.out.println(x++ - x-- + ++x + x-- - ++x); //10
```

```
int a=10;
int b = ++a + a++ - --a - a-- + a-- + ++a - ++a - a++;
```

```
int a =50;
int b =100;
int c = ++a + a++ - --b - a-- + b-- + ++a - ++b - a++ + --b - --a; //-3
```

## → Shifting Operator :-

### Right Shift :-

>>

Byte x = 14;

System.out.println(x>>1); //7

### Left Shift :-

<<

Byte a = 23;

System.out.println(a<<3); // 120

**Date :-** 29-12-2021

*[Important Note:*

int x =10; (same) int x =10;

x = x\*3;  x \*= 3;

Sop(x); Sop(x);

int x = 10; (same) int x = 10;

x = x+10;  x += 10;

Sop(x); Sop(x);

int x =10; (same) int x =10;

x = x%3;  x %= 3;

Sop(x); Sop(x);

]

### Example Program :-

```
class Uni
```

```
{
```

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

```
    {
```

```
        int x =10;
```

```
        x += 10; // x=x+10;
```

```
        System.out.println("x = "+x);
```

```
        int y = 20;
```

```
        y *= 4; // y=y*4;
```

```
        System.out.println("y = " + y);
```

```
        int z = 50;
```

```
        z /=4; // z=z/4;
```

```
        System.out.println("z = " + z);
```

```
        int a=100;
```

```
        a -=99; // a=a-4;
```

```
        System.out.println("a = " + a);
```

```
        int b = 44;
```

```
        b%=10; // b = b%10;
```

```
        System.out.println("b = " + b);
```

```
    }
```

```
}
```

**OUTPUT :-**

x = 20

y = 80

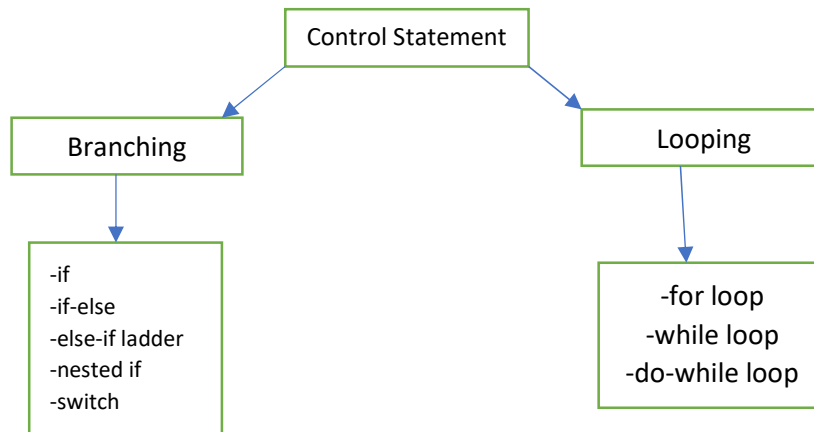
z = 12

a = 1

b = 4

## → CONTROL STATEMENT :-

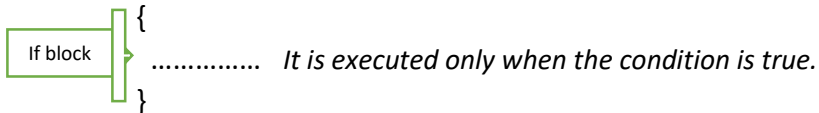
Control statements are used by the programmer to control the flow of execution of the program.



## → Branching :-

### 1. if :-

if(condition)



### Example Program:-

```
class IfExample
{
    public static void main(String[] args)
    {
        System.out.println("main starts");

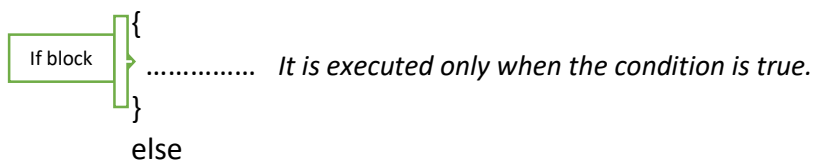
        if (100<=123)
        {
            System.out.println("if block executed successfully");
        }
        System.out.println("main ends");
    }
}
```

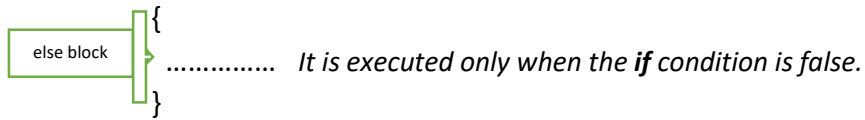
### OUTPUT :-

```
main starts
if block executed successfully
main ends
```

### 2. if- else :-

if(condition)





### Program Example :-

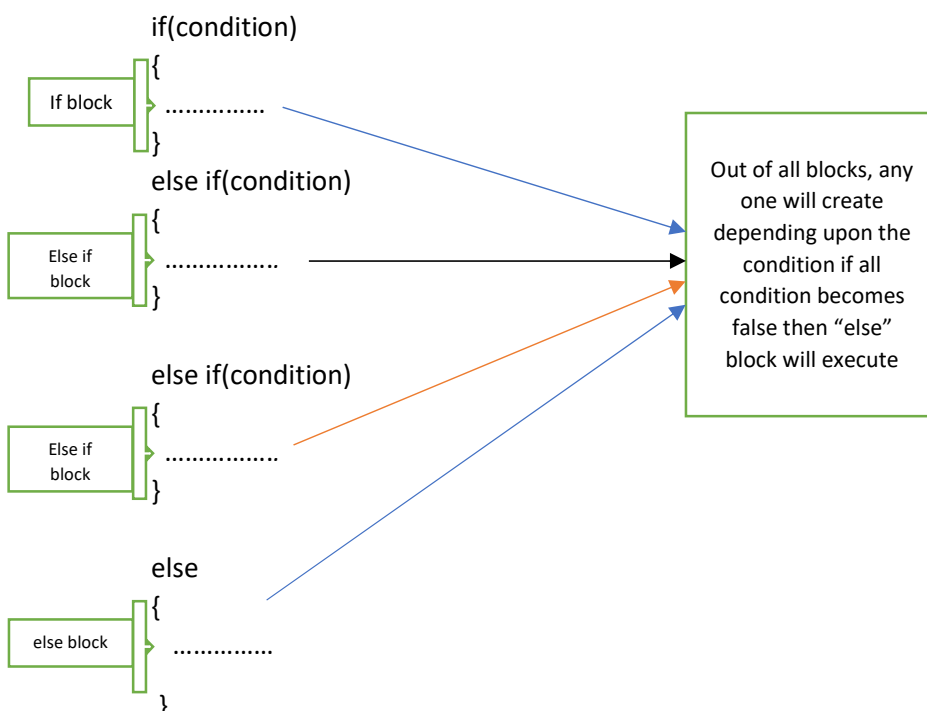
```
class IfElse
{
    public static void main(String[] args)
    {
        int marks=96;
        System.out.println("Welcome Dinga");

        if (marks>=35)
        {
            System.out.println("Dinga Rocks");
        }
        else
        {
            System.out.println("Dinga Shocks");
        }
        System.out.println("Bye Dinga");
    }
}
```

### OUTPUT :-

Welcome Dinga  
Dinga Rocks  
Bye Dinga

### 2. else-if ladder :-



### Example Program :-

```
class ElselfLadder
{
    public static void main(String[] args)
    {
        char signal ='o';

        if (signal=='r')
        {
            System.out.println("Rukk Jaa!!");
        }
        else if (signal=='y')

        {
            System.out.println("Tayaar Ho Jaawo");
        }
        else if (signal=='g')
        {
            System.out.println("Peheli Fursad Mein Nikal!!");
        }
        else
        {
            System.out.println("Jo karna haai karle");
        }
        System.out.println("Sukriya!! Phirr milna jarur");
    }
}
```

### OUTPUT :-

Jo karna haai karle  
Sukriya!! Phirr milna jarur

### CODE SNIPPET:

1. if(10==10 || 10<10)

{

--

--

}

else

{

--

--

}

2. if(10>=10 && 10 <10)

{

--

--

}

else

{

--

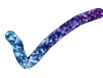
--


}


3. if (false || 10==10)


{




```
--
-- 
--
-- }
-- else
-- {
--
--
-- }
```

```
4. if(true && 10==10 && false)
{
--
--
-- }
else
{
--
-- 
--
-- }
```

```
5. if(10==10)
{
--
-- 
--
-- }
else if(true && false)
{
--
--
-- }
else
{
--
--
-- }
```

```
6. if(true && false)
{
--
--
-- }
else if (false || 10<10)
{
--
--
-- }
else
{
--
-- 
--
-- }
```

```
7. if(10==10 || false)
{
--
-- 
--
-- }
else if(true && 11==11)
{
--
```

```
--
}
else if(true)
{
--
--
}
else
{
--
--
}
```

### **ASSIGNMENT:-**

**Write program to check if a given number is even or odd.**

