Core Java

* Module - 1

1. Basic structure of programs
2. Architecture of java
3. Data types
4. Variables
5. Keywords and Identifiers
6. operators and types
7. Looping statement
8. pattern programming
9. General programs Based on For and While Loops

JAVA (Module-1)

Basic structure of program

Program:- A set of Instructions given to system in any programming language is called as program

Why do we develop a program

In order to develop an application

Why do we need app?

Application are developed to Convert manual work into automated.

ex:- you Girlfriend
hyderabad maldives 10,000

online transcation. gpay, phonepe, paytm

Java program

A Basic java program Consists of 3 parts

1. class declaration
2. main method
3. printing statement

Class declaration:-

* Syntax:- way of writing

(or) public class sample
public class Demo

public → Access modifier — provides permission

public means accessible to everyone

Access modifier:- It Indicates that program is accessible to other user or not

There are 4 access modifiers in java, private, public protected and default

* In above section class is public so it freely accessible

* All access modifiers will be in lower case

1. class :- class is a keyword (reserved word or predefined word) in java

* every program must start with class keyword

* all keyword must start with smaller case so class - c is small

* Class — keyword | keyword — important words of java

Class :- It is used to Represent a program

(or)

It is used to "write logic".

* Sample/demo :- our class name / program name

* every class has some name i.e class name or program name or file name

* class name for standard should start with Capital letter

* java file name and class name must be same for Remembering purpose

* class name can only be Combination of A-Z, a-z, 0-9, $ - - - - [A-Z, a-z, 0-9, $, _]

(_ — underscore)

* class name starts with capital and donot give space if you have more than one word

Myprogram

* ; Semi colon ⟶ End of statement / line

Note :- class Contains some logic so we should not End class with ;

* Note :- { } — for defining Body [Scope of class]

Ex :-
```
public class Myprogram
{
  logic of my program
}
```

Public class Demo

→ Access modifier – public says class accessible to everyone

→ keyword – for developing logic

→ programname/classname/filename – for identification

2. Main method :-

main method is a heart of our Java program (inside that we will write logic) because without main method our program will not work

Syntax :-

public static void main (String args[ ])
(capital letter)

* public indicates main method is accessible to everyone
* static is a keyword – indicates no need to create an object
* main – name of the method → ( )
* String args[ ] → Command Line Arguments (Array of programming)

Note :- every word first letter is small except

String – S

Void – is a return type → indicates not Returning any value

For Example:

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

class → main method → main method

class

③ Printing statement

System.out.println("This is any sample program");

System :- It is a predefined class [class System]

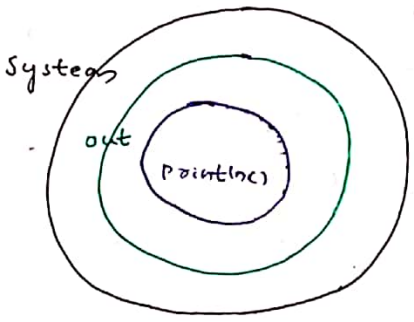
out :- It is an object (predefine)

println :- it is a method (predefine)

print() is accessed through out object but out object is present in system class

System is a class which Contains out object and out object is referring to println()

→ * whatever we gave in double quotes that message will be printed as it is



* Example :-

```
public class Saturday
{
public static void main(String args[])
{
    System.out.println("Today I studied, java program");
    System.out.println("And it is");
    System.out.println("Actually very easy");
}
}
```

output:-

```
Today I studied; java program
And it is
Actually very easy
```

```
public class Saturday
{
public static void main(String args[])
{
System.out.println("John - 9989899898");
System.out.println("22 years old");
System.out.println("Graduated in 2020");
}
}
```

output:-

John - 9989899898
22 years old
Graduated in 2020

```
public class Saturday
{
public static void main(String args[])
{
System.out.println("hello Friends");
System.out.println("stay Home");
System.out.println("stay safe");
}
}
```

output:-

hello Friends
stay Home
stay safe

* different variation in printing

* print :-

print next statement in "same" line

* println :-

print the next statement in "new" line

Note :- print or println will effect the next statement

It will not effect current statement.

Ex :-

```
public class Revision
{
public static void main(String args[])
{
System.out.print("We did Revision of");
System.out.print("%^$()!@@@###");
System.out.print("Topic is Basic structure of a program");
}
}
```

output:

We did Revision of %^$()!@@@###Topic is Basic structure of a program

Example :

```
public class Revision
{
public static void main(String args[])
{
System.out.println("We did Revision of");
System.out.println("Two classes");
System.out.println("Topic is Basic structure of program");
}
}
```

output :-

We did Revision of
Two classes
Topic is Basic structure of program

```
Public class Revision
{
public static void main(String args[])
{
System.out.print("We did Revision of ");
System.out.println("%^&*())!@@@##");
System.out.println("Topic is Basic Structure of a program");
}
}
```

output:
We did Revision of %^&*())!@@@##
Topic is Basic Structure of a program

```
public class Revision
{
public static void main(String args[])
{
System.out.print("Novel Corona Virus");
System.out.println("is Spreading");
System.out.print("please stay Home");
System.out.println("Stay safe");
System.out.println("Be strong nothing happen if you infected too");
}
}
```

output:

Novel Corona Virus is Spreading
please stay Home Stay safe
Be strong Nothing happen if you infected too

```
public class Revision
{
public static void main(String args[])
{
System.out.print("Novel Corona Virus");
System.out.print("is Spreading");  // System.out.println("is Spreading");
System.out.print("please stay Home");
```

```
        System.out.println("Stay      Safe");
        System.out.print("Be strong Nothing happens if you are infected too");
    }
}
```

output :-

```
Novel Corona Virus is          Spreading
please Stay Home Stay          safe
Be strong Nothing happens if you infected too
```

* \n (Backslash-n) :- should be under " " codes

Immediately move cursor go to next line

```
public class Saturday   or public class Q
{
  public static void main(String args[])
  {
    System.out.println("Do not expect\n Expectations always hurts");
  }
}
```

output :-

```
Do not expect
Expectations always hurts
```

Ex:-

```
public class Q
{
  public static void main(String args[])
  {
    System.out.print("Do not expect\n Expectations always hurts");
    System.out.print("if u want to be Happy");
  }
}
```

output :-

```
Do not expect
Expectations always hurts if u want to be happy
```

```
public class a
{
  public static void main(String args[])
  {
    System.out.print("Do not Expect \n Expectation always hurts");
    System.out.println(" if u want to be happy");
    System.out.print("Take care");
  }
}
```

output :-

Do not Expect
Expectation always hurts if u want to be happy
Take care

* Comment Section

It is used for 2 purpose

1. for not executing any part of program
2. for giving description (meaning of that line)

" if you keep any part of our program under Comments that part will not be Considered for Execution ".

2 types of Comments

Single line Comments

Syntax :- // statement

Ex :- // System.out.println("Hello");

multiple line Comments :-

Comments are used to provide description

```
/*
  System.out.println("Hi");
  System.out.println("Hello");
  System.out.println("Yeah");
*/
```

Comments Example:-

```
public class StudentInfo
{
  public static void main(String args[])
  {
    System.out.println("Name: Pooja");
    // System.out.println("Age: 22 years");
    System.out.println("Email: poo@gmail.com");
    // System.out.println("Contact: 8888888888");
    System.out.println("Insta id: pooja143");
    System.out.println("Address: daager Zoae, hyd-18");
  }
}
```

* We can write anything under " " double quotes it will display) *

* we should not disturb syntax in the program

output:-

```
Name : pooja
Email : poo@gmail.com
Insta id : pooja143
Address : daager zoae, hyd-18
```

```
public class StudentInfo
{
  public static void main(String args[])
  {
    System.out.println("Name: Pooja"); // Name of person
    System.out.println("Age: 22 years");
    /* System.out.println("Email: poo@gmail.com");
    System.out.println("Contact: 8888888888");
    System.out.println("Insta id: // pooja143");
    System.out.println("Address: daager Zoae, hyd-18"); */
  }
}
```

output:-

```
Name : Pooja
Age : 22 years
```

```
public class StudentInfo // this is class declaration
{
  public static void main(String args[]) // this is main method
  {
    System.out.println("Name : pooja"); // name of person
    /* System.out.println("Age : 22 years");
       System.out.println("Email: poo@gmail.com"); */
       System.out.println("Contact : 8888 8888 88");
       System.out.println("Insta id : pooja143");
       System.out.println("Address : Langer zone, hyd-18");
    // System.out.println("Address : Langer zone, hyd-18");
  }
}
```

output :-

```
Name : pooja
Contact : 88888888
Insta id : pooja143
```

* Program :- 1

Write a program to print following input

```
Name : Mr john P
Email : johnp22@gmail.com
  Age : 22 years
Aadhar : 456932145896
  PAN : ANBPK89Y
```

```
public class Myprogram
{
  public static void main(String args[]){
  {
    System.out.println("Name : Mr. john P");
    System.out.println("Email : johnp22@gmail.com");
    System.out.println("Age : 22 years");
    System.out.println("Aadhar : 456932145896");
    System.out.println("PAN : ANBPK89Y");
  }
}
```

Program:-2

write a program to display following output

Candidate Name : Rahul Tripathi
Interview date : 27/may/2021
Interviewer : Mrs divya
Company : HDFC
Address : 14-1-55/6, mind space, Building No-21 Hyd-18.

```
public class Myprogram
{
  public static void main(String args[])
  {
    System.out.println("caadidate Name : Rahul Tripathi");
    System.out.println("Interview date : 27/may/2021");
    System.out.println("Interviewer : Mrs Divya");
    System.out.println("Company : HDFC");
    System.out.println("Address : 14-1-55/6, mind space, Building No-21 Hyd
                        -18");
  }
}
```

Program:-3

Validate below program

```
public class Quiz
{
  public static void main(String args[])
  {
    System.out.println("Correct or Not");
  }
}
```

(a) Correct
(b) Incorrect

Incorrect because in main() we keep M as capital

program:-4

Validate below program

```
public class Quiz
{
  public static void main(String args[])
  {
    System.out.println("Hello world"\n"what's going on");
  }
}
```

(a) Correct (b) Incorrect

It is Incorrect because we can't keep \n without double quotes under two statements

program:-5

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

(a) Correct (b) Incorrect

Incorrect because in class declaration C is kept as Capital

* Architecture of Java

5 - Steps for developing an application in JAVA

1. Select Editor
2. develop a Code
3. Save a Code
4. Compilation
5. Excention

Step 1 :-

→ Select a place to write a program

Ex :- notepad, notepad++, edit++

Ex :- Tools (IDE - Integrated development Environment)

→ Eclipse, NetBeans, Android studio etc

Step 2 :-

Write a program by following Syntax's

step 3 :-

→ Save the program with "classname.java".

