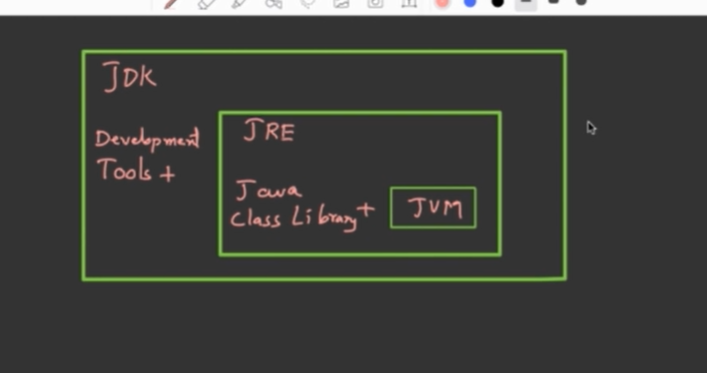
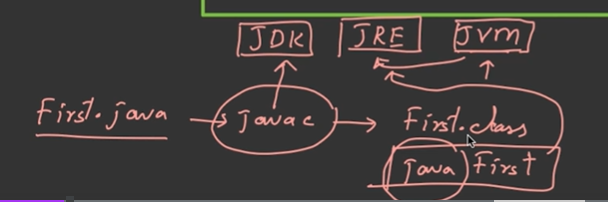
Java

Introduction:



JDK is a whole development kit for developing java application .

JDK is development kit which include JRE inside it JRE is run time environment program execution begins with JRE ,JRE consist of various java class Library + JVM,JVM converts the .class file or byte code into Machine code

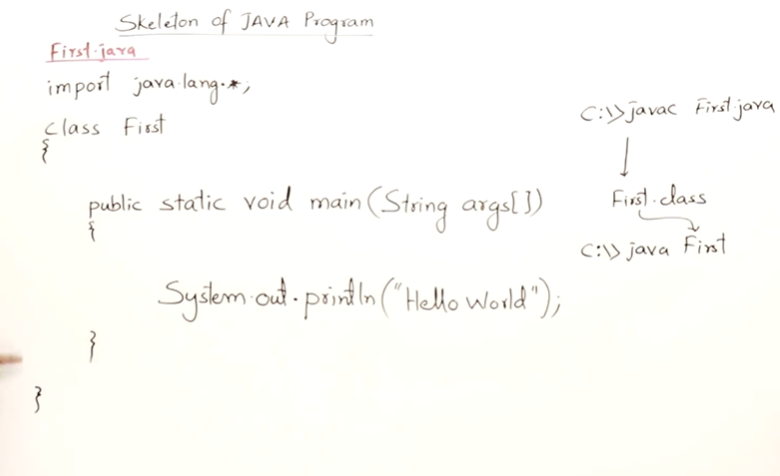


Javac present inside JDK 🡪javac First.java compile by JDK and First.class file generate and🡪 java First executed by JRE which contains JVM inside it

C:>path and press Enter(path is global variable of window )

C:>set path=c:/programFiles/java/jdk/bin;%path%; 🡪%path% to save previous path or backup of previous set path

Or🡪 simply go to Environment Variable and set System path globally so it can accessible by any user



Class name is First and its not compulsory so that the **file name and class name Should be same** but there is also one condition where the class Name and file Name should be same

Class Second 🡪if you save it file Name as 🡪First.java 🡪it will compile successfully and not giving any error but if you execute 🡪java First and it will

C:/Java>dir 🡪you get First.java and Second.class

throw Error that main method should not found so for Executing Program 🡪java Second

**Condition where the File name And Class Name should be same**

public class First

{

public static void main(String args[])

{

}

}

File Name save as:First.java then no issue but if you write

public class Second

{

}

Then it give Error class second is public it should be declared inside file name Second.java

**Import java.lang.\*;**

Is the package for System class if you not mentioned or import it it will automatically get imported

**In java inBuilt classes name Start with Capital Letter**

Ex. System.out.println() 🡪Here System is a class and out is object and println is a method of class System

Ex. String args[] 🡪String is a inbuilt class inside java

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

* main method should be **public** because public method can accessible by outside the class so when you run java Program

if it is not public then it cant accessible by Java Interpreter JVM

* **static** (as we know in java every things is about class and object so main method is static means we don’t need any object to call the method its is directly executed with class name c:>**java First** then it internally called 🡪 **java First.main()**
* **void** main method doesn’t return anything
* **main()** it’s a entry point of compiler program execution begins with main() method
* **String args[]** it is the command line arguments