```
class Oe
{
    public static void main(String[] args)
    {
        int x=10;

        if(x % 2 == 0)
        {
            System.out.println("Number is even");
        }
        else
        {
            System.out.println("Number is Odd");
        }

        System.out.println("Thank you!");
    }
}
```

**Write a program to check if a given number is positive or negative.**

```
class PN
{
    public static void main(String[] args)
    {
        int n=10;
        if (n > 0)
        {
            System.out.println("Number is positive");
        }
        else if (n<0)
        {
            System.out.println("Number is Negative");
        }
        else if (n==0)
        {
            System.out.println("Number is Null");
        }
        else
        {
            System.out.println("Enter the valid data");
        }

        System.out.println("Thank you!");
    }
}
```

```
}  
}
```

**Write a program to find the greatest of two number.**

```
class GN  
{  
    public static void main(String[] args)  
    {  
        int a=20;  
        int b=30;  
        if (a > b)  
        {  
            System.out.println("A is greater than B");  
        }  
        else if (b > a)  
        {  
            System.out.println("B is greater than A");  
        }  
        else  
        {  
            System.out.println("Both are equal");  
        }  
        System.out.println("Thank you!!");  
    }  
}
```

**Write a program to check if a student is eligible for exam or not(if attendance percentage is greater than or equal to 75, student is eligible else not eligible.)**

```
class Exam  
{  
    public static void main(String[] args)  
    {  
        int a = 74;  
        if (a<75)  
        {  
            System.out.println("Sorry, you're not eligible for exam.");  
        }  
        else  
        {  
            System.out.println("Well, good luck for the exam.");  
        }  
        System.out.println("Thank you!");  
    }  
}
```

**Date :- 31-12-2021**

**Write a program to take student's mark and display their result.**

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

```

    {
        double marks = 64.58;
        if (marks >=75 && marks <=100)
        {
            System.out.println("First class with Distinction");
        }
        else if (marks >=60 && marks <75)
        {
            System.out.println("First class!!");
        }
        else if (marks>=50 && marks < 60 )
        {
            System.out.println("Second Class!!");
        }
        else if (marks >= 35 && marks<50)
        {
            System.out.println("Third Class!");
        }
        else if (marks>=0 && marks < 35)
        {
            System.out.println("fail");
        }
        else
        {
            System.out.println("Invalid Marks");
        }
        System.out.println("Thank You!");
    }
}

```

#### OUTPUT :-

First class!!  
Thank You!

#### Write a program to check if a given character is vowel or constant.

```

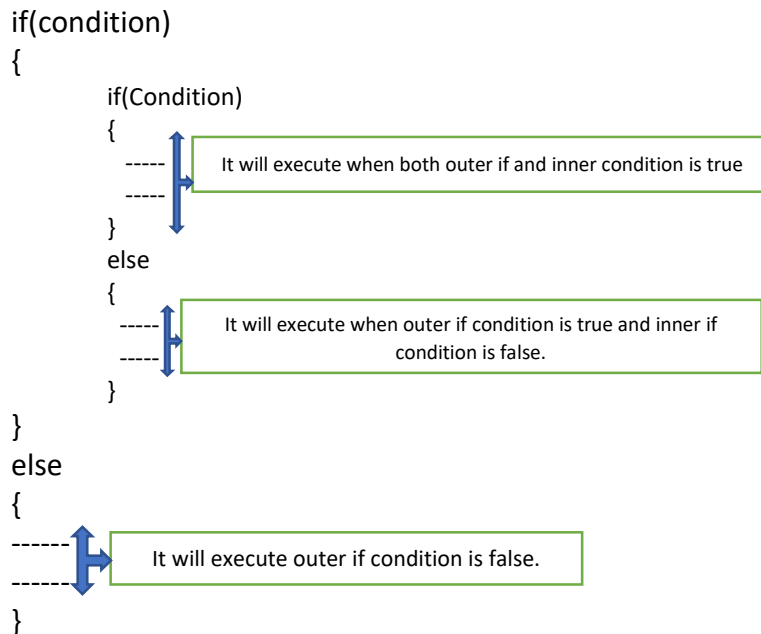
class Vowel
{
    public static void main(String[] args)
    {
        char x='P';
        if (x=='a' || x=='e' || x=='i' || x=='o' || x=='u' || x=='A' || x=='E' || x=='I' || x=='O' ||
x=='U')
        {
            System.out.println(x+" is a vowel");
        }
        else
        {
            System.out.println(x+" is a consonant");
        }
        System.out.println("Thank you!");
    }
}

```

#### OUTPUT :-

P is a consonant  
Thank you!

## → Nested if :-



## Example Program:-

```
class AtmMachine
{
    public static void main(String[] args)
    {
        int dbpin=1234;
        int dbamt = 5000;

        int pin = 1234;
        if(pin==dbpin)
        {
            int amt = 3400;
            if (amt<dbamt)
            {
                System.out.println("Transaction Successful!!");
                System.out.println("Available Balance="+dbamt-amt);
            }
            else
            {
                System.out.println("Insufficient Fund");
            }
        }
        else
        {
            System.out.println("Invalid Pin");
        }
        System.out.println("Thank you for using ATM!!");
    }
}
```

## OUTPUT :-

```
Transaction Successful!!
Available Balance=1600
Thank you for using ATM!!
```

Write a program to check if a given number is positive or negative. If the number is negative, print ORANGE | | if the number is positive or equal to zero, perform the below operation:

a. print MANGO if the number is even.

b. print PAPAYA if the number is odd.

```
class Positive
{
    public static void main(String[] args)
    {
        int n = 0;
        if (n < 0)
        {
            System.out.println("ORANGE");
        }
        else if (n >= 0)
        {
            if (n % 2 == 0)
            {
                System.out.println("MANGO");
            }
            else
            {
                System.out.println("PAPAYA");
            }
        }
        System.out.println("Thank you!");
    }
}
```

#### OUTPUT :-

MANGO  
Thank you!

Write a program to check if an employee is getting salary more than 5000 or not. If the employee is getting salary less than 5000, Display NO INCREMENT THIS YEAR

If the employee is getting salary more than or equal to 5000, Perform the below operation:

a. if employee gender is 'M', increase the salary by 2000.

b. if employee gender is 'F', increase the salary by 3000.

```
class SalEmp
{
    public static void main(String[] args)
    {
        int sal = 5400;
        if (sal < 5000)
        {
            System.out.println("NO INCREMENT THIS YEAR");
        }
        else
        {
            char g = 'F';
            System.out.println("ENJOY THE INCREMENT!!");
        }
    }
}
```

```

        if (g=='M')
        {
            System.out.println("NEW SALARY = " +(sal+2000));
        }
        else
        {
            System.out.println("NEW SALARY = " +(sal+3000));
        }
    }
}

```

### OUTPUT :-

ENJOY THE INCREMENT!!  
SALARY = 8400

## → ASSIGNMENT :-

→ Write a program to perform the below operation .  
If age is more than or equal to 18 then display “you can vote”  
Else display wait “wait for difference years”

```

class Vote
{
    public static void main(String[] args)
    {
        int age = 17;
        if (age >=18)
        {
            System.out.println("You can vote");
        }
        else
        {
            System.out.println("Wait for " +(18-age) +" years");
        }
    }
}

```

### OUTPUT :-

Wait for 1 years

→ Write a program to perform the below operation.  
If mark is greater than or equal to 75 then display “You can attend interview”  
Else display “you can’t attend interviews now”.

```

class Interview
{
    public static void main(String[] args)
    {
        int mark = 74;
        if (mark >=75)
        {
            System.out.println("You can attend interview");
        }
        else
        {

```

```

        System.out.println("You cannot attend interviews now");
    }
}
}

```

#### OUTPUT :-

You cannot attend interviews now.

→ | Write a program to perform the below operation.

If emp sal is between 0 to 5000 then increase sal by 200.

If emp sal is between 5001 to 10000 then increase sal by 500

If emp sal is between 10001 to 20000 then increase sal by 1000

For any other salary display no hike this year.