→ Always we should give .java as Exteasion

→ for remembering always save as classname.java

When Communication will happen properly

"Know Language"

If there is a problem of understanding what to do

"Translator".

Human	machine
programming language	Binary language / 0's and 1's language

* why do we Compile a program?
  * Compiler is one of the supporting tool
  * To Convert our program into System understable language

Step 4 :-

* To Convert our code into System understable/endable language
* During Compilation Syntax will be checked, if

there is syntax mistake we will get "Compile time error".

* If there is no syntax mistake "byte code file generated"
* for Compilation go to Command prompt
  and enter Command as javac filename.java
* Compilation is done at once
* During Compilation, Compiler will check syntax errors like [ ], ;, { }, :, spelling and case sensitivity

Step 5:-

* for getting output we have to execute the program
* JVM – Java virtual machine is Responsible to execute the programs.
* JVM can identify → public static void main(String args[])
* It is like one software one program
* Execution will happen in line by line manner
* during execution JVM will find logical error of program
* for Execution open cmd prompt type Command as

  java programmname

  Ex:- java Hello

1. Notepad

2.
```
public class Basic
{
 public static void
 main(String args[])
 {
   S.o.p("something");
 }
}
```

3. Save: Basic.java

Syntax:- classname.java

4. Compilation

How to Compile

- go to Command prompt
- → enter Command

javac filename.java

one byte code file generated

→ During the Compilation Syntax will be checked

5. Execution

JVM – java virtual machine

How to execute program

* Go to cmd prompt
* java programmname

Note:-

* for checking java is installed → java - version
* for coming out of folder - - → cd..
* for enter into folder - - - → cd foldername (case sensitivity doesn't matter)

for Compilation ------> javac programname.java

for Execution - - - -> java programname

for Changing drive . . . > drivename:

for Clearing Screen - - - -> cls

* Note:-

E :
cd java
javac Programname.java
java program

Ex:

E :
cd java
javac Student.java
java Student

---

* Data types and Variables

data :- A set of some information is data

Ex:- name, age, percentage, yop, adharno, accno, panno, email, salary, and dob etc

data type :- Type (Category) of information

* In java program we use data types to represent type of data (info)

Data types are divided into two types

1. primitive data types
2. Non primitive data types

Primitive data types (System define datatype)

primitive data types are classified into 8 types

* what are those 8 data types
* when to use which data type
* These are System define data types
* primitive data type are "fixed" in size

datatypes	Raage/usage	Sise (byte) 1byte = 8 bits	Examples
*Numeric (+/-)			
1. byte	-128 to +127	1 byte (very small values)	10, 2, 5, 55, 45, 101
2. short	-32768 to 32767	2 byte	100, 200, 220 - - (-32768 to 32767)
3. int	-2147483648 to 2147483647	4byte	1, 2, 777, 99999 - - (2,147483647)
4. loag	>2147483647	8byte	2222222232 - - - - -
*Decimals			
5. float	upto 6 digits after point	4byte (32.5858)	0.2, 0.3, 33.666 - - -
6. double	more than 6 digits	8byte	0.343434343434, 99.55555555
Character			
7. char	A-Z, a-Z, Special char (Siagle character)	2 byte	A, a, - - -
8. Boolean	True/false	no size/1 byte	True/false

* For Numeric

* $-2^{\text{no of bits}-1}$ to $+2^{\text{no of bits}-1}-1$

1 byte = 8 bits

-2^{8-1} to $+2^{8-1}-1$

-2^{7} to $+2^{7}-1$

-128 to +127

Note:-

* When we want 4-6 digits of accuracy we go float else we use double
* byte, short, int, loag : used for Numericals or Integers
* double and float use for decimals
* When we store value greater than 2,147,438,647 we need to use loag

Non primitive data type :-

* Non primitive data types are not fixed in size

String is a datatype that can Represents alphanumeric, data, email etc names email, PAN no, IFSC Code, Postal Code

Ex:- John, John@gmail.com

Note:- char is used for single characters

* If we have more than "one character" we should use string
* using string we can Represent any data info

Variables

Variables are used to store data

* In order to use variable we have to follow 2 steps

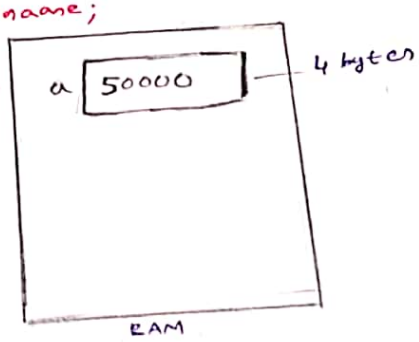
1. Variable declaration :-

When we declare a variable one memory block is created (RAM)

Note:- variablename can be anything (a-z, A-Z, 0-9, $ - -)
and preferably it should in smaller case

Syntax:- datatype variablename;

* Ex:- int a;
  byte b;
  char c;
  boolean d;
  float r;
  String s;



2. Variable Initialisation

When we initiliase a Variable value will be stored in memory block

Syntax:- Variablename = Value;

Ex:- a = 5000;
b = 100;
c = 'J'; // mandatory to give ' ' single quotes
d = true;
r = 544.654f; // mandatory to write f
S = "java145$%#"; // mandatory to give " " double quotes

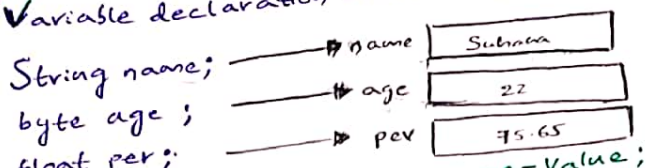
Important points to be followed

* when we initialise character value we should keep it under "single quotes".
* When we initialise string value we should keep it under "double quotes".
* when we initialise "float value" we should keep "f after value".

Program:-1

Write a simple program to print name, age, and percentage of a student using data types and variables

```
public class StudentInfo
{
  public static void main(String args[])
  {
    // using datatypes and variables
    // Variable declaration ---> datatype varname;
    String name;
    byte age;
    float per;
    // and Var initialisation ---> Varname = value;
      name = "Suhana";
      age = 22;
      per = 75.65f;
    // print statement
      System.out.println(name);
      System.out.println(age);
      System.out.println(per);
  }
}
```



* output:-

Suhana
22
75.65

*Note:-

Syntax for printing value stored in var is ---->

System.out.println(variablename without double quotes);

(2)

```
Public class StudentInfo
{
  public static void main(String args[])
  {
    //Variable declaration + Variable initilisation ---> datatype varname=Value;
    //var dec + var initilisation ---> datatype varname=value;

      String name = "suhana";
      byte age = 22;
      float per = 65.75f;

    //print statement

      System.out.println(name);
      System.out.println(age);
      System.out.println(per);

  }
}
```

output:-

Suhana
22
65.75

Program:-2

Write a program to print name of book, author, colour, no of pages using data-types and variables

```
public class Bookinfo
{
  public static void main(String args[])
  {
    // var dec + var initialisation - -> datatype varname = value;
       String bname = "Thermodynamics";
       String bauthor = "R.K. Rajput";
       String bcolour = "Black";
       Short bpages = 5000;
    // print statement
       System.out.println(bname);
       System.out.println(bauthor);
       System.out.println(bcolour);
       System.out.println(bpages);
  }
}
```

output:-

```
Thermodynamics
R.K. Rajput
Black
5000
```

```
public class Bookinfo
{
  public static void main(String args[])
  {
    // var dec + var initialisation - - -> datatype varname = value;
    String bname = "Thermodynamics", bauthor = "R.K. Rajput" // b colour = "Black";
      Short bpages = 5000;
    Short b
       System.out.println(bname);
       System.out.println(bauthor);
       System.out.println(bcolour);
       System.out.println(bpages);
  }
}
```

output:-

```
Thermodynamics
R.K. Rajput
Black
5000
```

Program:-3

write a program to print following statements information Consider your data as example

Studentname
Collegename
Collegeaddress
Studentage
studentpercentage
Studentyop
StudentContact

```
Public class Stdinfo
{
  public static void main(String args[])
  {
    String Sname="vikas", Cname="VBIT", Cadd="144/1, ghatkesar, hyderabad";
    byte Sage=22;
    float Sper=63.37f;
    short yop=2018; // (or) we can use int yop=2018;
    long Contact=8501045844l; //use l after number
    System.out.println(Sname);
    System.out.println(Cname);
    System.out.println(Cadd);
    System.out.println(Sage);
    System.out.println(Sper);
    System.out.println(yop);
    System.out.println(Contact);
  }
}
```

8501045844l

output:-

```
Vikas
VBIT
144/1, ghatkesar, hyderabad
22
63.37
2018
8501045844
```

Note:-

Rules with Reasons

* Every numeric value is defaultly treated as Integer so if we store the value as long if the value is greater than 2147483647 we have to add l to indicates its long

Ex:- long num = 2147483646
long num = 2147483647
long val = 21474836481 // Compulsory to add l

* Every decimal value is defaulty treated as double so inorder to indicate it is float we have to add f

Ex:- float r = 667.8787f;

Program-14

* Write a program to print following info

accountnumber - - - 658745687
Bankname - - - - ICICI
bankaddress - - - - 1145/sector-2, kukatpally, hyderabad
IFSC code - - - - - ICI002548
accountholdername - - - Mr john P
availableamt - - - - 55000.0 — or 55000.367
minimumbalance - - - - - 3000

```
public class Bankinfo
{
 public static void main(String args[])
 {
  long accno = 658745687; // here we can keep d also
  String bname = "ICICI", badd = "445/sector-, kukatpally, Hyderabad",
         ifsc = "ICI002548", accname = "Mr john p";
  float avalamt = 55000.0f;
  short minbal = 3000; //int minbal = 3000;
  System.out.println("Bank information");
  System.out.println(accno);
  System.out.println(bname);
  System.out.println(badd);
  System.out.println(ifsc);
  System.out.println(accname);
  System.out.println(avalamt);
  System.out.println(minbal);
 }
}
```

output:-

```
Bank information
658745687
ICICI
445/sector-, kukatpally, Hyderabad
ICI002548
Mr John p
55000.0
3000
```

* Operators *

1. Airthmetic operators

+ ⟶ Addition
- ⟶ Substraction
* ⟶ multiplication
/ ⟶ division (output = Coeffiecent)
% ⟶ mode (output - Reanainder)