**{**

**System.out.println(args[0]); //** it store 0 and 1 argument in args and print

**System.out.println(args[1]);**

**}**

**If you write like this 🡪public static void main()**

Program will compile successfully but at run time it give error because Main method should be only in this Format 🡪**public static void main(String args[])**

**Scanner class**

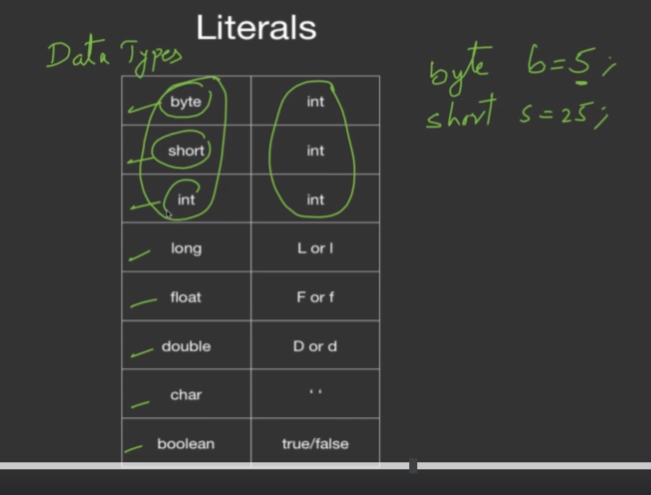
**Refer :** [**https://www.programiz.com/java-programming/scanner**](https://www.programiz.com/java-programming/scanner)

**javap tool is used to get the information of any class or interface**

**C:\Users\Admin>javap java.util.Scanner (this command is used to check the method inside that class**

**Literals :**

**Literal:** Any constant value which can be assigned to the variable is called literal/constant.



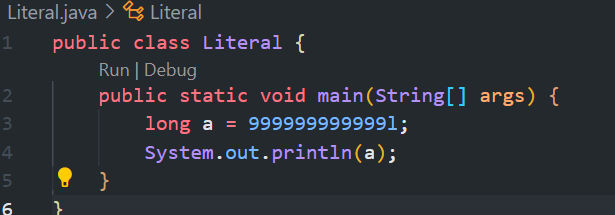
We can assign literal to only its compatable data types

Eg int i=124L

It give error incompatbale data type you cannot assign long literal to int

Eg long a=234 its is true it a interger type literal but if you exceed size

Eg long a=999999999999 its give error so fix by long a=999999999L

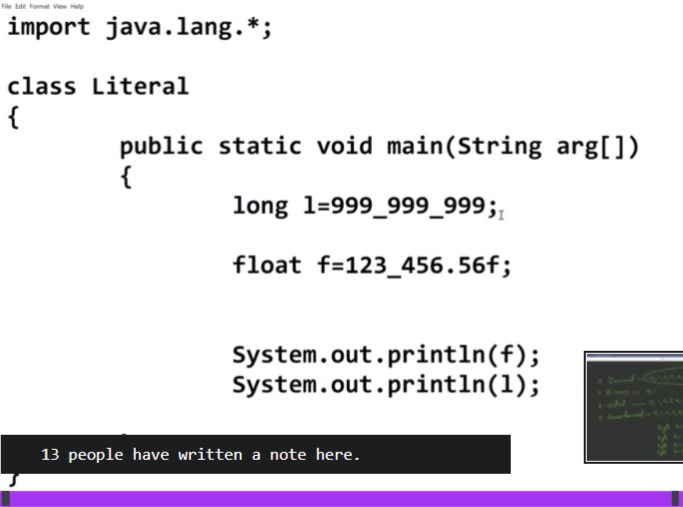


If you remove l you get error

float a=12.4

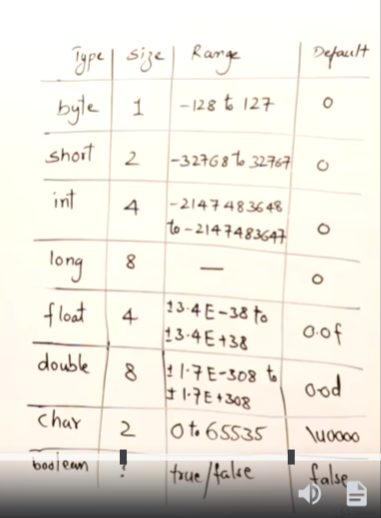
system.out.println(a) error so fix by adding float a=12.4f