```

class Salary
{
    public static void main(String[] args)
    {
        int sal = 1500;
        if (sal >= 0 && sal <= 5000)
        {
            System.out.println("New Salary = " + (sal + 200));
        }
        else if (sal >= 5001 && sal <= 10000)
        {
            System.out.println("New Salary = " + (sal + 500));
        }
        else if (sal >= 10001 && sal <= 20000)
        {
            System.out.println("New Salary = " + (sal + 1000));
        }
        else
        {
            System.out.println("No hike this year.");
        }
    }
}

```

#### OUTPUT :-

New Salary = 1700

Write a program to perform the below operation

If the number is equal to 0 then display "Suresh Rocks"

If number is positive the perform below operation

a. if the number is even then display "Hi Dinga"

b. if the number is odd then display "Hi Tingi"

else display "Ramesh Rocks"

```

class Op
{
    public static void main(String[] args)
    {
        int n = -6;
    }
}

```



```

        if (n==0)
        {
            System.out.println("Suresh Rocks");
        }
        else if (n>0)
        {
            if (n%2 == 0)
            {
                System.out.println("Hi Dinga");
            }
            else
            {
                System.out.println("Hi Tingi");
            }
        }
        else
        {
            System.out.println("Ramesh Rocks");
        }
    }
}

```

#### **OUTPUT :-**

Ramesh Rocks

→ | Write a program to perform below operation.

If emp experience in more than or equal to 3 years do the following

    If gender is male , display “Your trip is for Hebbal Lake”

    If gender is female, display “Your trip is for Ulsoor lake”

Else display “no trip this year”

```

class Trip
{
    public static void main(String[] args)
    {
        int exp_yr = 4;
        if (exp_yr >= 3)
        {
            char g = 'F';
            if (g == 'M')
            {
                System.out.println("Your trip is for Hebbal Lake");
            }
            else
            {
                System.out.println("Your Trip is for ulsoor lake");
            }
        }
        else
        {
            System.out.println("No trip this year!");
        }
    }
}

```

#### **OUTPUT :-**

Your Trip is for ulsoor lake

**DATE :-** 03-01-2022

### → Switch Statement :-

If we want to develop any menu-based program, we'll make use of the switch statement.

**Break** is used to terminate the switch block i.e., whenever the **JVM** encounters **break** statement, it will go out of the **switch** block and continue with other statements.

#### Syntax :-

Switch(condition)

```
{
Case 1 : -----
        -----
        Break;
Case 2 : -----
        -----
        Break;
Case 3 : -----
        -----
        Break;
default: System.out.println("Invalid choice");
}
```

### //Write a program to perform calculator operation

```
class Calculator
{
    public static void main(String[] args)
    {
        int x=100, y=50;

        int choice=4;

        switch (choice)
        {
            case 1 : System.out.println("sum = "+(x+y));
                    break;
            case 2 : System.out.println("difference = "+(x-y));
                    break;
            case 3 : System.out.println("product = "+(x*y));
                    break;
            case 4 : System.out.println("quotient = "+(x/y));
                    break;
            case 5: System.out.println("remainder = "+(x%y));
                    break;

            default : System.out.println("Invalid choice");
        }
    }
}
```

### OUTPUT :-

quotient = 2

### Example Program :-

```
class Grade
{
    public static void main(String[] args)
    {
        int grade='A';

        switch (grade)
        {
            case 'A' : System.out.println("FCD");
                        break;

            case 'B' : System.out.println("FC");
                        break;

            case 'C' : System.out.println("SC");
                        break;

            case 'D' : System.out.println("TC");
                        break;

            case 'E' : System.out.println("FAIL");
                        break;

            default: System.out.println("Invalid grade");
        }
    }
}
```

**OUTPUT :-**  
FCD

### → LOOPING :

If we want to perform the similar task repeatedly, we can go for any one of the looping statements.

### → For loop :

5.

```
for(initialization; condition; increment/decrement)
{
    -----
    -----
}
```

**Write a program to print all the numbers between 1 to 5 in the reverse order.**

```
for (int i = 5; i>=1; i--)
{
    Sop(i);
}
```

**Write a program to print the even numbers between 1 to 20.**

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

```

{
    System.out.println("Main Starts");

    for (int i=1; i<=20; i++)
    {
        if (i%2==0)
        {
            System.out.println(i);
        }
    }

    System.out.println("Main Ends");
}
}

```

**Write a program to print the odd numbers between 1 to 10 in the reverse order.**

```

class Sample
{
    public static void main(String[] args)
    {
        System.out.println("Main Starts");

        for (int i=10; i>=1; i--)
        {
            if (i%2!=0)
            {
                System.out.println(i);
            }
        }

        System.out.println("Main Ends");
    }
}

```

**Write a program to print “Dinga Rocks” 10 times.**

```

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

```

**Write a program to print all the numbers between 101 to 201.**

```

for (int i = 101; i<=201; i++)
{
    System.out.println(i);
}

```

**Write a program to print all the numbers between 251 to 301 in the reverse order.**

```

for (int i = 301; i>=251; i--)
{
    System.out.println(i);
}

```

**Write a program to all the even numbers between 137 to 256 in the reverse order.**

```

for (int i = 256; i>=137; i--)
{

```

```

        if(i%2==0)
        {
            System.out.println(i);
        }
    }
}

```

**Write a program to print all the odd numbers between 11 to 71.**

```

for (int i = 11; i<=71; i++)
{
    if(i%2!=0)
    {
        System.out.println(i);
    }
}

```

**Write a program to print the sum of all numbers between 1 to 5.**

```

class Sum
{
    public static void main(String[] args)
    {
        int sum=0;

        for (int i=1; i<=5 ; i++ )
        {
            sum=sum+i;
            //0+1
            //1+2
            //3+3
            //6+4
            //10+5
        }
        System.out.println("Sum = " + sum);
    }
}

```

**OUTPUT :-**

Sum = 15

**Write a program to print the sum of all even numbers between 1 to 10.**

```

class Sum
{
    public static void main(String[] args)
    {
        int sum=0;

        for (int i=1; i<=10 ; i++ )
        {
            if(i%2==0)
            {
                sum=sum+i;
            }
        }
        System.out.println("Sum = " + sum);
    }
}

```

**Write a program to print the sum of all odd numbers between 11 to 100.**

```
class Sum
{
    public static void main(String[] args)
    {
        int sum=0;

        for (int i=11; i<=100 ; i++ )
        {
            if (i%2 != 0)
            {
                sum=sum+i;
            }
        }
        System.out.println("Sum = " + sum);
    }
}
```

**Date :- 04-01-2022**

**MANISHA MA'M**

**Write a program to check whether the given number is odd or even without using modulus(%).**

**i.e using Bitwise operator.**

```
class Mam
{
    public static void main(String[] args)
    {
        int n = 7;
        if ((n&1)==0)
        {
            System.out.println("Even");
        }
        else
        {
            System.out.println("Odd");
        }
    }
}
```

**Write a program to count number of digits.**

```
class Mam
{
    public static void main(String[] args)
    {
        int n = 4523;
        int d = 0;
        while(n>0)
        {
            d=d+1;
            n=n/10;
        }
        System.out.println(d);
    }
}
```

**Write a program to print sum of given digits.**

```
class Mam1
{
    public static void main(String[] args)
    {
        int n = 4523;
        int d = 0;

        while(n>0)
        {
            d=d+ (n%10);
            n=n/10;
        }
        System.out.println("Sum of given digits is " + d);
    }
}
```

**Date :- 05-01-2022**

**MANISHA MA'M**

**Write a program to count even digits.**

```
class Evendigits
{
    public static void main(String[] args)
    {
        int n =84312;
        int d=0, g=0;
        while (n>0)
        {
            g=n%10;
            if (g%2==0)
            {
                d++;
            }
            n=n/10;
        }
        System.out.println("No. of Even digits = " + d);
    }
}
```

**Or**

```
class Evendigits
{
    public static void main(String[] args)
    {
        int n =84312;
        int d=0;
        while (n>0)
        {
            if (((n%10)&1)==0)
            {
                d++;
            }
            n=n/10;
        }
        System.out.println("No. of Even digits = " + d);
    }
}
```

```
}
```

**Write a program to print even digits from the number.**

```
public static void main(String[] args)
{
    int n =84312;
    int g=0;
    while (n>0)
    {
        g=n%10;
        if ((g&1)==0)
        {
            System.out.println(g);
        }
        n=n/10;
    }
}
```

**OUTPUT:-**

2  
4  
8

**Write a program to print sum of even digits from the number.**

```
public static void main(String[] args)
{
    int n =84312;
    int d=0, g=0;
    while (n>0)
    {
        g=n%10;
        if ((g&1)==0)
        {
            d=d+g;
        }
        n=n/10;
    }
    System.out.println("Sum of even digits is " + d);
}
```

**Write a program to print the factors of the given number.**

```
class Factors
{
    public static void main(String[] args)
    {
        int n=12;
        System.out.println("Factors of given number are: ");
        for (int i=1; i<=n; i++)
        {
            if((n%i)==0)
            {
                System.out.println(i);
            }
        }
    }
}
```



### OUTPUT :-

Factors of given number are:

1  
2  
3  
4  
6  
12

### Write a program to check if a given number is prime or not.