5) 12 (2
10
(2)

Special operator - - → +

addition

Concatination

+ operator as Concatination

Ex:- a + b

a ⟶ operand
b ⟶ operand
+ ⟶ operator

* out of a and b if anyone of them is string type + operator will acts as Concatination and Result value is "string type".

```
public class Trial
{
    public static void main(String args[])
    {
        String name="java";
        int cost=500;
        System.out.println("Name of the Book :" +name);
    }
}
```

* output:-

Name of the Book : java

```
// SOP(name); ⟶ java
// SOP("name"); ⟶ name
```

*

```
Public class Trial
{
  public static void main(String args[])
  {
    String name="java"
    int cost = 500;
  System.out.println("Name of book is :" +name+" It's cost is :" + cost + "Rs/-");
  }
}
```

// String String String int String

➡ output:-

Name of the Book : java it's cost is : 500 Rs/—

*

```
Public class Trial
{
  public static void main(String args[])
  {
    String name="java";
    int cost = 500;
    byte rating=5;
  System.out.println(name+"\n" + cost +"\n" +rating);
  }
}
```

// "\n" ---> immediately Break the line

output :-

java
500
5

2. Comparison operator (==)

* It is used to Compare two values
* If the values are same output will be true
* If the values are not same output will be false
* It is Represented as ==

Program

```
public class Eq                                    // (space)
{
public static void main(String args[])
{
   int a=100, b=200;                               =  storing
   boolean res=(a==b);                             ==  Comparing
   System.out.println(res);
   int c=500, d=500;
   boolean des=(c==d);
   System.out.println(des);
}
}
```

Output:-

False
True

3. Assignment operator (=)

* It is used to "store" or "assign" the value into a variable
* It works as right to left

```
    int a=200, b=300;
    int c=a+b;

public class Assignment
{
  public static void main(String args[])
  {
    int a=200, b=300;
    int c=a+b;
    System.out.println(c);
  }
}
```

Output:-

500

4. Relational operators

It checks relation between two values

- `>` ——→ greater than
- `<` ——→ lesser than
- `>=` ——→ greater than or equals
- `<=` ——→ lesser than or equals
- `!=` ——→ not equals

```
public class Relational
{
  public static void main(String args[])
  {
    int a = 100, b = 200;

    boolean res = (a > b)
    boolean des = (a < b)

    System.out.println(res);
    System.out.println(des);
  }
}
```

output:-
false
true

5. Logical operators

It Compares logical Relationship such as

1. AND (&&)

* If both the input are true then only output is true otherwise false
* If anyone is false or both are false then output will be false
* Represented as &&

Input 1	Input 2	&& AND
true	true	true
true	false	false
false	true	false
false	false	false

```
public class Logical
{
  public static void main(String args[])
  {
    int a=100, b=200, c=52, d=90;
    boolean res=(a<b && c<d);
    System.out.println(res);
    boolean des=(a>b && c<d);
    System.out.println(des);
  }
}
```

output:-
true
false

2. OR (||) :-

* If anyone is true output is true otherwise false
* if both are false then only output is false
* Represented as ||

⇒ *

input 1	input 2	||(OR)
true	true	true
true	false	true
false	true	true
false	false	false

```
public class Logical
{
  public static void main(String args[])
  {
    int a=100, b=200, c=52, d=90;
    boolean res=(a<b || c<d);      // true || true
    System.out.println(res);
    boolean des=(a>b || c<d);      // false || true
  }
}
```

output:-
True
True

3. Not (!)

we will study in Conditional programming

Note :-

Every character in java will be stored in UTF (unicode transfor format) ; i e

** for every character there it a unicode associated to it

A - 65, B - 66, C - 67 - - - - Z - 90

a - 97, b - 98, C - 99 - - - - Z - 122

* Note :-

when we add two characters JVM will Consider there unicode values and perform addition

Ex:- Char ch = 'A'

Char chh = 'B'

System.out.println(ch + chh); // 131

Note:-

when we normally print values of character JVM will print character only not unicode for that character

```
Char res = 'A';
System.out.println(res); // A
```

```
public class Character
{
  public static void main(String args[])
  {
    char ch = 'A';
    char chh = 'B';
    System.out.println(ch + chh);
  }
}
```

⇒ * output

131

*Unary operators

Increment operator:-

* It is used to Increase the value by 1
* It is denoted as ++

1. pre Increment:-

Rule:- for pre increment first Increase the value, then assign/store/print
then update the incremented value

It is represented as ++variable name

program:-1

```
public class Increment
{
  public static void main(String arg:[])
  {
    int a=100;
    int b=++a;
  System.out.println(b);
  System.out.println(a);
  }
}
```

1. Increase
2. assign
3. update

output:-

101
101

program:-2

```
public class Increment
{
  public static void main(String args[])
  {
    int a=100;
    int b=++a;
    int c=a+++b;

  System.out.println(a);
  System.out.println(b);
  System.out.println(c);
}
}
```

output:-

101
102
203

2. post Increament

Rule:- for post increament assign/store/print then increase the value, then update the incremented value

* It is Represented as Variablename++

*

Program:-1

```
Public class Increment
{
  public static void main(String args[])
  {
     int a = 100;
     int b = a++;
   System.out.println(a);
   System.out.println(b);
  }
}
```

1. Assign/store
2. increement
3. update

output:-
101
100

```
public class Increment
{
  public static void main(String args[])
  {
     int a=100;
     int b = a++;
     int c = ++a + b++;

   System.out.println(a);
   System.out.println(b);
   System.out.println(c);
  }
}
```

output:
102
101
202

* Decreament operator

* It is use to decrease the value by 1
* It is denoted as --
  1. pre decreament
  2. post decreament

1. pre decrement:-

Rule:- For pre decrement first decrease the value, then assign/store/ print then update the decrement value

* It is represented as --variablename

Program:-1

```
public class Decrement
{
   public static void main(String args[])
   {
      int a = 100;
      int b = --a;
    System.out.println(a);
    System.out.println(b);
     }
}
```

output:-
99
99

Program :- 2

```
public class Decrement
{
   public static void main(String args[])
   {
      int a = 100;
      int b = --a;
      int c = a - --b;
    System.out.println(a);
    System.out.println(b);
    System.out.println(c);
     }
}
```

output:-
99
98
1

2. post decreament :-

Rule:- for post decreament assign/store/print then decrease the value, then update the decreament value

It is Represented as Variablename--

Program 1:-

```
public class Decrement
{
public static void main(String args[])
 {
    int a = 100;
    int b = a--;
System.out.println(a);
System.out.println(b);
 }
}
```

output:-

99
100

* Program: 2

```
public class Decrement
{
  public static void main(Stiag args[]).
  {
     int a = 100;
     int b = a--;
     int c = --a - b--;

  System.out.println(a);
  System.out.println(b);
  System.out.println(c);

  }
}
```

output:-

98
99
-2

Identifiers :-

Identifiers are the names given by the programmer for Identification

Ex :- classname, variablenaame, methodnaame, packagenaame, projectnaame

Note :- methodnaame, packagenaame, and projectnaame (we will discuss in future)

Rules for defining Identifiers

1. An Indentifer should be a Combination of A-Z, a-z, 0-9, $, _
2. An Indeatifier should not start with a digit
3. If an Ideatifier have more than one word we should not give space between words
4. An Indeatifer Cannot be a keyword

Examples :-

* public class S@mple , int a#gW=211; ⟶ Invalid (because @ and # is invalid symbols)
* public class $ample_123, int a_ge11 = 55; ⟶ Valid (because $ and _ are valid Combination)
* public class 22Details, int 2age = 30; ⟶ Invalid (Identifier Can't start with digit)
* public class My Info, int my age=22; ⟶ Invalid (Because Space are there)
* public class static, int class = 200; ⟶ Invalid (because class and static are keywords)
* public class A1 ⟶ Valid
* public class MyDetails, int myperceatage=50; ⟶ Valid (because no violation of Rule)

Coding Standards ⟹

* It is not Compulsory to follow, if you follow its good otherwise No problem

1. Class naame should start with CAPITAL LETTER
2. variable naame should start with SMALLER LETTER
3. Ideatifier should not be more than 15 letters
4. If a classnaame has more than word for every first letter keep as CAPITAL

Ex :- public class MyInfoForColl;

5. If a variable name has more than one word from second word, every word first letter should be CAPITAL

ex: int myAgeIs = 22;

*

```
public class Revision
{
  public static void main(String args[])
  {
    int i=300, j=400;
    long con=9847856235l;
    System.out.println(i);
    System.out.println("Contact Number is: "+con);

    boolean b1=(i==j)
    System.out.println(b1);

    int a=5000, b=2000;
    boolean c=((a>b) || (a!=b)); //true || true

    int d=a+b++; //5000+2000

    System.out.println(d); //7000

    //String @name="suhan"; Invalid identifier

    //String MYNAME="Rohan"; Valid but standard is not there
    //String MyName="Rohan"; Valid standard is there
      char ch='A', ch1='b';
      int sub=(ch1-ch);
      System.out.println(sub);

  }
}
```

* output:-

```
300
Contact Number is: 9847856235
false
true
7000
33
```

Control / Decision / Conditional Statements:

* When we want to take a decision in our program we will use Decision

If else

Syntax :-

```
if(condition)
{
    statement1;
}
else
{
  statement2;
}
```

⇒ * if and else are keywords
* Condition should must be of "Boolean Type".
  if(100>200), if(200==200), if(a!=b)
  if(a=100) —— Invalid if(a==100) Valid
* If Condition is true then only statement1 excented
* if Condition is false then only statement2 excented
* "Never Ever" statement1 and statement2 will be excented together
* It is not Compulsory to write else
* {} ——→ Boundary of if and else
* if we have only one Statement we can skip boundary

program 1 :-

1. write a program to check whether atm pin is 7426 or not if it is 7426 display the output as welcome to our Bank otherwise display as Invalid pin

```
Public class Pin
{
  public static void main(String args[])
  {
    int atm_pin=7426;
    if(atm_pin==7426)                    if(atm_pin==7426)
    {
      System.out.println("Welcome to our bank");
    }
```

```
        else
        {
          System.out.println("Invalid pin");
        }
    }
}
```

output:-
welcome to our bank

```
public class Pin
{
  public static void main(String args[])
  {
    int atm_pin = 8456;
    if(atm_pin==7426)
    {
      System.out.println("Welcome to our bank");
    }
    else
    {
      System.out.println("Invalid pin");
    }
  }
}
```

output:-
Invalid pin

Program:-2 :-
write a program to check whether a person is eligible to vote or not

```
public class Vote
{
  public static void main(String args[])
  {
    int age=55;
    if(age>=18)
    {
      System.out.println("As per Rules person is eligible to vote");
    }
    else
    {
      System.out.println("Not Eligible");
    }
  }
}
```

output:-
As per Rules person is eligible to vote

program:-3 :-

write a program to find greatest of two numbers

```
public class Greatest
{
  public static void main(String args[])
  {
    int a=1000, b=250;
    // for taking decision
    if(a>b)
    {
      System.out.println(a+" is the greatest of 2no's");
    }
    else
    {
      System.out.println("b +" is the greatest of 2 no's");
    }
  }
}
```

output :-
1000 is the greatest of 2no's

program:-4

write a program to check whether person is eligible for IAS or not

eligibility criteria is age should be between 22 to 35

```
public class IAS
{
  public static void main(String args[])
  {
    int age=45;
    if(age>22 && age<35)
    {
      System.out.println("person is eligible for IAS");
    }
    else
    {
      System.out.println("not eligible for IAS");
    }
  }
}
```

output :-
Not eligible for IAS

* Program-5

write a program to check whether 22 is an even or odd number

```
public class EvenOdd
{
  public static void main(String args[])
  {
    int num = 22
    if(num%2==0)
    {
      System.out.println(num+" is an Even number");
    }
    else
    {
      System.out.println(num+" is an odd number");
    }
  }
}
```

even nber

(if) int num = 23
if(num%2!=0) → odd number

* output:-
22 is an even number

if num = 23
if (num%2!=0) or if (num%2==1)

output:- 23 is an odd number

* If – else if – else :-

Syntax:-

```
if(condition1)
{
   statement1;
}
else if(condition2)
{
   statement2;
}
else if(condition3)
{
   statement3;
}
   ---- so on ----
else
{
   statement;
}
```

→ statement 1 will executed only when Condition 1 is true
→ statement2 will be executed only when Condition 2 is true
→ statement3 will be executed only when Condition 3 is true
→ none of the Condition is true statement will be executed

Note:-

* Never ever all the statement will be executed together

* program:-1

```
public class Greatest
{
   public static void main(String args[])
   {
      int a= -55, b = 250;
      if(a>b)
      {
         System.out.println(a + " is the greatest of 2 no's");
      }
```

```
        else if(b>a)
        {
          System.out.println(b+" is the greatest of 2 no's");
        }
        else
        {
          System.out.println(a+" and "+b+" may be equal numbers");
        }
    }
}
```

output:-

250 is the greatest of 2 no's

* program:2

write a program to find the greatest of 3 numbers

a b c

```
public class Greatest
{
  public static void main(String args[])
  {
    int a=45, b=22, c=35;
    if(a>b && a>c)
    {
      System.out.println(a+" is the greatest of 3 no's");
    }
    else if(b>a && b>c)
    {
      System.out.println(b+" is the greatest of 3 no's");
    }
    else if(c>a && c>b)
    {
      System.out.println(c+" is the greatest of 3 no's");
    }
    else
    {
      System.out.println("May be all are equal numbers or any 2
                          of them are equal");
    }
  }
}
```

* output:-

45 is the greatest of 3 no's

* if input is — See the before query program for reference

int a=55, b=55, c=55

output :-

may be all are equal nbers or any 2 of them are equal

* if input is

int a=45, b=55, c=55

output :-

May be all are equal nbers or any (2) of them are equal

---

program :- 3 :-

write a program to check whether -88 is positive or negative number

```
Public class PasitiveNegative
{
  public static void main(String args[])
  {
    int num=-88;
    if(num>0)
    {
      System.out.println(num+" positive number");
    }
    else
    {
      System.out.println(num+" Negative number");
    }
  }
}
```

* output :-
-88 Negative number

program :- 4 :-

write a program to check whether 55 is multiple of 5 or not

```
public class Multiple
{
  public static void main(String args[])
  {
    int num=55
    if(num%5==0)        // if(num%5==0)
    {
      System.out.println(num + " is multiple of 5");
    }
```

```
        else
        {
          System.out.println(a+" is not multiple of 5")
        }
    }
}
```

output:-
55 is multiple of 5

program:-5

Write a program to check whether E is vowel or Consonant?

```
public class English
{
  public static void main(String args[])
  {
    Char ch='E';
    if(ch=='A'||ch=='E'||ch=='I'||ch=='O'||ch=='U')
    {
      System.out.println(ch+" is vowel");
    }
    else
    {
      System.out.println(ch+" is Consonant");
    }
  }
}
```

output:-
E is vowel

program:-6 :-

Generate the bill amount Consider following info

Amount	discount
<500	0%
>500 and <1000	25%
>1000 and <1500	50%
>1500	60%

take amount as input by your choice and display final bill amount according to data

```
public class Bill
{
   public static void main(String args[])
   {
      int amt = 1500;
      float final_amt;
      if(amt <= 500)
      {
         final_amt = amt;
         System.out.println(final_amt);
      }
      else if(amt > 500 && amt <= 1000)
      {
         final_amt = amt - (amt * 25/100);
         System.out.println(final_amt);
      }
      else if(amt > 1000 && amt <= 1500)
      {
         final_amt = amt-(amt * 50/100)
         System.out.println(final_amt)
      }
      else
      {
         final_amt = amt - (amt * 60/100)
         System.out.println(final_amt)
      }
   }
}
```

output:-
750.0

* write a program to print JAVA 10 times

```
public class J10
{
    public static void main(String args[])
    {
        System.out.println("Java");
        System.out.println("Java");
        System.out.println("Java");
        System.out.println("Java");
        System.out.println("Java");
        System.out.println("Java");
        System.out.println("Java");
        System.out.println("Java");
        System.out.println("Java");
        System.out.println("Java");
    }
}
```

above we can write 10 times

if the program is WAP to print JAVA "500 times". "here we can't" have we go to looping Concept

*Imp

Looping Statements or Iterative Statements

If a part of a Code is repeatedly coming, then rather than writing multiple times we can write it once and run as many times as we want by using loops

Loops are divided into 4 types

1. for loop
2. while loop
3. do while loop
4. for each loop / enhanced loop / advance loop

*Important things from Interview

programs on for loop and while loop

Important things from learning prospective

* Syntax and how loop will work
* Prequisite: datatype, variables, operators (complete info), Conditional statements

* Syntax:-

```
for(initialisation; condition; inc/dec)
{
    statement1;
    statement2;
    - - - - -
    - - - - -
    - - - - -
    statementn;
}
```

(Excecution till false condition evaluated)

* for ⟶ keyword
* { } ⟶ boundary

* Hint:-

initialisation Condition updation

* Working:-

Step-1 :- initialisation (varname = value)

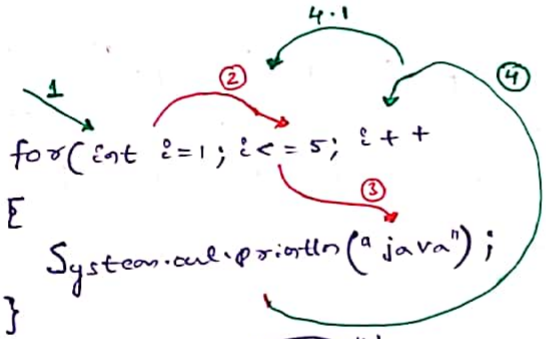
Step-2 :- checking condition (if true ⟶ excecute step-3)
(if false ⟶ excecute step-5)

Step-3 :- excecute statement1, statement2 - - - statementn

Step-4 :- perform updation (increment or decrement) and go to step 2

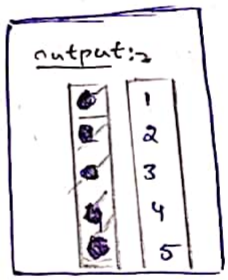
Step 5 :- Come out of loop

*

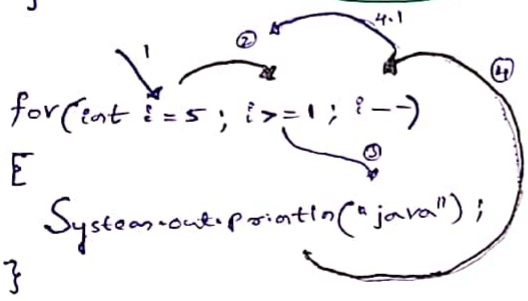



```
for(int i=1; i<=5; i++)
{
  System.out.println("java");
}
```

*



*




```
for(int i=5; i>=1; i--)
{
  System.out.println("java");
}
```

output:-

5
4
3
2
1

(1) *

```
for (int j=1; j>=1; j++)
{
  System.out.println("Loop");
}
```

output:-

1, 2, - - - - - - - ∞

a) 3, 2, 1 b) 1, 2, 3 c) 3 d) Nothing will print

e) Infinite times loop ⇒ for stopping infinite loop type ctrl - c

```
for(int K=3; K<=1; K++)
{
   System.out.println("Loop");
}
```

(3<=1) ??

a) 3,2,1 b) 1,2,3 c) 3 d) Nothing will be print

* for stopping infinite loop type ctrl-c

program :-1 :-

(1.) write a program to print 1 to 100

```
public class Loop1
{
  public static void main(String args[])
  {
    for(int i=1; i<=100; i++)
    {
      System.out.print(i+" ");
    }
  }
}
```

output :-

1 2 3 4 5 6 7 8 9 10
11 12 13 14 15 16 17 18 19 20
— — — — —
— — — — —
— — — — 98 99 100

// initialisation ⟶ starting point
// Condition ⟶ ending point
// updation ⟶ starting to ending what we are doing

(2.) write a program to print 65 to 35

```
public class Loop1
{
  public static void main(String args[])
  {
    for(int i=65; i>=35; i--)
    {
      System.out.print(i+" ");
    }
  }
}
```