By default any number in decimal form is double



\_ is allowed make readability of number

DataTypes:



**Integral datatype :**

When java was introduced 32 bit machine where is in market so therefore integer can take 4 bytes (1 bytes=8 bits ,8\*4=32bit) at a single cycle or one snap shot it can process 32 bit

If you give larger value then it can process in two step so therefore long datatypes are available therefore long is compatibly slow than int depending on hardware

At the time of c and c++ 32 bit and 16 bit processor are available so depending on architecture it can take 2 or 4 byte for integer

In java ,byte and short dataTypes are also available which take 1 and 2 byte so we are underutilizing the processor not fully utilize the processor then **why java has support for byte and short?**

When we declared variable its occupy some space in memory so byte has occupy less space in memory its save space of memory

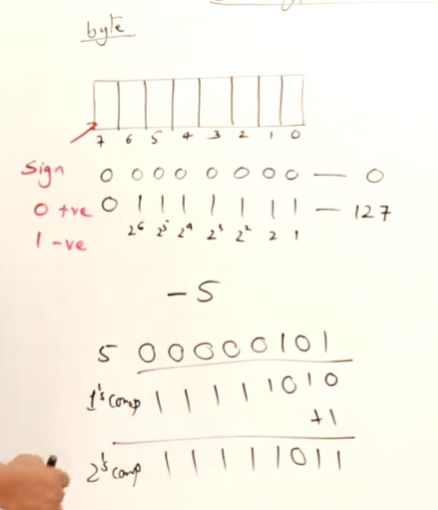
When you declared smaller number range values then instead of using int use byte or short so it can save space this is **the one reason why java has support for byte and short**

when java was introduced older language like c,c++,cobal are also in use so to provide the compatibility support for those language or communicating and integration with those language this **is the another reason why java has support for byte and short**

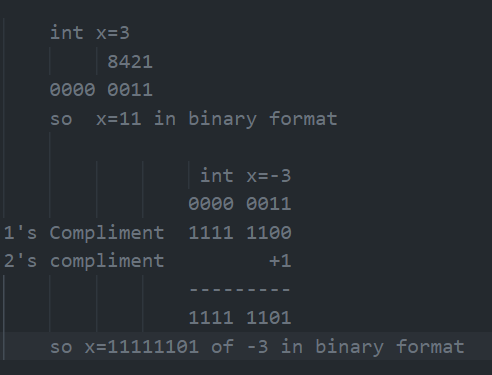
**Data type stores both Positive and Negative Number:**

Then how negative number stored this is the reason due to of sign bit

**0🡪positive 1🡪Negative (finds 2s compliment)**

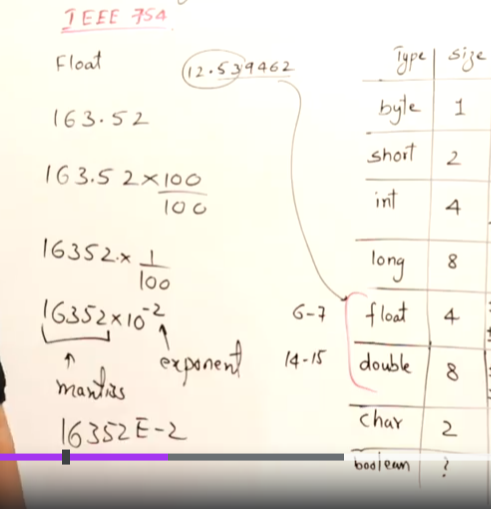
****

**How negative no stored in memory 🡪**

****

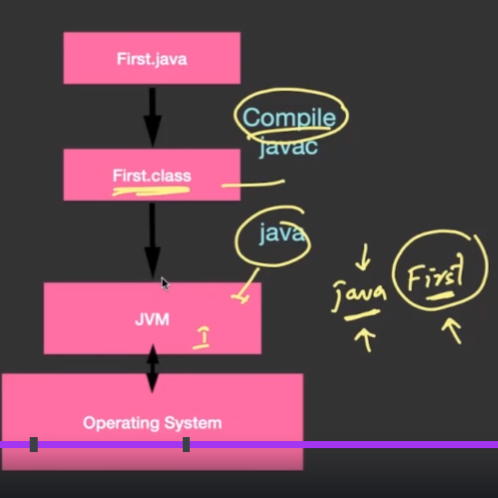
**How Floating value stored in Memory 163.52 its don’t store . in memory**