```
class Prime
{
    public static void main(String[] args)
    {
        int n=7;
        int c=0;
        for (int i=2; i<n; i++ )
        {
            if((n%i)==0)
            {
                c = 1;
            }
        }
        if(c==0)
        {
            System.out.println("Number is prime");
        }
        else
        {
            System.out.println("Number is not prime");
        }
    }
}
```

### Write a program to print factorial of a given number.

(factorial of 4 i.e  $4! = 4 \times 3 \times 2 \times 1$ )

```
class Factorial
{
    public static void main(String[] args)
    {
        int n=6;
        int b=1;
        for (int i =1;i<=n ;i++)
        {
            b *= i;
        }
        System.out.println("Factorial of given number is " +b);
    }
}
```

### Write a program to check if a given number is strong number.

[Strong number is a special number whose sum of the factorial of digits is equal to the original number. For example : 145 is strong number.

Since  $1! + 4! + 5! = 145$ . Other lists of Strong numbers are 1,2,40585]

```
class StrongNum
{
    public static void main(String[] args)
    {
        int n=40585;
        int g=0;
```

```

int d=n;
while (n>0)
{
    int s=1;
    for (int i=1;i<=(n%10) ; i++)
    {
        s *= i;
    }
    g=g+s;
    n=n/10;
}
if (d==g)
{
    System.out.println("Number is Strong");
}
else
{
    System.out.println("Number is not strong");
}
}
}

```

### Write a program to check if number is happy or unhappy.

[A **happy number** is a number which eventually reaches 1 when replaced by the sum of the square of each digit. For instance, 13 is a happy number because  $1^2 + 3^2 = 10$  and  $1^2 + 0^2 = 1$ . Other **happy** numbers are 7,28,100,320. Otherwise, it is called **sad** or **unhappy** number.]

Class HappyNum

### Write a program to check if number is xylem or phloem.

[A number  $N$  will be a **xylem number** if the sum of its extreme (first and last) digits is equal to the sum of its mean (except first and last) digits. Otherwise, it is **phloem number**.]

```

class Xyph
{
    public static void main(String[] args)
    {
        int n=12348;
        int d=n;
        int esum=0, bsum=0;

        while(n!=0)
        {
            if(n==d || n<10)
            {
                esum=esum + n%10;
            }
            else
            {
                bsum=bsum+n%10;
            }
            n=n/10;
        }

        System.out.println("Sum of Extreme digits is "+esum);
        System.out.println("Sum of mean digits is "+bsum);

        if(esum==bsum)
        {
            System.out.println(d+" is xylem number");
        }
        else
        {

```

```

        System.out.println(d+" is phloem number");
    }
}
}

```

**Date :- 06-01-2022**

**MANISHA MA'M**

Step 1 :- import java.util.Scanner;

Step 2 :- Scanner sc = new Scanner(System.in);

Step 3 :- sc.nextInt();  
 sc.nextDouble();  
 sc.nextLong();  
 sc.next();  
 sc.nextLine();  
 sc.next().charAt(0);

### **Example Program :-**

```

import java.util.Scanner;

class Mam2
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter Student Name");
        String name = sc.nextLine();
        System.out.println("Enter Student ID");
        int id = sc.nextInt();
        System.out.println("Enter Student Marks");
        double marks = sc.nextDouble();

        System.out.println("Student Name = " + name);
        System.out.println("Student ID = " + id);
        System.out.println("Student Marks = " + marks);
    }
}

```

### **OUTPUT :-**

```

Enter Student Name
Sagar
Enter Student ID
091
Enter Student Marks
629
Student Name = Sagar
Student ID = 91
Student Marks = 629.0

```

**Write a program to check if the given number is palindrome or not.**

```

import java.util.Scanner;

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

```

```

Scanner sc = new Scanner(System.in);
System.out.println("Enter the number");
int n= sc.nextInt();
int d=n;
int sum=0;
while(n>0)
{
    int g = n%10;
    sum = (sum*10)+g;
    n=n/10;
}
if (sum==d)
{
    System.out.println(d+" is a palindrome number");
}
else
{
    System.out.println(d+" is not a palindrome number");
}
}
}

```

#### OUTPUT :-

Enter the number

121

121 is a palindrome number

**Write a program to check if a given number is perfect or not.**

```

class Mam6
{
    public static void main(String[] args)
    {
        int n=28, s=0;
        for (int i=1; i<n; i++ )
        {
            if((n%i)==0)
            {
                s=s+i;
            }
        }
        if (s==n)
        {
            System.out.println("Number is perfect");
        }
        else
        {
            System.out.println("Number is not perfect");
        }
    }
}

```

**Write a program to print only palindrome number from 100  $\leftrightarrow$  9999.**

```

class Pldnrm
{
    public static void main(String[] args)
    {
        for (int i=100; i<=9999 ; i++ )
        {
            int j=i;
            int sum=0;

```

```

        while (j>0)
        {
            int g = j%10;
            sum = (sum*10)+g;
            j=j/10;
        }
        if (sum==i)
        {
            System.out.println(sum);
        }
    }
}

```

**Write a program to print even numbers from 1 to 100.**

```

class Mam5
{
    public static void main(String[] args)
    {
        for (int i=2; i<=100; i++)
        {
            if ((i&1)==0)
            {
                System.out.println(i);
            }
        }
    }
}

```

**Write a program to print perfect numbers from 1 to 50.**

```

class Mam7
{
    public static void main(String[] args)
    {
        for (int j=1; j<=50 ; j++ )
        {
            int s=0;
            for (int i=1; i<j; i++ )
            {
                if((j%i)==0)
                {
                    s=s+i;
                }
            }
            if (s==j)
            {
                System.out.println(j);
            }
        }
    }
}

```

**Date :- 10-01-2022**

**Shubam Sir**

**Write a program to print the count of even numbers between 1 to 10.**

```
class Ceven
{
    public static void main(String[] args)
    {
        int c=0;
        for (int i=1; i<=10; i++)
        {
            if ((i&1)==0)
            {
                c++;
            }
        }
        System.out.println("Total number of even number is " + c);
    }
}
```

**Write a program to count all the odd numbers between 1 to 20.**

```
class Codd
{
    public static void main(String[] args)
    {
        int c=0;
        for (int i=1; i<=20; i++)
        {
            if ((i&1)!=0)
            {
                c++;
            }
        }
        System.out.println("Total no. of odd number is " + c);
    }
}
```

### **Interview Questions on FOR LOOP :-**

**Write a program to print factors of a given number. [/.....Previously executed above]**

**Write a program to count the number of factors for a given number.**

```
class Cfactor
{
    public static void main(String[] args)
    {
        int n=12, c=0;
        for (int i=1; i<=n; i++)
        {
            if ((n%i)==0)
            {
                c++;
            }
        }
        System.out.println("Total no. of factors is " + c);
    }
}
```

**Write a program to check if a given number is prime number or not. [/....already executed above]**

**Write a program to check if a given number is perfect number or not. [/....already executed above]**

**Write a program to print the Fibonacci series for the given length.**

```
class Fibonacci
{
    public static void main(String[] args)
    {
        int length = 7;
        int n1=0, n2=1;
        if (length==0)
        {
            System.out.println("Nikal Pehli Fursad Mein ! ");
        }
        else if (length==1)
        {
            System.out.println(n1);
        }
        else if (length==2)
        {
            System.out.println(n1);
            System.out.println(n2);
        }
        else
        {
            System.out.println(n1);
            System.out.println(n2);
            for (int i=3; i<=length ; i++ )
            {
                int n3=n1+n2;
                System.out.println(n3);
                n1=n2;
                n2=n3;
            }
        }
    }
}
```

**OUTPUT :-**

0  
1  
1  
2  
3  
5  
8

**While Loop :-**

Syntax :-

```
While(condition)
{
    -----
    -----
}
```

## Example Programs :-

### 1. Write a program to print "HELLO WORLD" 5 times.

```
int l=1;
while(l<=5)
{
    System.out.println("HELLO WORLD");
    l++;
}
```

### 2. Write a program to print all the numbers between 1 to 10.

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

### 3. Write a program to print all the even numbers between 1 to 10.

```
int i=1;
while(i<=10)
{
    if(i%2==0)
    {
        System.out.println(i);
    }
    i++;
}
```

### 4. Write a program to print all the numbers between 1 to 10 in the reverse order.

```
int i=10;
while(i>=1)
{
    System.out.println(i);
    i--;
}
```

### 5. Write a program to print all the odd numbers between 100 to 200 in the reverse order.

```
int i=200;
while(i>=100)
{
    if(i%2!=0)
    {
        System.out.println(i);
    }
    i--;
}
```

### 6. Write a program to print the sum of all even numbers between 1 to 100.

```
int i=1, s=0;
while(i<=100)
{
```



```

    if(i%2==0)
    {
        s=s+i;
    }
    i++;
}
System.out.println("The sum of even numbers is " + s);

```

## 7. Write a program to count the number of odd numbers between 20 to 70.

```

int i=20, c=0;
while(i<=70)
{
    if(i%2!=0)
    {
        c++;
    }
    i++;
}
System.out.println("The total of odd numbers is " + c);

```

## Interview Questions on WHILE LOOP :

### 1. Write a program to find factorial of a given number.

```

int num=5, fact=1;
while(num>=1)
{
    fact= fact*num;
    num--;
}
System.out.println(fact);

```

### 2. Write a program to print digit by digit for a given number.

```

int num =123;
while(num>0)
{
    int rem = num%10;
    System.out.println(rem);
    num = num/10;
}

```

### 3. Write a program to print the sum of digits of the given number.