* output :-

65 64 63 62 61 60 59 58
57 56 55 54 — — — —
— — — — — —
— — — — 37. 36. 35

③ write a program to print all even numbers from 1 to 50?

```
Public class Loop1
{
  public static void main(String args[])
  {
     for(int i=1; i<=50; i++)
     {
        if(i%2==0)
        {
           System.out.println(i)
        }
     }
  }
}
```

output:-
2
4
6
8
10
:
50

} All even numbers 1 to 50

Program

④ write a program to print Count of all even numbers from 1 to 50?

```
public class Loop1
{
  public static void main(String args[])
  {
     int count=0;
     for(int i=1; i<=50; i++)
     {
        if(i%2==0)
        {
           count++;  // count = count+1
        }
     }
     System.out.println("Count is :"+count);
  }
}
```

or

```
int i, count=0;
for(i=1; i<=50; i++
```

output:-
Count is : 25

Program:-

⑤ write a program to print sum of all even number from 1 to 50 ?

```
public class Loop1
{
    public static void main(String args[])
    {
        int sum=0;
        for(int i=1; i<=50; i++)
        {
            if(i%2==0)
            {
                sum=sum+i;
            }
        }
        System.out.println("sum is :" + sum);
    }
}
```

```
int i, sum=0;
for(i=1; i<=50; i++)
```

output:-
sum is : 650

program:-6 :-

Write a program to print product of number from 1 to 10

1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 * 10

```
public class Loop1
{
    public static void main(String args[])
    {
        int pdt=1;
        for(int i=1; i<=10; i++)
        {
            pdt=pdt*i
        }
        System.out.println("product of 1-10 numbers is :" + pdt);
    }
}
```

~~int i, pdt=1;~~

```
int i, pdt=1; // product
int i, product pdt=1;
for(i=1; i<=10; i++)
```

output:-
product of 1-10 number is : 362800

Program:-7 :-

write a program to print all numbers from 1 to 100 excluding numbers which are divisible by 7 ?

```
public class Loop1
{
  public static void main(String args[])
  {
    for(int i=1 ; i<=100; i++
    {
      if(i%7!=0)
      {
        System.out.print(i+" ");
      }
    }
  }
}
```

* output:-

1 2 3 4 5 6 8 9 10 11 12 13
15 16 17 18 19 20 22 23 24
25 26 27 29 30 31 32 33 34 36
- - - - - - - -99,100.

* Program:- 8

write a program to swap two numbers using extra variable

Temp

input a=10, b=20
output a=20, b=10

```
public class Loop1
{
  public static void main(String args[])
  {
    int a=10, b=20, c;
    System.out.println("Before swap");
    System.out.println(a);
    System.out.println(b);
        c=a;
        a=b;
        b=c;
    System.out.println("After swap");
    System.out.println(a);
    System.out.println(b);
  }
}
```

output:-

Before swap
10
20
After swap
20
10

Program:-9

Write a program to swap two numbers without using extra variable?

Imp

```
public class Loop1
{
    public static void main(String args[])
    {
        int a=10, b=20;
        System.out.println("Before swap");
        System.out.println(a);
        System.out.println(b);
            a = a+b;
            b = a-b;
            a = a-b;
        System.out.println("After swap");
        System.out.println(a);
        System.out.println(b);
    }
}
```

output:-

```
Before swap
10
20
After swap
20
10
```

Imp

Program:-10 :-

Write a program to print fibonacci series

0 1 1 2 3 5 8 13

```
public class Loop1
{
    public static void main(String args[])
    {
        int t1=0, t2=1, t3;
        System.out.print(t1 + " ");
        System.out.print(t2 + " ");
    //i=1 to i<= how many times we are doing t1+t2
        for(int i=1; i<=6; i++)
        {
            t3 = t1 + t2;
            System.out.print(t3 + " ");
            t1 = t2;
            t2 = t3;
        }
    }
}
```

output:-

```
0 1 1 2 3 5 8 13
```

* Loop
* Program:-11 :-

Write a program to check 22 is prime number or not

Prime number is a number which is divisible only 1 and itself

Ex:- 2, 3, 5, 7, 11, 13, 17, 19

Steps:-

1. num=22 Count=0
2. for loop (i=1 to i=22)
3. Condition for divisibility
4. if divisble make a Count
5. after loop give Condition if(count==2) ⇒ prime number

```
public class Loop1
{
  public static void main(String args[])
  {
    int num=22, count=0;
    // checking with how many number it is divisible
    for(int i=1; i<=num; i++)
    {
      if(num%i==0)
      {
        count++
      }
    }
    if(count==2)
    {
      System.out.println(num+" is a prime Number");
    }
    else
    {
      System.out.println(num+" is not a prime Number");
    }
  }
}
```

or

```
int i, num=22, count=0;
for(i=1; i<=num; i++)
```

output:-

22 is not a prime number

* if 23

output will be 23 is a prime number

program:-12 :-

Write a program to print the value 7! (factorial)

Tag

```
public class Loop1
{
    public static void main(String args[])
    {
        int num=7, fact=1;
        for(int i=num; i>=1; i--)   // for(int i=1; i<=7; i++)
        {
            fact = fact * i;
        }
        System.out.println(num+ "Factorial value is :" + fact);
    }
}
```

```
int i, num=7, fact=1;
for(i=num; i>=1; i--)
```

output:-
7 factorial value is 5040

program:-13

write a program to print all number from 1 to 100 which has digit 5 in it

op: 5 15 25 35 45 50 51 52 53 54 55 56 57 58 59 65 75 85 95

* Hint:

one digit / two digit / three digit number / four digit

* If you divide with 10 then Remainder you will get as last digit and quotient you will get as Remaining digits

Ex:- 10) 25 (2 (5)

10) 113 (11 (3)

10) 50 (5 (0)

* How to get Remainder Value → %
* How to get Quotient Value → /

25/10 Quo - 2
25%10 Rem - 5

Quo - 11
Rem - 3

Quo - 5
Rem - 0

10) 25 (2 (5) 1 → %

* Ques

⇒ public class SatProgram

```
{
  public static void main(String args[])
  {
    for(int i=1; i<=100; i++)
    {
      // (5 is) in last place || 5 is, in first place)
      if(i%10==5 || i/10==5)
      {
        System.out.println(i);
      }
    }
  }
}
```

* output:-

5 15 25 35 45 50 51
52 53 54 55 56 57 58
59 65 75 85 95

* Program:-14

Write a program to print multiplication table of 5

Public class MultiplicationTable

```
{
  public static void main(String args[])
  {
    int i, num=5
    for(i=1; i<=10; i++)
    {
      System.out.println(num + " * " + i + " = " + (num * i));
    }
  }
}
```

or

```
int num=5;
for(int i=1; i<=10; i++)
```

output:-

5 * 1 = 5
5 * 2 = 10
5 * 3 = 15
:
:
5 * 10 = 50

program:-15

* write a program to print Count of all number divisible by 5 present b/w 22 to 65

```
public class Loop1
{
  public static void main(String args[])
  {
    int i, Count = 0;
    for(i = 22; i <= 65; i++)
    {
      if(i%5 == 0)
      {
        count++; // count = Count+1
      }
    }
    System.out.println("Final Count is: "+Count);
  }
}
```

```
int count = 0
for(int i = 22; i <= 65; i++)
```

output:-

Final Count is: 9

program-16

* write a program to print numbers -32 to -65

```
public class Loop1
{
  public static void main(String args[])
  {
    int i;
    for(i = -32; i >= -65; i--)
    {
      System.out.println(i);
    }
  }
}
```

(or) for(int i = -32; i >= -65; i--)

output:-

-32	-33	-34	-34	-35	-36
-37	-38	-39	-40	-41	-42
-43	-44	-	-	-	-
-	-	-	-	-	-65

* Program:-17 :-

* Write a program to print sum of first 10 natural numbers

1 + 2 + 3 + 4 - - - +10 = ?

```
public class Loop1
{
   public static void main(String args[])
   {
      int i, sum = 0
      for(i = 1; i <= 10; i++)
      {
         sum = sum + i;
      }
      System.out.println("sum of first 10 Natural number is "+sum);
   }
}
```

* output:-

Sum of first 10 Natural number is 55

Program:-18:-

Write a program to print product of all Even numbers from 1 to 10?

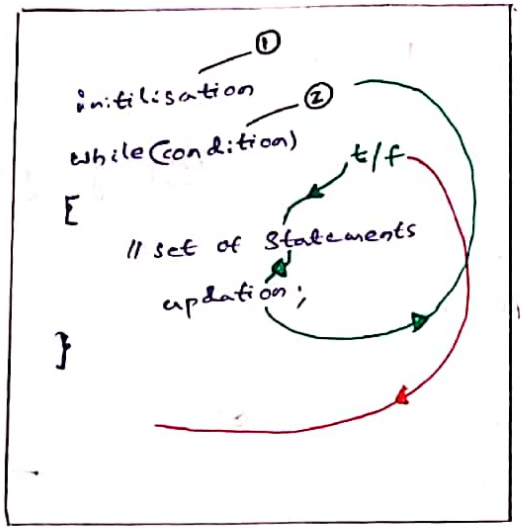
```
public class Loop1
{
   public static void main(String args[])
   {
      int i, product = 1
      for(i = 1; i <= 10; i++)
      {
         if(i%2 == 0)
         {
            product = product * i
         }
      }
      System.out.println("final product value is :" + product);
   }
}
```

* output:-

final product value is 3840

*While Loop :-

```
initialisation;
while(condition)
{
    // set of statements
    updation;
}
```



step1 :-

Initialisation

step2 :-

Checking Condition if true go to step 3 if false execute step 4

step3 :-

Execute set of statements and perform updation goto step 2

step 4 :-

Come out of loop

*programs :-1 :

write a program to print 10 to 90

```
public class WhileLoop
{
  public static void main(String args[])
  {
     int i=10;
     while(i<=90)
     {
        System.out.print(i+" ");
          i++;
     }
  }
}
```

* output :-

10	11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30	31
-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-
-	-	.	-	-	-	-	-	89	90	

Program:-2

write a program to print 66 to 36?

```
public class WhileLoop
{
   public static void main(String args[])
   {
      int i=66;
      while(i>=36)
      {
         System.out.print(i+" ");
         i--
      }
   }
}
```

output:-

66 65 64 63 62 61 60 59 58
57 56 54 53 52 51 50 49 48
47 46 - - - - - - -
- - - - - - -37 36

Program:-3

write a program to print -32 to 63

```
public class WhileLoop
{
   public static void main(String args[])
   {
      int i=32;
      while(i>=-63)
      {
         System.out.print(i+" ");
         i--
      }
   }
}
```

output:-

-32 -33 -34 -35 -36 -37 -38
-39 -40 -41 - - - - -
- - - - - - - -
- - - - - - -62 -63

Imp

Program:-4

write a program to find Reverse of a number

Algorithm:-

→ step1:- take a num=123, rem, rev=0;

→ step2:- Take while loop and give Condition as (num>0)