**It stores like 16352 \*10^-2**

****

**Floating value stores upto 6 to 7 digit after the decimal if you want more precise value then use double 14 to 15 digit after decimal**

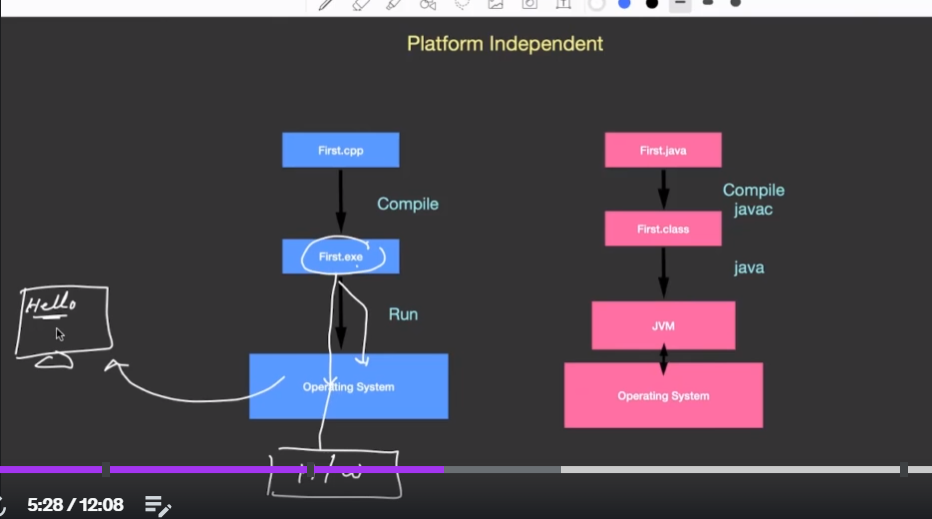
**Java take 2 byte for char to support the Unicode for Unicode 7 bits is not sufficient other programming language like c,c++ has support for only ASCII(American standard code for information interchange) that’s c and c++ has 1 byte for char**

****

**(Java is both compiled and interpreter language ) javac is a compiler which converts the source code into byte code and this is done by compiler javac and then java interpreter execute byte code line by line into machine code**

**Compiler is faster than Interpreter …interpreter has read line and execute it again it read and execute so interpreter is also called and execution of program both simultaneously running where as in compiler at a stretch it compile all high lvl code into machine code once so once you compile you can as many times you want so only execution happen therefore compiler is fast**

**Platform Independent**

****

In other programming language such as c ,c++ when we execute the code it is directly make (a single binary executable file).exe file i.e machine executable code or In linux .out file which will interact with os by system call and produce output on hardware

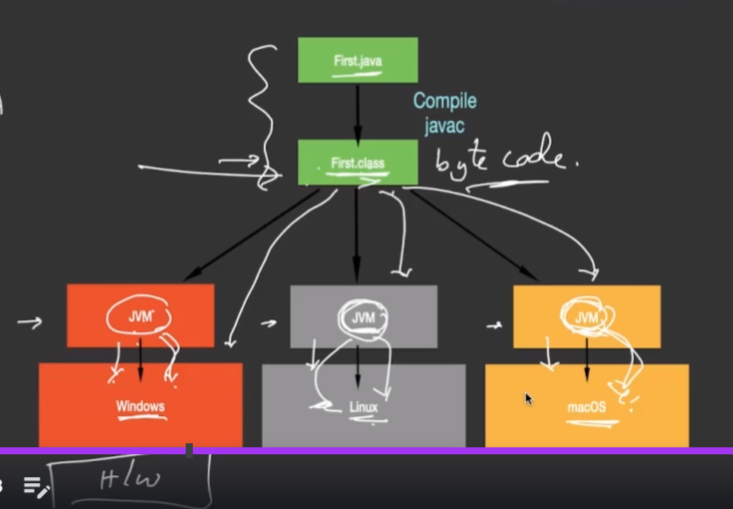
Any code directly doesn’t produce output on hardware first it interact with os and then hardware **code🡪os🡪hardware**

In java javac compiler creates a .class files that is byte code and byte code is not a machine code it is error free code again java interpreter line by line execute the byte code into machine level code with the help of JIT and Java interpreter