```

int num =4257, sum=0;
while(num>0)
{
    int rem = num%10;
    sum = sum+rem;
    num = num/10;
}
System.out.println("Sum of all the digits is " +sum);

```

➔ **UNSTRUCTURED WAY OF PROGRAMMING :-**

Writing the same piece of code multiple times as per customer requirement is known as **unstructured way of programming**.

According to the IT standards unstructured way of programming is highly not recommended:

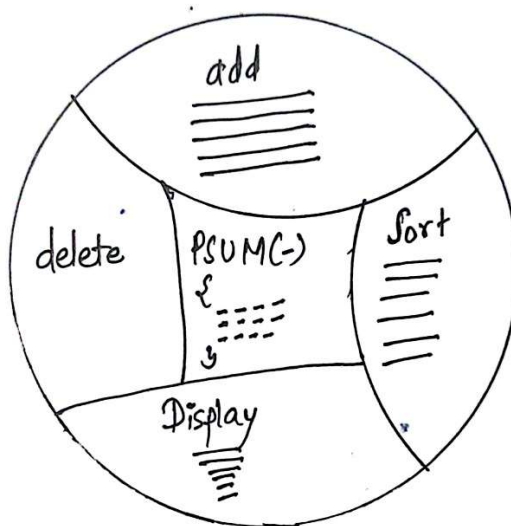
PSUM( - )

[illegible]

➔ **STRUCTURED WAY OF PROGRAMMING :-**

Writing the code once & executing the same piece of code multiple time as per the customer requirement is known as **structured way of programming**.

As per the IT-standards structured way of programming is recommended



*[Note :- In order to achieve structured way programming, we make use of methods or functions]*

## ➔ METHODS or FUNCTIONS :

Methods are set of instructions written by the programmer or developer in order to perform done specific task.

### Advantages of Methods :


- a. code reusability
- b. using methods we can achieve modularity (Structured way of programming)

### SYNTAX :

<access modifier> <modifier> <return type> method name(args)

Or

<access specifier>



{	Public	Static	Void
---- Task	protected	abstract	int
----	default	final	double
}	private	synchronize	Boolean
			float

### Example :-

```
public static void m1()  
{  
  ----  
  ----  
  ----  
  ----  
}
```

A user-defined method will be executed only when it is called explicitly by the user within the main method.

We can call a user defined method from the main method by using the below syntax:

Methodname(arg): ➔ *method calling statement*

Code:-

```
class A  
{  
  P S V main(-)  
{  
  Sop("hi");  
  m1();  
  m2(); method calling  
  Sop("Bye");  
}  
  
  P S V m1()  
{  
  Sop("hello m1");  
}  
}
```

### OUTPUT :-

```
hi  
hello m1  
hello m1  
bye
```

We can define any number of user defined methods within a class. These methods will be executed only when they are called explicitly by the user within the main method.

```
class Example
{
    public static void main(String[] args)
    {
        System.out.println("Hello main");
        x1();
        m1();
        System.out.println("Bye main");
    }
    public static void x1()
    {
        System.out.println("Hello x1");
    }
    public static void m1()
    {
        System.out.println("hello m1");
    }
}
```

### **OUTPUT :-**

```
Hello main
Hello x1
hello m1
Bye main
```

We can call one user-defined method from another user defined method but the initial call must always be from the main method.

code :

```
class Example2
{
    public static void main(String[] args)
    {
        System.out.println("Hello main");
        m1(); // method calling statement
        System.out.println("Bye Main");
    }

    public static void m1()
    {
        System.out.println("Hello m1");
        m2(); //method calling statement
        System.out.println("Bye m1");
    }

    public static void m2()
    {
        System.out.println("Hello m2");
        m3(); //method calling statement
        System.out.println("Bye m2");
    }

    public static void m3()
    {
        System.out.println("Hello m3");
        System.out.println("Bye m3");
    }
}
```

## NOTE :-

If we are defining a method & not calling it , there will be no issues as shown in class Demo1.  
But if we are calling a method & not defining it, we will get CTE as shown in class demo2.

```
class Demo1
{
    public static void main(String[] args)
    {
        System.out.println("Hello");
        System.out.println("Bye");
    }
    public static void m1()
    {
        System.out.println("hi");
    }
}
```

## OUTPUT :-

Hello  
Bye

```
class Demo2
{
    public static void main(String[] args)
    {
        System.out.println("Hello");
        m1(); //can't find m1 in class Demo2
        System.out.println("Bye");
    }
}
```

**Date :- 18-01-2022**

## PASSING/RETURNING VALUES TO/FROM A METHOD :

### I. Passing nothing to a method and returning nothing from the method.

#### Scenario 1:

```
class Demo
{
    p s v main(-- )
    {
        sop("Hi");
        m1();
        sop("Bye");
    }
}
```

```
p s v m1()
{
    sop("hey");
}
```

#### Scenario 2 :

```
class Demo
{
    p s v main(-- )
    {
        x1(); //method calling
    }
}
```

```
p s v x1()
```

```
{  
sop("hi x1");  
}
```

## II. Passing something to a method and returning nothing from the method.

### Scenario 1 :-

```
class Demo  
{  
    public static void main(String[] args)  
    {  
        System.out.println("main starts");  
        m1(10,10.12); //method calling  
        System.out.println("main ends");  
    }  
  
    public static void m1(int a, double b)  
    {  
        System.out.println(a);  
        System.out.println(b);  
    }  
}
```

### **OUTPUT :-**

```
main starts  
10  
10.12  
main ends
```

### Scenario 2 :-

```
class Sample  
{  
    public static void main(String[] args)  
    {  
        char x='A';  
        boolean y=false;  
        m1(x,y);  
    }  
  
    public static void m1(char a, boolean b)  
    {  
        System.out.println(a);  
        System.out.println(b);  
    }  
}
```

### **OUTPUT :-**

```
A  
false
```

## Interview Questions :-

### 1. Write a method that accepts 2 int values and returns nothing.

```
class Demo
{
    public static void main(String[] args)
    {
        m1(10,20); //method calling    or    int x=10, y=20;
                                           m1(x,y);

    }

    public static void m1(int a, int b)
    {
        System.out.println(a);
        System.out.println(b);
    }
}
```

### 2. Write a method that accepts 1 double, 1 char, 1 boolean value and returns nothing.

```
class Demo
{
    public static void main(String[] args)
    {
        m1(12.12, 'a', true); //method calling
    }

    public static void m1(double a, char b, boolean c)
    {
        System.out.println(a);
        System.out.println(b);
        System.out.println(c);
    }
}
```

### 3. Write a program to find if a given number is even or odd. [Using method]

```
class EvenorOdd
{
    public static void main(String[] args)
    {
        check(13);
        check(18);
        check(67);
    }

    public static void check(int num)
    {
        if (num%2==0)
        {
            System.out.println(num + " is an even number");
        }
        else
        {
            System.out.println(num + " is an odd number");
        }
    }
}
```

**OUTPUT :**

13 is an odd number  
18 is an even number  
67 is an odd number

**ASSIGNMENT :-****1. Write a method that accepts 2 float value and returns nothing .**

```
class Task2
{
    public static void main(String[] args)
    {
        float a=90.56f, b=67.201f;
        m1(a,b);
    }
    public static void m1(float a, float b)
    {
        System.out.println(a);
        System.out.println(b);
    }
}
```

**2. Write a method that accepts 1 char , 1 boolean value and returns nothing.**

```
class Task3
{
    public static void main(String[] args)
    {
        char a='k';
        boolean b=true;
        m1(a,b);
    }

    public static void m1(char x, boolean y)
    {
        System.out.println(x);
        System.out.println(y);
    }
}
```

**3. Write a method that accepts 1 int , 1 double ,1 float value and returns nothing.**

```
class Task4
{
    public static void main(String[] args)
    {
        m1(15,98.3,60.52f,'d');
    }
    public static void m1(int a, double b, float c, char d)
    {
        System.out.println(a);
        System.out.println(b);
        System.out.println(c);
        System.out.println(d);
    }
}
```



#### 4. Write a program to check if a given number is positive or negative.

```
class PnN
{
    public static void main(String[] args)
    {
        check(6);
        check(-8);
        check(4);
    }
    public static void check(int i)
    {
        if (i>0)
        {
            System.out.println(i+" is a positive number");
        }
        else
        {
            System.out.println(i+ " is a negative number");
        }
    }
}
```

##### OUTPUT :-

6 is a positive number  
-8 is a negative number  
4 is a positive number

#### 5. Write a program to check if a given character is vowel or consonant.

```
class VoC
{
    public static void main(String[] args)
    {
        check('i');
        check('g');
        check('d');
        check('a');
    }
    public static void check(char l)
    {
        if(l=='a' || l=='e' || l=='i' || l=='o' || l=='u')
        {
            System.out.println(l + " is a vowel character.");
        }
        else
        {
            System.out.println(l+ " is a consonant character.");
        }
    }
}
```

##### OUTPUT :-

i is a vowel character.  
g is a consonant character.  
d is a consonant character.  
a is a vowel character.

## 6. Write a program to find greatest of 2 number.