→ step3:- mode num with 10 and store it rem variable

→ step4:- divide num with 10 and store in it num variable

→ step5:- rev*10+rem store in it rev

```
public class Reverse
{
  public static void main(String args[])
  {
    int num=123, rem, rev=0;
    while(num>0)
    {
      rem = num%10;
      num = num/10;
      rev = (rev*10)+rem;
    }
    System.out.println("Reverse of a number is:"+rev);
  }
}
```

output:-

Reverse of a number is : 321

* Working:-

num = 123 rev = 0

123 > 0 (first time loop runs)

- - - -

rem = 123 % 10 ⟶ 3

num = 123/10 ⟶ 12

rev = 0*10+3 ⟶ 3

rem = 3

num = 12

rev = 3

12 > 0 (second time loop runs)

- - -

rem = 12 % 10 ⟶ 2

num = 12/10 ⟶ 1

rev = (3*10)+2 ⟶ 32

rem = 2

num = 1

rev = 32

1 > 0 (third time loop runs)

- - - - - -

rem = 1 % 10 ⟶ 1

num = 1/10 ⟶ 0

rev = (32*10)+1 ⟶ 321

rem = 1

num = 0

rev = 321

0 > 0 - - → false

Paliqdroase nuasber:-

Imp If origiaal auaber aad reverse of a auaber are saase Such auaber is called as paliadroase auaber

Ex:-

origiaal auaber	Reverse Number
121	121
1441	1441
4994	4994

program:-5

- write a program to read the paliadroase of a auaber

```
public class Paliadroase
{
  public static void aain(String args[])
  {
     int auas=1441, reas, rev=0;
     int backup = auas;

     while(auas>0)
     {
        reas = auas%.10
        auas = auas/10
        rev = (rev*10)+reas;
     }
       System.out.println("Actual Auaber is :" +backup);
       System.out.println("Reverse of a Auaber is :" +rev);
       if(backup==rev) //1441=1441
      {
        System.out.println("paliadroase auaber");
      }
       else
       {
          System.out.println("Non paliadroase auaber");
       }
  }
}
```

output:-

Actual Auaber is : 1441
Reverse of a Auaber is : 1441
paliadrome Number

Armstrong Number:-

Imp

If we cube every digit of a number and add them we should get same number

Ex:- 153 → 1*1*1 + 5*5*5 + 3*3*3 = 153

program:-6

```
public class Armstrong
{
  public static void main(String args[])
  {
     int num=153,rem,arm=0;
     int backup=num;
     while(num>0)
     {
        rem = num%10;
        num = num/10;
        arm = arm + (rem*rem*rem);
     }
      System.out.println("Actual Number is: " + backup);
      System.out.println("Individual Cube and sum result is: " + arm);
      if(backup == arm) //1441 = 1441
      {
         System.out.println("Armstrong Number");
      }
      else
      {
        System.out.println("Non Armstrong Number");
      }
  }
}
```

⇒ output:-

```
Actual number is : 153
Armstrong number is : 153
Armstrong number
```

working:-

num=153 arm=0

while(153>0)

rem = 153%10 → 3

num = 153/10 → 15

arm = 0 + (3*3*3)

num = 15

arm = 27

while(15>0)

rem = 15%10 → 5

num = 15/10 → 1

arm = (3*3*3) + (5*5*5)

num = 1

arm = 152

while(1>0)

rem = 1%10 → 1

num = 1/10 → 0

arm = 152 + (1*1*1)

num = 0

arm = 153

while(0>0) — false

* Differences between for loop and while loop :-

for loop	while loop

Syntax:-
for(initilication; condition; inc/dec)
{
statement 1;
statement 2;
!
Statement n;
}
	Syntax:-
	initialisation;
	while (condition)
	{
	set of statements
	updation;
	}

2. when we will use for loop when we already knows no of Iterations Ex:- prime numbers	2. we will go for while loop when we dont know no of iterations Ex:- armstrong number.

3. It we doat give condition in for loop always JVM will consider it as true and loop will run Infinite times
public class D
{
public static void main(String args[])
{
for(int i=1; ;i++)
{
System.out.println("Java");
}
}
}
	if we did not give condition in while loop then we will get Compile time Error
	int i=1;
	while( ) //illegal start of expression
	{
	System.out.println(i);
	}

* **Do while loop:-** we will use do while loop when we want to Excute the loop atleast once irrespective of Condition.

⇒ * Syntax:-

```
initialisation;
do
{
   // statements
      updation;
} while(condition);
```

* write a programs to print 1-10 using do while

```
public class DoWhile          // class DoWhile
{
  public static void main(String args[])
  {
     int i=1;
     do
     {
        System.out.println(i);
        i++;
     } while(i<=10)
  }
}
```

* Keywords :-

These are the reserve words or predefine words of Java which has some reserve meaning

Various Keywords in java are :-

1. Accessible Keywords :-

public, private, protected, static, final and abstract, return, this, super.

2. Conditional keywords :-

if, else, else if, switch, break, Continue, goto, Concat and default.

true and false

3. Iterative Keywords :-

for, while, do while

4. class level :-

class, package, import, extends, implements, interface, new

5. Exception level :-

try, catch, throw, finally

6. Datatypes level :-

int, short, byte, long, double, float, char and Boolean

7. others :-

Volatile, transient, Synchronized, native, null etc.

Note:-

1. In Java, there is no access modifier called as default meaning is there

Ex:- class Sample — here access modifier is default

② * When we didnot give public/private/protected access modifier JVM will Consider it as default

③ All Keywords must starts with Smaller case

Combinational operators :-

It is a Combination of aithemetic and assignment operator

Ex:- a = a+b can be written as a += b

Ex:- a = a-b can be written as a -= b

Ex:- a = a*b can be written as a *= b

Ex:- a = a/b can be written as a /= b

Ex:- a = a%b can be written as a %= b

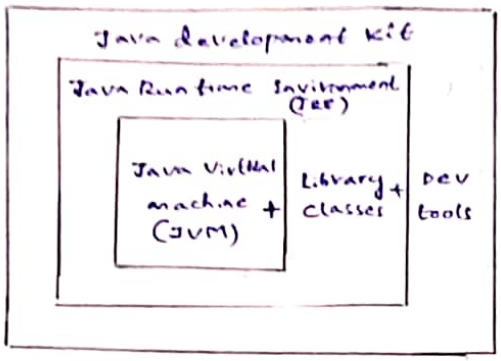
Ex:- Count = Count+1 ⟶ Count += 1

```
class D
{
  public static void main(String args[])
  {
    int i=1, j=3
    // i = i+j
      i += j
      System.out.println(i);
  }
}
```

output:-
4

JAVA DEVELOPMENT KIT (JDK)

If we want to develop and execute java program in our system we have to Install JDK



*Components of JDK

JDK = JRE + other developing tools (Compiler, debugger etc)

JRE : Java Runtime Environment

It is a part of JDK which is used for executing Java programs or Running Java app"

JRE = JVM + other predefine classes

JVM : Java Virtual machine :-

It is Responsible to execute every program

***Note :-** When we install JDK with that JVM and JRE is available.

* For Recording class in laptop or pc
  - obs studio
  - ocam
  - Apowersoft

* Switch Case Statement :-

* we will go for switch case statement when we want to test multiple conditions

* Syntax:-

```
Switch(expression/variable)
{
    case value1 : // statement
                  break;
    Case value2 : // statement
                  break;
                  :
    Case valuen : // statement
                  break;
    default : // statement
              break;
}
```

* Break:- It is a keyword which makes JVM to come out of switch body.

* Program:-1

write a program to print dayname by using following data

daynumber	dayname
1	monday
2	Tuesday
3	wednesday
4	Thursday
5	friday
6	Saturday
7	Sunday

```
public class Day
{
  public static void main(String args[])
  {
     int day_num = 4;
     Switch(day_num)
     {
        case1 : System.out.println("monday");
                break;
```

Case: 2
Case 3
Case 4
Case
Case
Case
defa
}
}
}
* day_
① * Is it
Ans

```
        Case 2 : System.out.println("Tuesday");
                 break;
        Case 3 : System.out.println("Wednesday");
                 break;
        Case 4 : System.out.println("Thursday");
                 break;
        Case 5 : System.out.println("Friday");
                 break;
        Case 6 : System.out.println("saturday");
                 break;
        Case 7 : System.out.println("sunday");
                 break;
        default : System.out.println("Couples day");
                 break;
    }
}
}
```

output:-
Thursday

* day_num = 45
output :- Couples day

① * Is it compulsory that case value should be in sequential order

Syntax:- switch(variablename)

Ans No

```
class Day
{
    public static void main(String args[])
    {
        int day_num = 4;
        switch(int day_num=1)
        {
            case 4 : System.out.println("Thursday");
                     break;
            Case 2 : System.out.println("Tuesday");
                     break;
            Case 1 : System.out.println("Monday");
                     break;
            default: System.out.println("Couples day");
                     break
        }
    }
}
```

* output
Monday

(2)
* what if we did not give break keyword in case?

If we skip break keyword and case value is matching then JVM will print all Remaining Statements without checking there Values untill it finds break keyword

If we skip break keyword and case value is not matching then it does not matter

If we skip break keyword in default then it doesnot matter because after default JVM will come out of switch body

(3)
* Can we have duplicate case value?

No the case values must be unique, if we gave duplicate ~~JVM will come out of switch body~~ we will get Compilation Error as duplicate labels

(4)
* what are the datatypes allowed for case value?

byte, short, int and char only these data types allowed

String is allowed if we have 1.7 or >1.7 version of JDK

(5)
* what are not allowed data types)

Boolean, float, double, and long

(6)
* What is a maximum limit for case values)

2147483647

(7)
* Can we check more than one Case Values?