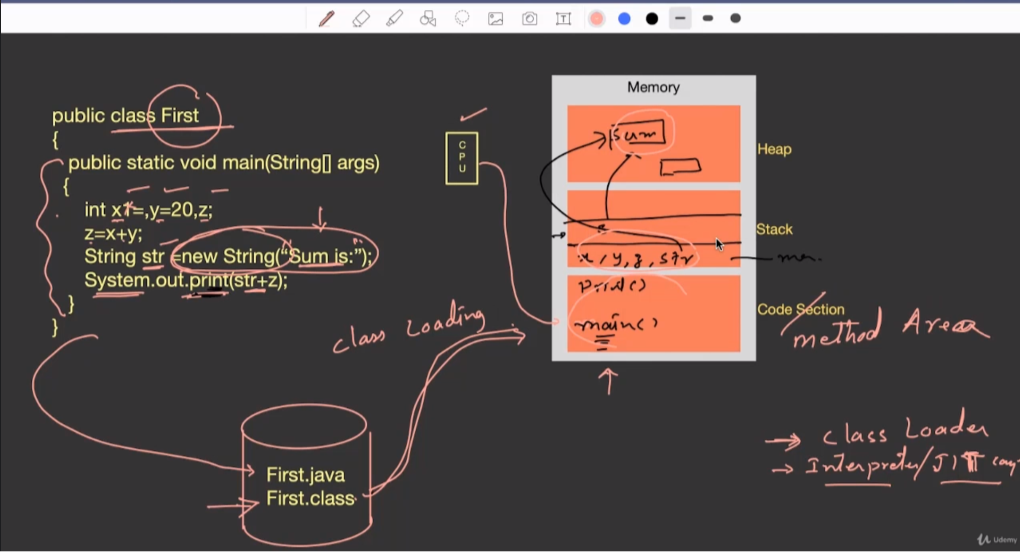
.class file can run on any machine except it has a corresponding jvm install on that machine jvm is a platform dependent

Wora 🡪write once run anywhere or compile once and run anywhere

JVM is the key component that’s makes the java platform independent

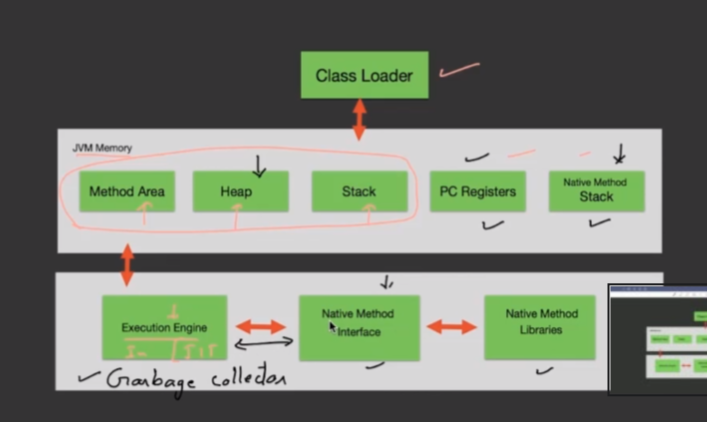


Architecture:



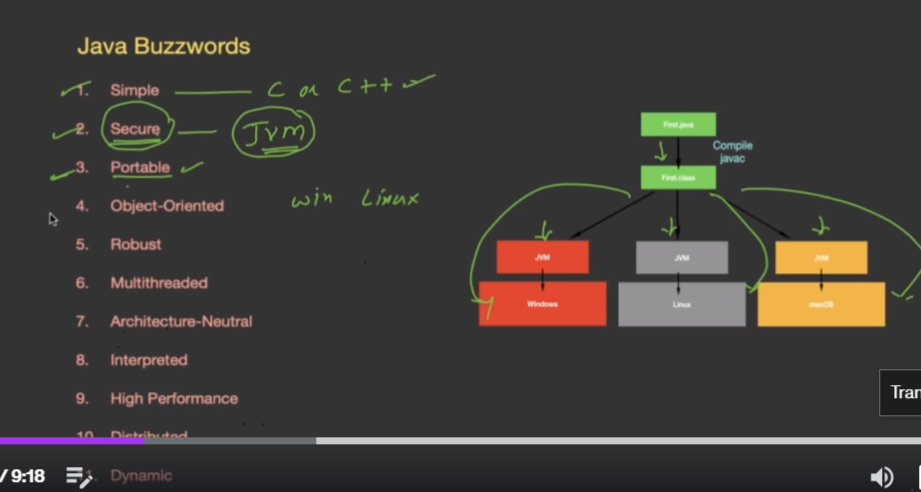
Local variable are store inside the stack and variable that created during run time or execution of program is stored in heap i.e object is stored in heap