```
class G2
{
    public static void main(String[] args)
    {
        int a=10;
        int b=20;
        m1(a,b);
    }
    public static void m1(int x, int y)
    {
        if(x>y)
        {
            System.out.print(x+" is greater than "+y);
        }
        else
        {
            System.out.print(y+" is greater than "+x);
        }
    }
}
```

### OUTPUT :-

20 is greater than 10

Date : - 19-01-2022

## III. PASSING NOTHING TO METHOD BUT RETURNING SOMETHING FROM METHOD

1. Whenever a method wants to return some value back to its **caller**, it will make use of **return** word.
2. Whenever a method is returning some value, the return type of the method will also change based on the type of value it is returning.

```
class Demo3
{
    public static void main(String[] args)
    {
        System.out.println("Main Starts");
        int x=m1(); //method calling
        System.out.println(x);
        System.out.println("Main End");
    }
    public static int m1()
    {
        System.out.println("hello m1");
        return 100;
    }
}
```

datatype  
will always  
be same

100

### OUTPUT :

Main Starts  
hello m1  
100  
Main End

```

class Demo4
{
    public static void main(String[] args)
    {
        char x=m1();
        System.out.println(x);
    }
    public static char m1()
    {
        return 'A';
    }
}

```

#### OUTPUT :-

A

#### [Important Note 1:

Whenever a method is returning some value, the return statement must be the last statement of the method body, if at all we write any code after the return statement we get **CTE**(Compile Time Error) as shown below:]

```

class Demo3
{
    public static void main(String[] args)
    {
        double x=m1(); //method calling
        System.out.println("x= "+ x);
    }
    public static double m1()
    {
        return 12.12;
        System.out.println("hello m1");
    }
}

```

#### Error output :

Demo3.java:13: error: unreachable statement

```

    System.out.println("hello m1");
    ^

```

Demo3.java:14: error: missing return statement

```

    }
    ^

```

2 errors

#### [Important Note 2:

A user defined method can accept any numbers of values but it can return only one value, if at all we try to return multiple values from a method we will get **CTE** as shown below:]

```

class Demo3
{
    public static void main(String[] args)
    {
        double x=m1(); //method calling
        System.out.println(x);
    }
}

```

```

        public static double m1()
        {
            return 12.12;
            return 15.2;
        }
    }

```

### OUTPUT Error :

Demo3.java:13: error: unreachable statement

```

        return 15.2;
        ^

```

1 error

### [Important Note 3:

We can call a method directly inside the print statement if and only if the method is returning some value.  
If at all we can a method inside a print statement which is not returning any value then we will get compile time error as shown below:

### Scenario 1 :

```

class Demo3
{
    public static void main(String[] args)
    {

        double x=m1(); //method calling
        System.out.println(x);

    }
    public static double m1()
    {
        return 12.12;
    }
}

```

### OUTPUT :

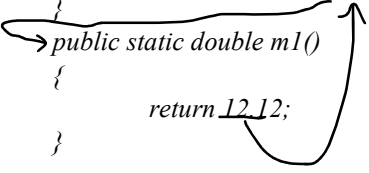
12.12

### Scenario 2:

```

class Demo3
{
    public static void main(String[] args)
    {
        System.out.println(m1());
    }
    public static double m1()
    {
        return 12.12;
    }
}

```



### OUTPUT :

12.12

### Scenario 3:

```

class Demo3
{

```

```

    public static void main(String[] args)
    {
        System.out.println(m1()); //produce CTE error
    }
    public static double m1()
    {
        System.out.println("hi");
    }
}

```

### Output Error:

Demo3.java:10: error: missing return statement

```

    }
    ^

```

1 error

*[i.e., can't call a method inside a print statement if the method returns nothing]*

## IV. PASSING SOMETHING TO A METHOD AND RETURNING SOMETHING FROM THE METHOD.

### Scenario 1:

```

class Demo5
{
    public static void main(String[] args)
    {
        System.out.println("Hi main");
        boolean x=m1(100,'A');
        System.out.println(x);
        System.out.println("Bye main");
    }

    public static boolean m1(int a , char b)
    {
        System.out.println("hi m1");
        System.out.println(a);
        System.out.println(b);
        return false;
    }
}

```

### OUTPUT :

```

Hi main
hi m1
100
A
false
Bye main

```

### Scenario 2:

```

class Demo6
{
    public static void main(String[] args)
    {
        int a=100;
        float b=12.2f;
        String x=m1(a,b);
    }
}

```

```

        System.out.println(x);
    }

    public static String m1(int x, float y)
    {
        System.out.println(x);
        System.out.println(y);
        return "hello";
    }
}

```

**OUTPUT :**

100  
12.2  
hello

**Date :- 20-01-2022**

**Interview Questions :**

**1. Write a method that accepts 1 int, 1 float value and returns double value.**

```

class Task7
{
    public static void main(String[] args)
    {
        double j=m1(54,65f);
        System.out.println(j);
    }
    public static double m1(int a, float b)
    {
        System.out.println(a);
        System.out.println(b);
        return 35.21;
    }
}

```

**OUTPUT :-**

54  
65.0  
35.21

**2. Write a method that accepts 1 char, 1 boolean, 1 int value and return string value.**

```

class Task8
{
    public static void main(String[] args)
    {
        String g=m1('f',false);
        System.out.println(g);
    }
    public static String m1(char a, boolean b)
    {
        System.out.println(a);
        System.out.println(b);
        return "Sagar";
    }
}

```

```
}
```

**OUTPUT :-**

```
f  
false  
Sagar
```

**3. Write a program to find if a given number is perfect number or not.**

**Type 2:**

```
class Perfect2  
{  
    public static void main(String[] args)  
    {  
        checkPerfect(6);  
        checkPerfect(24);  
        checkPerfect(28);  
    }  
  
    public static void checkPerfect(int num)  
    {  
        int sum=0;  
        for (int i=1; i<num ; i++ )  
        {  
            if (num%i==0)  
            {  
                sum=sum+i;  
            }  
        }  
        if (sum==num)  
        {  
            System.out.println(num+" is a perfect number");  
        }  
        else  
        {  
            System.out.println(num+" is not a perfect number");  
        }  
    }  
}
```

**OUTPUT :-**

```
6 is a perfect number  
24 is not a perfect number  
28 is a perfect number
```

**TYPE 4 :**

```
class Perfect4  
{  
    public static void main(String[] args)  
    {  
        int x=6;  
        int a=checkPerfect(x);  
        if (x==a)  
        {  
            System.out.println(x+" is a perfect number");  
        }  
        else  
        {  

```

```

        System.out.println(x+" is not a perfect number");
    }
}

public static int checkPerfect(int num)
{
    int sum=0;
    for (int i=1; i<num ; i++ )
    {
        if (num%i==0)
        {
            sum=sum+i;
        }
    }
    return sum;
}
}

```

### OUTPUT :

6 is a perfect number

### [Important Note :-

*Two variables can have same name provided if they are in different methods and if we do changes to one variable, it will not affect, the other variable which is present in the other method as shown below :]*

```

class Demo7
{
    public static void main(String[] args)
    {
        int x=10;
        double y=10.12;
        m1(x,y);
        System.out.println(x); // 10
        System.out.println(y); //10.12
    }
    public static void m1(int x, double y)
    {
        System.out.println(x); //10
        System.out.println(y); //10.12
        x--;
        System.out.println(x); //9
    }
}

```

### OUTPUT :-

```

10
10.12
9
10
10.12

```



## **ASSIGNMENT-4:**

### **1. Write a method that accepts 2 String Value and returns char.**

```
class Asnmt
{
    public static void main(String[] args)
    {
        int x = m1("Hello", " World");
        System.out.println(x);
    }
    public static int m1(String a, String b)
    {
        System.out.print(a);
        System.out.println(b);
        return 100;
    }
}
```

#### **OUTPUT :**

Hello World  
100

### **2. Write a method that accepts 1 double , 1 char value and returns boolean value.**

```
class Asmnt2
{
    public static void main(String[] args)
    {
        boolean x = m1(65.02,'S');
        System.out.println(x);
    }
    public static boolean m1(double a, char b)
    {
        System.out.println(a);
        System.out.println(b);
        return true;
    }
}
```

#### **OUTPUT :**

65.02  
S  
true

### **3. Write a method that accepts 2 char value , 1 float value and returns int .**

```
class Asmnt3
{
    public static void main(String[] args)
    {
        int x = m1('f', 65.92f);
        System.out.println(x);
    }
    public static int m1(char a, float b)
    {
        System.out.println(a);
        System.out.println(b);
        return 100;
    }
}
```

```

    }
}

```

**OUTPUT :**

```

f
65.92
100

```

**4. Write a program to check if a number is prime number or not.(using both type 2 and type 4 approach)**

**TYPE 2 :**