No because we can use only Single Variable in Switch

Develop a Calculator using switch case

Character	logic
+	perform addition
-	perform Substraction
*	perform multiplication
/	perform division
%	perform modulus

```
public class Calculator
{
  public static void main(String args[])
  {
    char ch='*'
    int a=10,b=2, c;
    switch(ch)
    {
      Case '+' : c = a+b;
                 System.out.println("Addition Result is :" +c);
                 break;
      Case '-' : c = a - b;
                 System.out.println("Substraction Result is :" +c);
                 break;
      Case '*' : c = a * b
                 System.out.println("Multiplication result is :" +c);
                 break;
      Case '/' : c = a/b
                 System.out.println("Division Result is :" +c);
                 break;
      Case '%' : c = a % b
                 System.out.println("Modulus Result is :" +c);
                 break;
      default : System.out.println("Invalid operation");
                break;
    }
  }
}
```

output:-
Multiplication Result is :20

* Pattern programming via nested for loop

① * * * *
* * * *
* * * *

Approach :- one loop is for No of Rows → outer loop
one for loop is for No of Columns → Inner loop

* Program :-1

```
Public class Pat
{
  public static void main(String args[])
  {
    for(int i=1; i<=3; i++) ---> outer loop - No of Rows
    {
      for(int j=1; j<=4; j++) ---> Inner loop - No of Columns
      {
        System.out.print("* ");
      }
      System.out.println(" ");
    }
  }
}
```

* output :-

* * * *
* * * *
* * * *

For System.out.print statement don't give space after *

* program :-2

* * * *
* * * *
* * * *
* * * *

```
public class Pat2
{
  public static void main(String args[])
  {
    for(int i=1; i<=4; i++)
    {
      for(int j=1; j<=4; j++)
      {
        System.out.print("* ");
      }
      System.out.println(" ");
    }
  }
}
```

* output :-

* * * *
* * * *
* * * *
* * * *

*Iteration:-

first i=1 check Condition i<=4 true

j=1 j=2 j=3 j=4

-for one Iteration of outerloop there are multiple Iterations of Inner loop

second i=2 check Condition i<=4 true

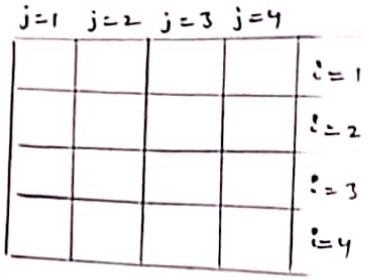
j=1 j=2 j=3 j=4

Third i=3 check Condition i<=4 true

j=1 j=2 j=3 j=4

fourth i=4 check Condition i<=4 true

j=1 j=2 j=3 j=4



*Program: 3

```
* * * *
# # # #
* * * *
# # # #
```

```
public class Part3
{
  pubic static void main(String args[])
  {
    for(int i=1; i<=4; i++)
    {
      for(int j=1; j<=4; j++)
      {
        if(i%2==0)
        {
          System.out.print("#");
        }
        else
        {
          System.out.print("*");
        }
      }
        System.out.println(" ");
    }
  }
}
```

*output:-

```
* * * *
# # # #
* * * *
# # # #
```

Program:-4

```
* # * #
* # * #
* # * #
* # * #
```

```
public class Pat4
{
  public static void main(String args[])
  {
    for(int i=1; i<=4; i++)
    {
      for(int j=1; j<=4; j++)
      {
        if(j%2==0)
        {
          System.out.print("#");
        }
        else
        {
          System.out.print("*");
        }
      }
        System.out.println(" ");
    }
  }
}
```

output:-

```
* # * #
* # * #
* # * #
* # * #
```

Program:-5

```
* * * *
* * * *
* * # *
* * * *
```

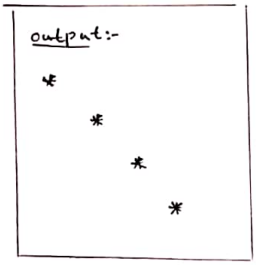
```
public class Pat5
{
  public static void main(String args[])
  {
    for(int i=1; i<=4; i++)
    {
      for(int j=1; j<=4; j++)
      {
        if(i==3 && j==3)
        {
          System.out.print("#");
        }
        else
        {
          System.out.print("*");
        }
      }
        System.out.println(" ");
    }
  }
}
```

output:-

```
* * * *
* * * *
* * # *
* * * *
```

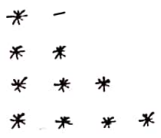
Program:-6

output:-



```
public class Pat6
{
  public static void main(String args[])
  {
    for(int i=1;i<=4;i++)
    {
      for(int j=1; j<=4 ; j++)
      {
        if(i==j)
        {
          System.out.print("*");
        }
        else
        {
          System.out.print(" ");
        }
      }
        System.out.println(" ");
    }
  }
}
```

Program:-7



Sol

	j=1	j=2	j=3	j=4	
i=1	*				
i=2	*	*			output
i=3	*	*	*		
i=4	*	*	*		

i	j	
1	1	
2	1, 2	2>1
3	1, 2, 3	3>1, 3>2
4	1, 2, 3, 4	4>1, 4>2, 4>3

Check Conditions

(i) i==j (iii) i<j (v) i-j

(ii) i>j (iv) i+j

i==j + i>j ⟶ i>=j

```
public class Pat7
{
  public static void main(String args[])
  {
    for(int i=1; i<=4; i++)
    {
      for(int j=1; j<=4; j++)
      {
        if(i>=j)
```

```
        {
            System.out.print("* ");
        }
        else
        {
            System.out.print("  ");
        }
      }
        System.out.println(" ");
    }
  }
}
```

here no space after *

program:-8:-

```
* * * *
* * *
* *
*
```

	j=1	j=2	j=3	j=4	
i=1	*	*	*	*	
i=2	*	*	*		
i=3	*	*			
i=4	*				

i	j
1	1, 2, 3, 4
2	1, 2, 3
3	1, 2
4	1

Check Condition

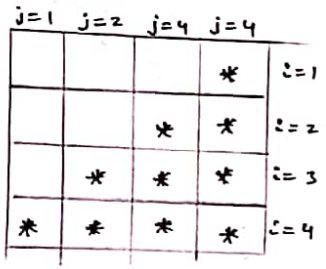
i == j	i < j	⇒ i+j <= 5
i > j	i + j	
	i - j	

```
public class Pat8
{
  public static void main(String args[])
  {
    for(int i=1; i<=4; i++)
    {
      for(int j=1; j<=4; j++)
      {
        if(i+j<=5)
        {
          System.out.print("* ");
        }
        else
        {
          System.out.print("  ");
        }
      }
    }
  }
}
```

output:-

```
* * * *
* * *
* *
*
```

program:-9



i	j	Coaditions
1	4	i == j
2	4, 3	i > j
3	4, 3, 2	i < j
4	4, 3, 2, 1	i + j
		i - j

i + j >= 5

```
public class Pat9
{
  public static void main(String args[])
  {
    for(int i=1; i<=4; i++)
    {
      for(int j=1; j<=4; j++)
      {
        if(i+j >= 5)
        {
          System.out.print("* ");
        }
        else
        {
          System.out.print("  ");
        }
      }
      System.out.println(" ");
    }
  }
}
```

*output:-

```
   *
  * *
 * * *
* * * *
```

program:-10

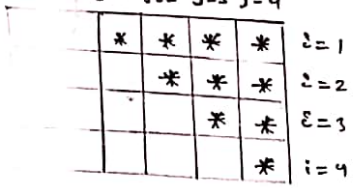
```
    *
   * *
  * * *
 * * * *
```

Note:- After "*" we need to give "* " (space)

```
public class Pat10
{
  public static void main(String args[])
  {
    for(int i=1; i<=4; i++)
    {
      for(int j=1; j<=4; j++)
      {
        if(i+j >=5)
        {
          System.out.print("* ");
        }
        else
        {
          System.out.print("  ");
        }
      }
      System.out.println(" ");
    }
  }
```

space

Program :-11



i	j
1	4, 3, 2, 1
2	4, 3, 2
3	3, 4
4	4

2>1, 3>1, 4>1
3>2 4>2
4>3

* i <= j

```
Public class Pat11
{
  public static void main(String args[])
  {
    for(int i=1; i<=4; i++)
    {
      for(int j=1; j<=4; j++)
      {
        if(i<=j)
        {
          System.out.print("*");
        }
        else
        {
          System.out.print(" ");
        }
      }
      System.out.println(" ");
    }
  }
}
```

Program :-12

```
    *  *  *  *
       *  *  *
          *  *
             *
public class Pat12
{
  public static void main(String args[])
  {
    for(int i=1; i<=4; i++)
    {
      for(int j=1; j<=4; j++)
      {
        if(i<=j)
        {
          System.out.print("* ");   (space should be given)
        }
        else
        {
          System.out.print("  ");
        }
      }
      System.out.println(" ");
    }
  }
}
```

* Type - 2 :-

(13) program :- 13

```
1 1 1
2 2 2
3 3 3
```

```
public class Pat13
{
  public static void main(String args[])
  {
    for(int i=1; i<=3; i++)
    {
      for(int j=1; j<=3; j++)
      {
        System.out.print(i);
        // System.out.print(i+" ");
      }
        System.out.println(" ");
    }
  }
}
```

output :-

```
1 1 1
2 2 2
3 3 3
```

program :- 14

```
1 2 3
1 2 3
1 2 3
```

```
public class Pat14
{
  public static void main(String args[]
  {
    for(int i=1; i<=3; i++) // no of rows
    {
      for(int j=1; j<=3; j++) // no of Columns
      {
        System.out.print(j+" ");
      }
        System.out.println(" ");
    }
  }
}
```

output :-

```
1 2 3
1 2 3
1 2 3
```

Program:-15

```
1  2  3
4  5  6
7  8  9
```

```
public class Pat15
{
  public static void main(String args[])
  {
    int k=1;
    for(int i=1; i<=3; i++)
    {
      for(int j=1; j<=3; j++)
      {
        System.out.print(k+" ");
        k++;
      }
      System.out.print("  ");
    }
  }
}
```

output:-

```
1  2  3
4  5  6
7  8  9
```

Program:-16

```
1
2  3
4  5  6
7  8  9  10
```

```
public class Pat16
{
  public static void main(String args[])
  {
    int k=1;
    for(int i=1; i<=4; i++)
    {
      for(int j=1; j<=4; j++)
      {
        if(i>=j)
        {
          System.out.print(k+"  ");
          k++;
        }
      }
      System.out.println("  ");
    }
  }
}
```

output:-

```
1
2  3
4  5  6
7  8  9  10
```

Program:-17

```
1
2  2
3  3  3
4  4  4  4
```

```
public class Pat17
{
  public static void main(String args[])
  {
    for(int i=1; i<=4; i++)
    {
      for(int j=1; j<=4; j++)
      {
        if(i>=j)
        {
          System.out.print(i+"  ");
        }
      }
      System.out.println("  ");
    }
  }
}
```

output:-

```
1
2  2
3  3  3
4  4  4  4
```

Type 3

Program:-18

```
A B C
A B C
A B C
```

```
public class Pat18
{
  public static void main(String args[])
  {
    for(int i=1;i<=3;i++)
    {
      char ch = 'A';
      for(int j=1;j<=3;j++)
      {
        System.out.print(ch);  (ch+" ");   // for space
        ch++;
      }
      System.out.println(" ");
    }
  }
}
```

output:-

```
A B C
A B C
A B C
```

Program:-19

```
A A A
B B B
C C C
```

```
public class Pat19
{
  public static void main(String args[])
  {
    char ch='A';
    for(int i=1; i<=3; i++)
    {
      for(int j=1; j<=3; j++)
      {
        System.out.print(ch);  (ch+" ");
      }
      System.out.println(" ");
      ch++;
    }
  }
}
```

output:-

```
A A A
B B B
C C C
```

Program:-20

```
A B C
D E F
G H I
```

```
public class Pat20
{
  public static void main(String args[])
  {
    char ch='A';
    for(int i=1; i<=3; i++)
    {
      for(int j=1; j<=3; j++)
      {
        System.out.print(ch);  (ch+" ");
        ch++;
      }
      System.out.println(" ");
    }
  }
}
```

output:-

```
A B C
D E F
G H I
```

Program :- 21

```
A
B C
D E F
G H I J
```

```
public class Pat21
{
  public static void main(String arg[])
  {
    char ch = 'A';
    for(int i=1; i<=4; i++)
    {
      for(int j=1; j<=4; j++)
      {
        if(i>=j)
        {
          System.out.print(ch+" ");
          ch++;
        }
      }
      System.out.println(" ");
    }
  }
}
```