new String(“Sum is:”) new types variable stored in objects

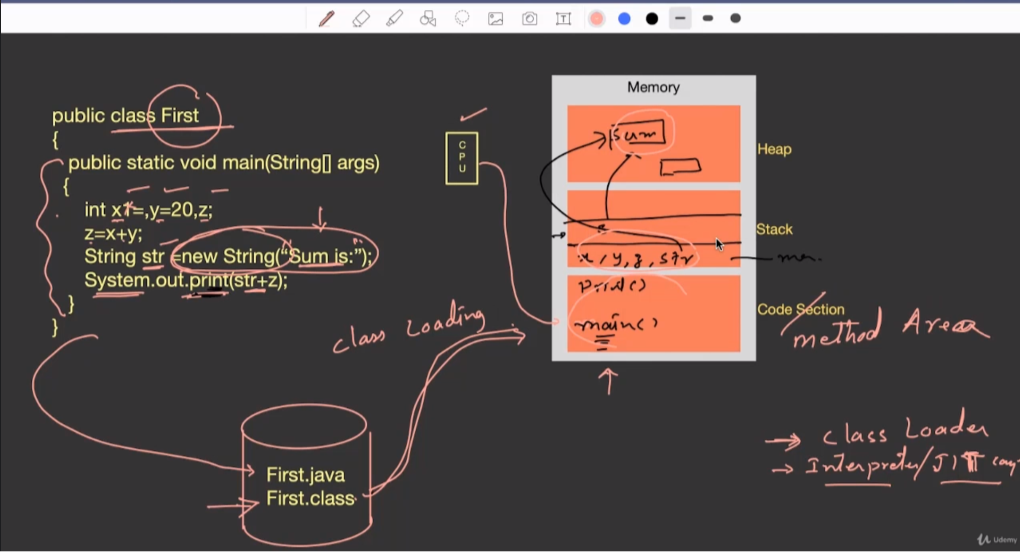


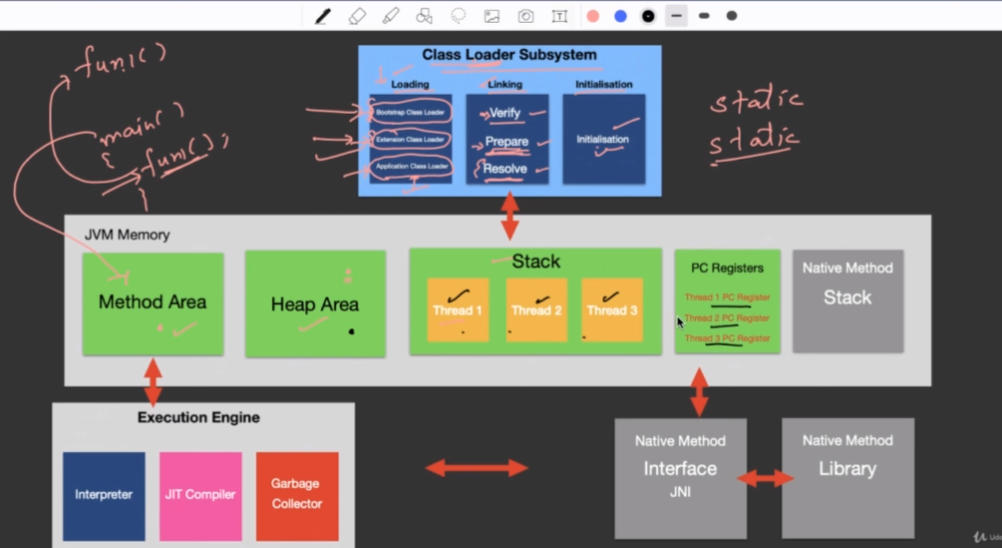
Garbage collector removed the unused memory when objects are created in heap and now objects are not in used so garbage collector removed that objects from the heap