```

class Asmnt42
{
    public static void main(String[] args)
    {
        check(7);
        check(21);
        check(23);
        check(27);
    }
    public static void check(int n)
    {
        int c=0;
        for (int i=2; i<n ;i++ )
        {
            if (n%i==0)
            {
                c++;
            }
        }
        if (c==0)
        {
            System.out.println(n+" is a prime number.");
        }
        else
        {
            System.out.println(n+" is not a prime number.");
        }
    }
}

```

**OUTPUT :**

```

7 is a prime number.
21 is not a prime number.
23 is a prime number.
27 is not a prime number.

```

**TYPE 4:**

```

class Asmnt44
{
    public static void main(String[] args)
    {
        int x=check(9);
        if (x==0)
        {

```

```

        System.out.println("Number is prime");
    }
    else
    {
        System.out.println("Number is not prime.");
    }
}
public static int check(int n)
{
    int c=0;
    for (int i=2; i<n ;i++ )
    {
        if (n%i==0)
        {
            c++;
        }
    }
    return c;
}
}

```

#### OUTPUT :

Number is not prime.

Date :- 21-01-2022

#### → **WHY USER DEFINED METHODS?**

As per the IT rules, no logic should be written inside the **main** method. The logic must be written inside a **user defined method** and the **main** method should only contain the following:

- Variable Declaration
- Initialize the variable by taking input from the user.
- Call the method that contains logic.

#### → **WHAT IS THE IT STANDARD FOR GIVING CLASS NAME?**

```
class evenorodd
```

```
{
```

```
-----
```

```
-----
```

**ALLOWED BUT NOT RECOMMENDED**

```
-----
```

```
}
```

```
class EvenOrOdd
```

```
{
```

```
-----
```

```
-----
```

**ALLOWED AND RECOMMENDED**

```
-----
```

```
}
```

→ **WHAT IS THE IT STANDARD FOR GIVING METHOD NAME?**

```
public static void checkprimenumber()
```

```
{
```

```
-----
```

```
----- ALLOWED BUT NOT RECOMMENDED
```

```
-----
```

```
}
```

```
public static void checkPrimeNumber()
```

```
{
```

```
-----
```

```
----- ALLOWED AND RECOMMENDED
```

```
-----
```

```
}
```

## TAKING INPUT FROM THE USER:

\*\*\*\*\*

1.

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

2.

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

The first 2 statements are common for all the programs if we want to read input from the user.

3.

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

```
double b = sc.nextDouble();
```

```
float c = sc.Float();
```

```
Boolean d = sc.Boolean();
```

```
char e = sc.next().charAt(0);
```

```
String f=sc.next();
```

```
String g=sc.nextLine();
```

The third will differ based on the type of input that we want to enter.

### EXAMPLE\_PROGRAM :

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

```
class InputExample
```

```
{
```

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

```
    {
```

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

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

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

```
        System.out.println("Enter a double value");
```

```
        double b=sc.nextDouble();
```

```
        System.out.println("Enter a boolean value");
```

```
        boolean c=sc.nextBoolean();
```

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

```
        char d=sc.next().charAt(0);
```

```
        System.out.println("Char value is "+d);
```

```
        System.out.println("Number is "+a);
```

```
        System.out.println("Boolean Value is "+c);
```

```
        System.out.println("Double Value is "+b);
```

```
    }
```

```
}
```

### OUTPUT :

**Enter a number**

15

**Enter a double value**

65.02

**Enter a boolean value**

true

**Enter a char value**

g

Char value is g

Number is 15

Boolean Value is true

Double Value is 65.02

## Taking Input as STRING:

1. `sc.next()` → It will consider space as end of the String.

**Example:**

```
import java.util.Scanner;

class Exmple1
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a String:");
        String s = sc.next(); //Considers Space button as end of the String.
        System.out.println("S = "+s);
    }
}
```

**OUTPUT :**

**Enter a String:**

Hello Sagar

S = Hello

2. `sc.nextLine()` → It will consider enter button as end of the String

**Example:**

```
import java.util.Scanner;

class Exmple2
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a String:");
        String s = sc.nextLine(); //considers ENTER button as end of the String
        System.out.println("S = "+s);
    }
}
```

**OUTPUT :**

**Enter a String:**

Hello World

S = Hello World

## → Interview Question:

**Write a program to print digit by digit for a given number. (Using Methods & taking input from the User.)**

```
import java.util.Scanner;

class DigitByDigit
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a number");
        int num= sc.nextInt();
        printDigit(num);
    }
    public static void printDigit(int num)
    {
        System.out.println("=====");
        System.out.println("Digits are : ");
        System.out.println("=====");
        while (num>0)
        {
            int rem=num%10;
            System.out.println(rem);
            num=num/10;
        }
    }
}
```

### **OUTPUT :**

**Enter a number**

15436

=====

Digits are :

=====

6

3

4

5

1

### **IMPORTANT NOTE :**

***/\* While reading input from the user if the input type in which we are storing the input are not same then we will get runtime error.***

***Exceptions are type of runtime error \*/***

### **EXAMPLE\_PROGRAM :**

```
import java.util.Scanner;
class Demo
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a int value");
    }
}
```

```

        int n=sc.nextInt();
        System.out.println("number is "+n);
    }
}

/*
OUTPUT
=====
Enter a int value
12.12
RUNTIME ERROR(InputMismatchException, int value was expected).
*/

```

## Scenarios :

1.

```
char ch = sc.next().charAt(0);
sop(ch); //A
```

**Input:**

-----

A

2.

```
char ch = sc.next().charAt(3);
sop(ch); //I
```

**Input:**

-----

Hello

3.

```
char ch = sc.next().charAt(5);
sop(ch); //g
```

**Input:**

-----

asdsfgh

4.

```
char ch = sc.next().charAt(4);
sop(ch); //O
```

**Input:**

-----

WELCOME

5.

```
char ch = sc.next().charAt(6);
sop(ch); //u
```

**Input:**

-----

qwertyuiop



6.

```
char ch = sc.next().charAt(-1);  
sop(ch);
```

**Input:**

-----

ASD

RTE  
OUT OF INDEX EXCEPTION

**[Important Note :**

While reading char input the user can pass series of character as input but the **JVM** will read only one char out of them based on the specified index value.

If the user goes beyond or below the index value, we will get Exception During Runtime as above **Example no. 6.]**

**ASSIGNMENT 5 :**

**1. Write a program to print factors of a given number.**(use methods and take input from user)

```
import java.util.Scanner;  
class FactNum  
{  
    public static void main(String[] args)  
    {  
        Scanner sc = new Scanner(System.in);  
        System.out.println("Enter a number:");  
        System.out.println("=====");  
        int num = sc.nextInt();  
        m1(num);  
    }  
  
    public static void m1(int num)  
    {  
        System.out.println("=====");  
        System.out.println("Factors of "+num+" are: ");  
        for (int i=1; i<=num ; i++ )  
        {  
            if (num%i==0)  
            {  
                System.out.println(i);  
            }  
        }  
    }  
}
```

**OUTPUT :**

Enter a number:

=====

6

=====

Factors of 6 are:

1

2

3

6

## 2. Write a program to print sum of all digits for a given number.(use methods & take input from user)

```
import java.util.Scanner;
class SumOfDigit
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a number:");
        System.out.println("=====");
        int num = sc.nextInt();
        m1(num);
    }
    public static void m1(int num)
    {
        int sum=0;
        while (num>0)
        {
            int rem=num%10;
            sum=sum+rem;
            num=num/10;
        }
        System.out.println("=====");
        System.out.println("Sum of digits of given number is "+sum);
    }
}
```

### OUTPUT :

Enter a number:

=====

6502

=====

Sum of digits of given number is 13

## 3. Write a program to print digit by digit of a given number. (use methods & take input from user)

```
import java.util.Scanner;

class DigitByDigit
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a number");
        int num= sc.nextInt();
        printDigit(num);
    }
    public static void printDigit(int num)
    {
        System.out.println("=====");
        System.out.println("Digits are : ");
        System.out.println("=====");
        while (num>0)
        {
            int rem=num%10;
            System.out.println(rem);
            num=num/10;
        }
    }
}
```

```
    }
}
```

**OUTPUT :**

**Enter a number**

15436

=====

Digits are :

=====

6

3

4

5

1

**4. Write a program to check if a given number is prime number or not. (use methods and take input from user)**

```
import java.util.Scanner;
class PrimeNum
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a number:");
        System.out.println("=====");
        int num = sc.nextInt();
        m1(num);
    }
    public static void m1(int num)
    {
        int c=0;
        for (int i=2; i<num; i++ )
        {
            if(num%i==0)
            {
                c++;
            }
        }
        System.out.println("=====");
        if(c==0)
        {
            System.out.println(num+" is a prime number.");
        }
        else
        {
            System.out.println(num+" is not a prime number.");
        }
    }
}
```

**OUTPUT :**

**Enter a number:**

=====

5

=====

5 is a prime number.

**5. Write a program to print the fibonacci series for the given length. (use methods & input from user)**