Program :- 22

```
A
B B
C C C
D D D D
```

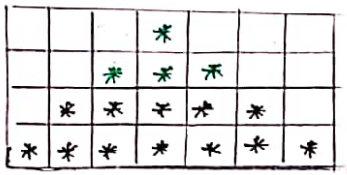
```
public class Pat22
{
  public static void main(String arg[])
  {
    Char ch = 'A'; // char ch = 'A';
    for(int i=1; i<=4; i++)
    {
      for(int j=1; j<=4; j++)
      {
        if(i>=i)
        {
          System.out.print(ch+" ");
        }
      }
      System.out.println(" ");
    }
  }
}
```

In program 22 In place of Int we can use char

```
public class Pat22
{
  public static void main(String args[])
  {
    char ch = 'A'
    for(char i = 'A'; i<='D'; i++)
    {
      for(char j = 'A'; j<='D'; j++)
      {
        if(i>=j)
        {
          System.out.print(i);
        }
      }
      System.out.println(" ");
    }
  }
}
```

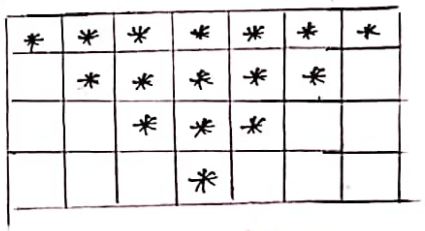
Type-4

23.



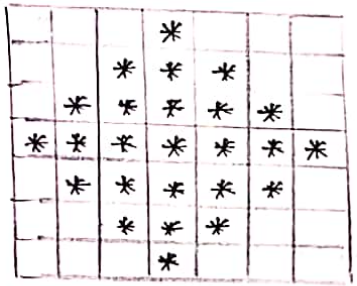
```
public class Pat23
{
  public static void main(String args[])
  {
    int star=1, space=3;// first Row no of star
                           & space before star
    //3 loops
    for(int i=1; i<=4; i++) // no of rows
    {
      for(int j=1; j<=space; j++) // no of spaces
      {
        System.out.print(" ");
      }
      for(int k=1; k<=star; k++) // no of stars
      {
        System.out.print("*");
      }
      System.out.println()
      star = star+2;
      space= space-1;
    }
  }
}
```

Program:-24



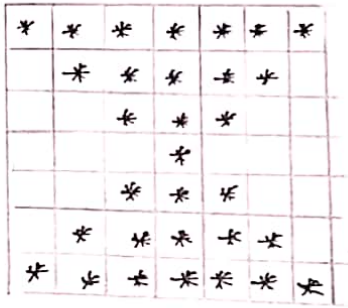
```
public class Pat24
{
  public static void main(String args[])
  {
    int star=7, space=0; // first row no of star and
                            space before star
    //3 loops
    for(int i=1; i<=4; i++) // no of rows
    {
      for(int j=1; j<=space; j++) // no of spaces
      {
        System.out.print(" ");
      }
      for(int k=1; k<=star; k++) // no of stars
      {
        System.out.print("*");
      }
      System.out.println();
        star = star-2;
        space = space+1;
    }
  }
}
```

Program:-25



```
public class Pat25
{
  public static void main(String args[])
  {
    int star=1, space = 3;
    // 3 loops
    for(int i=1; i<=7 ; i++)
    {
      for(int j=1; j<=space; j++)
      {
        System.out.print("  ");
      }
      for(int k=1; k<=star; k++)
      {
        System.out.print(" *");
      }
        System.out.println(); // conditions
      if(i<=3)
      {
        star = star+2;
        space = space-1;
      }
      else
      {
        star = star-2;
        space = space+1;
      }
    }
  }
}
```

Program:-26



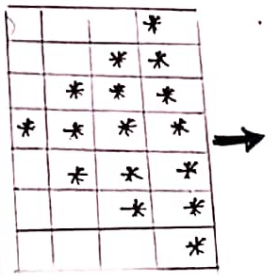
```
public class Pat26
{
  public static void main(String args[])
  {
    int star=7, space=0;
    for(int i=1; i<=7; i++)  // no of rows
    {
      for(int j=1; j<=space; j++) // no of space
      {
        System.out.print("  ");
      }
      for(int k=1; k<=star; k++) // no of spaces
      {
        System.out.print(" *");
      }
        System.out.println();
      if(i<=3)
      {
        star = star-2;
        space = space+1;
      }
      else
      {
        star = star+2;
        space = space-1;
      }
    }
  }
}
```

program :- 27 :-

```
    *
   * *
  * * *
 * * * *
  * * *
   * *
    *
```

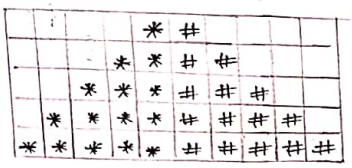
```
public class Pat27
{
  public static void main(String args[])
  {
    int star=1;
    // 3 loops
    for(int i=1; i<=7; i++)
    {
      for(int k=1; k<=star; k++)
      {
        System.out.print("*");
      }
        System.out.println();
      if(i<=3)
      {
        star = star+1;
      }
      else
      {
        star = star-1;
      }
    }
  }
}
```

program :- 28 :-



```
public class Pat28
{
  public static void main(String args[])
  {
    int star=1, space=3;
    for(int i=1; i<=7; i++)   // no of rows
    {
      for(int j=1; j<=space; j++) // no of spaces
      {
        System.out.print(" ");
      }
      for(int k=1; k<=star; k++) // no of stars
      {
        System.out.print("*");
      }
      if(i<=3)
      {
        star = star+1;
        space = space-1;
      }
      else
      {
        star = star-1;
        space = space+1;
      }
    }
  }
}
```

Program:-29



```
public class Pattern29
{
  public static void main(String args[])
  {
    int star=1, space=4, ash=1;
    // 4 loops
    for(int i=1; i<=5; i++)
    {
      for(int j=1; j<=space; j++)
      {
        System.out.print(" ");
      }
      for(int k=1; k<=star; k++)
      {
        System.out.print("*");
      }
      for(int l=1; l<=ash; l++)
      {
        System.out.print("#");
      }
      System.out.println(" ");
      star = star+1;
      space = space-1;
      ash = ash+1;
    }
  }
}
```

* Write a program to print all prime numbers 2 to 15

Sol:

```
public class Prime
{
  Public static void main(String args[])
  {
    for(int num = 2; num <= 15; num++)
    {
      int count = 0;
      for(int i=1; i<=num; i++)
      {
        if(num % i == 0)
        {
          count++;
        }
      }
        if(count == 2)
        {
          System.out.println(num+" is a prime number);
        }
    }
  }
}
```

output:-

```
2 is a prime number
3 is a prime number
5 is a prime number
7 is a prime number
11 is a prime number
13 is a prime number
```

* Write a program to print sum of prime numbers from 2 to 15

```
public class PrimeSum
{
  public static void main(String args[])
  {
    int sum = 0; // int sum=0;
    for(int num=2; num<=15; num++)
    {
      int count=0;
      for(int i=1; i<=num; i++)
      {
        if(num%i==0)
        {
          count++
        }
      }
      if(count==2)
      {
        sum = sum+num;
      }
    }
    System.out.println("Sum of prime numbers from 2 to 15 is:" + sum);
  }
}
```

output:-

Sum of prime number from 2 to 15 is:41

41

Write a program to print Sum of prime numbers from -11 to -32

```
public class PrimeNeg
{
  public static void main(String args[])
  {
    int Sum=0;
    for(int num=-11; num>=-32; num--)
    {
      int Count=0;
      for(int i=-1; i>=num; i--)
      {
        if(num%i==0)
        {
          Count++
        }
      }
      if(Count==2)
      {
        Sum=Sum+num;
      }
    }
    System.out.println("Sum of prime no from -11 to -32 is:"+Sum);
  }
}
```

* output:-
Sum of prime no from -11 to -32 is : -143

Write a program to print all prime numbers from -22 to -65

```
public class PrimeNeg
{
  public static void main(String args[])
  {
    for(int num=-22; num>=-65; num--)
    {
      int Count=0;
      for(int i=-1; i>=num; i--)
      {
        if(num%i==0)
        {
          Count++
        }
      }
      if(Count==2)
      {
        System.out.println(num+" is a prime number");
      }
    }
  }
}
```

output:-
-23 is a prime number
-29 is a prime number
-31 is a prime number
-37 is a prime number
-41 is a prime number
-43 is a prime number
-47 is a prime number
-53 is a prime number
-59 is a prime number
-61 is a prime number

write a program to Count prime numbers from 2 to 15

```
public class PrimeCount
{
public static void main(String args[])
{
int Count_pn = 0;
for(int num = 2; num <= 15; num++)
{
int count = 0;
for(int i = 1; i <= num; i++)
{
if(num % i == 0)
{
Count++
}
}
if(count == 2)
{
System.out.println(num + " is a prime number");   wrong
Count_pn++
}
}
System.out.println("Total no of prime numbers from 2 to 15
is :" + count_pn);
}
}
```

* output:-

```
Total no of prime number from 2 to 15
is : 6
```

* features of Java :-

1. Simple : Easy to understand syntax
2. object oriented programming
3. portable : Adjustable with anyother language
4. platform independent

* Is JVM is platform independent?

No it is platform dependent

Q. Is Java an operating System language / platform dependent lang?

Sol No it is platform independent language