**Features:**

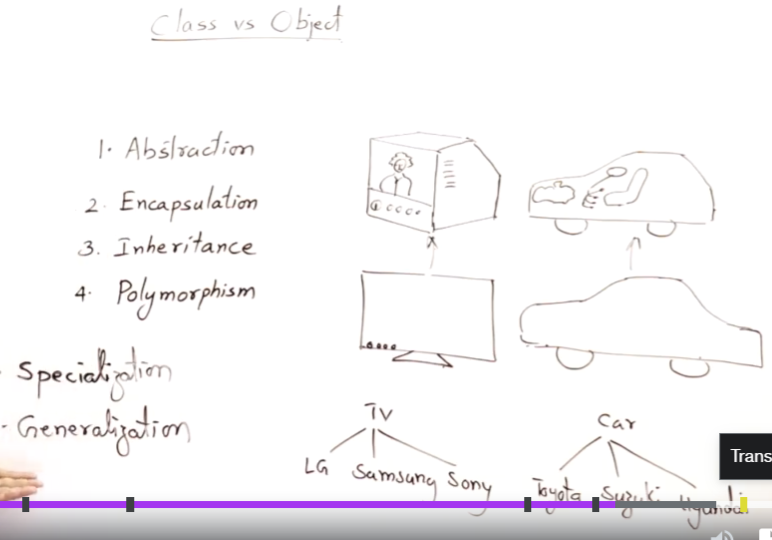


Notes:  
Static variable are not stored in stack or heap they are stored in method area





OOPS



Specialization :refers to inheritance where something are already there we just add some new features and launch it as new products

Generalization means refers set of things togeather under one names it helps to handling many things just by a single refrence

(tv🡪we just refer term television we just learn to operate television and by this we can able to operate any brand television this is generalization on general term i.e TV same for car we just learn car driving by this we can drive any car brand we don’t learn separate driving of honda ,tata car etc

Generalization refers to polymorphism ::one things behave in multiple forms

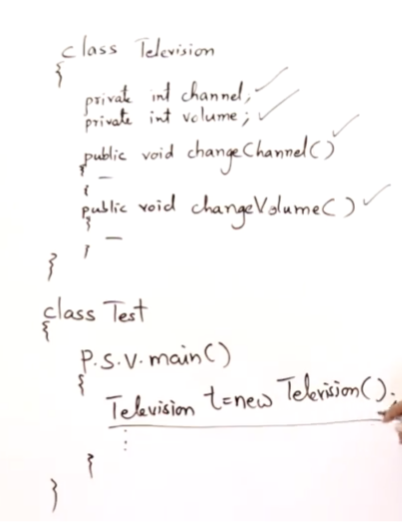
Its has ability of display one message in many forms eg a Man 🡪 he can be a father ,husband ,employee a man is generalization terms a man can acts as many roles

Abstraction:hiding the internal details and showing only the required features

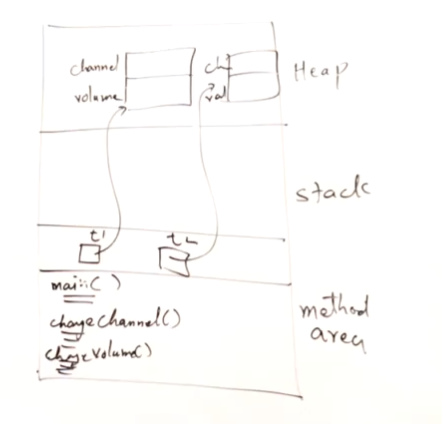
**Object:** are deined in the terms of two 1) **is Properties and 2) Behaviour**

Any things in the world has its properties and behaviour

We make the object eg abc a=new abc() 🡪a is the properties of abc and we call some methods like a.display() 🡪that print something on screen that’s it bhevaiour(its functions )



Here is also class has its properties and behaviour variables/data members are the properties and methods are the behaviour



Variables inside the main methods are local variable and stored in stack

Eg Television t1=new Television()

Variable t is stored in stack but objects are always created in heap it take reference to that objects seen in diagram

If Television t2=new Television() t2 is stored in stack but object of t2 is created in heap and t2 is referred to that object

Note :For every class java will create a seperate class files

Eg class abc{

}

Class prq{

}

If you compile and see two class files will be generated.