```
import java.util.Scanner;
class FibonacciSeries
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        int n1=0, n2=1;
        System.out.println("Enter length of fibonacci series");
        System.out.println("=====");
        int length = sc.nextInt();
        m1(length,n1,n2);
    }
    public static void m1(int length, int n1, int n2)
    {
        System.out.println("=====");
        System.out.println("Fibonacci series of given length are:");
        if (length<=0)
        {
            System.out.println("Oops!..Please Enter the valid integer length");
        }
        else if (length==1)
        {
            System.out.println(n1);
        }
        else if (length==2)
        {
            System.out.println(n1);
            System.out.println(n2);
        }
        else
        {
            System.out.println(n1);
            System.out.println(n2);
            for (int i=3; i<=length; i++)
            {
                int n3=n1+n2;
                System.out.println(n3);
                n1=n2;
                n2=n3;
            }
        }
    }
}
```

**OUTPUT :**

**Enter length of fibonacci series**

=====

7

=====

**Fibonacci series of given length are:**

0

1

1

2

3

5

8

**6. Write a program to print factorial of a given number. (use methods and take input from user)**

```
import java.util.Scanner;
class FactorialNum
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a number:");
        System.out.println("=====");
        int num = sc.nextInt();
        int fac=1;
        m1(num,fac);
    }

    public static void m1(int num,int fac)
    {
        System.out.println("=====");
        if(num==0)
        {
            System.out.println("The factorial of 0 is 1.");
        }
        else if(num>0)
        {
            for (int i=1; i<=num ; i++ )
            {
                fac=fac*i;
            }
            System.out.println("Factorial of "+num+" is "+fac);
        }
        else
        {
            System.out.println("Oops!.. Please enter a valid number.");
        }
    }
}
```

**OUTPUT :**

**Enter a number:**

=====

5

=====

Factorial of 5 is 120

## BASICS OF STRING :

In java String is not a datatype, it is **pre-defined class**/inbuilt class which is available in **inbuilt** library folder. **String class** contains lot of predefined methods which are used to perform operations on string value.

### Methods of String class:

-----

#### a. int length()

*example :*

```
String s1 = "Welcome";  
int len = s1.length();  
Sop(len); or sop(s1.length()); //7
```

```
String str = "Hello";  
int x = str.length();  
Sop(x); or sop(str.length()); //5
```

```
String s = "Hi";  
int x = s.length();  
Sop(x); or sop(s.length()); //2
```

#### b. char charAt(int index)

*Example :*

**1.**

```
String s1= "Hello";  
char a = charAt(3);  
Sop(a); //l  
or  
Sop(charAt(3)); //l  
Sop(charAt(0)); //H  
Sop(charAt(4)); //o
```

**2.**

```
String s2 = "Wel Come";  
char ch= charAt(2);  
Sop(ch); //l
```

(or)

```
Sop(s2.charAt(6)); //m  
Sop(s2.charAt(4)); //C  
Sop(s2.charAt(3)); // (space)  
Sop(s2.charAt()); //CTE, INT ARG EXPECTED
```

#### c. Boolean equals(String s)

-----

It is responsible for comparing two string values, if the content is same then it will return true else it will return false.

Example :

-----

```
String s1= "Hello";  
String s2 = "Welcome";  
String s3= "Hello";
```

```
Boolean x=s1.equals(s2);  
Sop(x); // false  
(or)  
Sop(s1.equals(s2)); // false  
Sop(s1.equals(s3)); // true  
Sop(s2.equals(s3)); // false
```

**[Note :** We can't compare two strings using "==" operator because "==" is used to compare only primitive values where to compare String values we use an inbuilt method called "**equals(String str)**".]

**Interview question :**

**\*\*\*\*\***

**1. Write a program to print char by char for a given string value.**

```
import java.util.Scanner;  
  
class CharByChar  
{  
    public static void main(String[] args)  
    {  
        Scanner sc = new Scanner(System.in);  
        System.out.println("Enter a String");  
        String s1=sc.nextLine();  
        System.out.println("Character by character of given String are:");  
        for (int i=0; i<s1.length();i++ )  
        {  
            System.out.println(s1.charAt(i));  
        }  
    }  
}
```

**OUTPUT :**

```
Enter a String  
sagar  
Character by character of given String are:  
s  
a  
g  
a  
r
```

**2. Write a program to print char by char for a given String in reverse order.**

```
import java.util.Scanner;  
class ReverseChar  
{  
    public static void main(String[] args)
```

```

    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a String");
        String s1=sc.nextLine();
        System.out.println("Character by character of given String in reverse order are:");
        for (int i=s1.length()-1; i>=0;i--)
        {
            System.out.println(s1.charAt(i));
        }
    }
}

```

#### **OUTPUT :**

**Enter a String**

sagar

**Character by character of given String in reverse order are:**

r

a

g

a

s

### **3. Write a program to print vowel characters from the given string.**

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

```
class Vowel
```

```

{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a String");
        String s1=sc.nextLine();
        System.out.println("Vowel Characters of given String are:");
        for (int i=0; i<s1.length();i++ )
        {
            if (s1.charAt(i)=='a' || s1.charAt(i)=='e' || s1.charAt(i)=='i' || s1.charAt(i)=='o'
            || s1.charAt(i)=='u' ||
            s1.charAt(i)=='A' || s1.charAt(i)=='E' || s1.charAt(i)=='I' || s1.charAt(i)=='O' || s1.charAt(i)=='U'
            )
            {
                System.out.println(s1.charAt(i));
            }
        }
    }
}

```

#### **OUTPUT :**

**Enter a String**

sagar sunar

**Vowel Characters of given String are:**

a

a

u

a



#### 4. Write a program to count the number of vowels in a given string value.

```
import java.util.Scanner;
class CountVowel
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a String");
        String s1=sc.nextLine();
        System.out.println("Vowel Characters of given String are:");
        int c=0;
        for (int i=0; i<s1.length();i++ )
        {
            if (s1.charAt(i)=='a' || s1.charAt(i)=='e' || s1.charAt(i)=='i' || s1.charAt(i)=='o'
                || s1.charAt(i)=='u' || s1.charAt(i)=='A' || s1.charAt(i)=='E' || s1.charAt(i)=='I'
                || s1.charAt(i)=='O' || s1.charAt(i)=='U' )
            {
                c++;
            }
        }
        System.out.println("Total no. of vowel letter is: "+c);
    }
}
```

##### **OUTPUT :**

Enter a String

**hello sagar world**

Vowel Characters of given String are:

**Total no. of vowel letter is: 5**

#### 5. Write a program to print uppercase alphabet from a given string.

```
import java.util.Scanner;
class UppercaseAlphabet
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a string:");
        String str = sc.nextLine();
        for (int i=0;i<str.length();i++ )
        {
            if (str.charAt(i)>='A' && str.charAt(i)<='Z')
            {
                System.out.println(str.charAt(i));
            }
        }
    }
}
```

##### **OUTPUT :**

Enter a string:

**SagAr SuNar**

**S**

**A**

**S**

**N**

## 6. Write a program to print only the lowercase alphabets from given string value.

```
import java.util.Scanner;
class LowercaseAlphabet
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a string:");
        String str = sc.nextLine();
        for (int i=0;i<str.length();i++)
        {
            if (str.charAt(i)>='a' && str.charAt(i)<='z')
            {
                System.out.println(str.charAt(i));
            }
        }
    }
}
```

### **OUTPUT :**

Enter a string:

**SagAr SuNar**

**a**

**g**

**r**

**u**

**a**

**r**

## 7. Write a program print only the numbers from the given string value.

```
import java.util.Scanner;
class PrintNumber
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a string:");
        String str = sc.nextLine();
        System.out.println("=====");
        System.out.println("The number values are:");
        System.out.println("=====");
        for (int i=0;i<str.length();i++)
        {
            if (str.charAt(i)>='0' && str.charAt(i)<='9')
            {
                System.out.println(str.charAt(i));
            }
        }
    }
}
```

### **OUTPUT :**

Enter a string:

**sagarsunar13**

=====

The number values are:

=====

1  
3

## 8. Write a program to print if the given string is Palindrome or Not.

```
import java.util.Scanner;
class Palindrome
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a string:");
        String str = sc.nextLine();
        String sr="";

        for (int i= str.length()-1; i>=0 ; i-- )
        {
            sr=sr+str.charAt(i);
        }
        System.out.println("=====");
        System.out.println("The reverse of given string is : "+sr);
        System.out.println("=====");
        if (sr.equals(str))
        {
            System.out.println("Hence, "+str+" is a palindrome string");
        }
        else
        {
            System.out.println("Hence, "+str+" is not a palindrome string");
        }
    }
}
```

### OUTPUT :

Enter a string:

**malayalam**

=====

The reverse of given string is : **malayalam**

=====

Hence, **malayalam** is a palindrome string