**Data Hiding:**

Data are hidden and operations are visible to users eg in television circuitory Is hidden like inside tv wiring all circuit are hide and operation like buttons are visible to user to by the operation they can operate the data .(operation or methods is perform on data only)

private int length;

private int breadth;

by using private access modifier we cant now access the data outside the class .we make the data is hidden

eg

class abc{

private int length;

private int breadth

public int area()

{

return length\*bredth;

}

}

Class test{

p.s.v.m(String args[])

{

abc a =new abc()

a.length=10;//it is not allowed now because length is private

}

}

Data hiding make the data more organize and classes main organize

To Access and organize data methods we can write getter and setter methods

We can,t access the property directly of private we can access it by using property methods i.e getter and setter methods



Eg:

class reactangle {

    private int length;

    private int breadth;

    public int area() {

        return length \* breadth;

    }

    public int perimeter() {

        return 2 \* area();

    }

    public int getLength() {

        return length;

    }

    public int setLength(int length) {

        return this.length = length;

    }

    public int getBreadth() {

        return breadth;

    }

    public int setBreadth(int breadth) {

        return this.breadth = breadth;

    }

    public boolean isSquare() {

        if (length == breadth) {

            return true;

        } else {

            return false;

        }

    }

}

public class demo {

    public static void main(String[] args) {

        reactangle r = new reactangle();

        r.setLength(10);

        r.setBreadth(9);

        System.out.println("Area is:" + r.area());

        System.out.println("perimeter is:" + r.perimeter());

        System.out.println("Square is:" + r.isSquare());

    }

}

Constructor:

Constructor is the method of class which is automatically called whenever the object of class is created

Compiler always provide a one constructor so if you don’t even make your own constructor so default constructor is called

If you want to defined your own constructor then its should have same name of class name and it should not have any return type not even void

Constructor usually declared with public you can also declared as private also

Two types of constructor is 1.Non parametrize(it is the replacement of default constructor) and 2.parametrize

**Operator:**

**Coercion**: Internally compiler convert the byte and short datatype to int because byte and short are integer type

Byte+short=int

Byte+byte=int

Short+byte=int

Int+float=float (first it convert the int to float and then add and result will produce as a float)

Char+int=int

Char+sort=int

Eg

Byte z=2

Short x=4

Shot c=z+x //give error

So is should be like int c=x+z

eg

  char a = 'a';

        int b = 1;

        int c = a + b;

        System.out.println(c);

Here In above eg Ascill value of a is 97 so int c=a+b internally convert the char a to its corresponding integer value and then add it so you get the result as 98

Char c=a+b; not allowed because you cant convert the char to int

Eg

 int a, b, c;

        float s;

        double area;

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter the 3 side of triangle:");

        a = sc.nextInt();

        b = sc.nextInt();

        c = sc.nextInt();

        s = (a + b + c) / 2f;

        System.out.println("Semi perimeter:" + s);

        area = Math.sqrt(s \* (s - a) \* (s - b) \* (s - c));

        System.out.println("Area:" + area);

        sc.close();

eg:

import java.util.\*;

public class qudratic {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        /\*

         \*

         \* r1=(-b+Math.sqrt(b\*b-4\*a\*c))/2\*a

         \*/

        int a, b, c;

        System.out.println("Enter the value of a b and c:");

        a = sc.nextInt();

        b = sc.nextInt();

        c = sc.nextInt();

        double r1, r2;

        r1 = ((-b + Math.sqrt(b \* b - 4 \* a \* c)) / 2 \* a);

        r2 = ((-b - Math.sqrt(b \* b - 4 \* a \* c)) / 2 \* a);

        System.out.println("Roots of r1:" + r1);

        System.out.println("Roots of r2:" + r2);

        sc.close();

    }

}

Enter the value of a b and c:

1

5

6

Roots of r1:-2.0

Roots of r2:-3.0

Note:Only integer type is applied on bitwise operator not float and doubled



**Widening**: also known as upcasting

Storing the data type of smaller size data type into larger data type can be done **Widening** automatically (implicitly)by a compiler this process is known as widening

Source 🡪 destination

Byte,short 🡪int

**Narrowing:** also known as downcasting

Here we store higher data type to lower data type .in narrowing there is a chance of loose of data

Eg

Byte b=10;

Short s=100;

b=s(byte mai daalo short ka value) this is not allowed but we can use narrowing concept so

b=byte(s) //but it has a possibility of losing the data

eg

byte b=10;

short s=200;

b=byte(s)// but we know the size limit of byte is 127 so if we try to exceed the value so we can lossy of data so compiler will give you any lossy value

eg

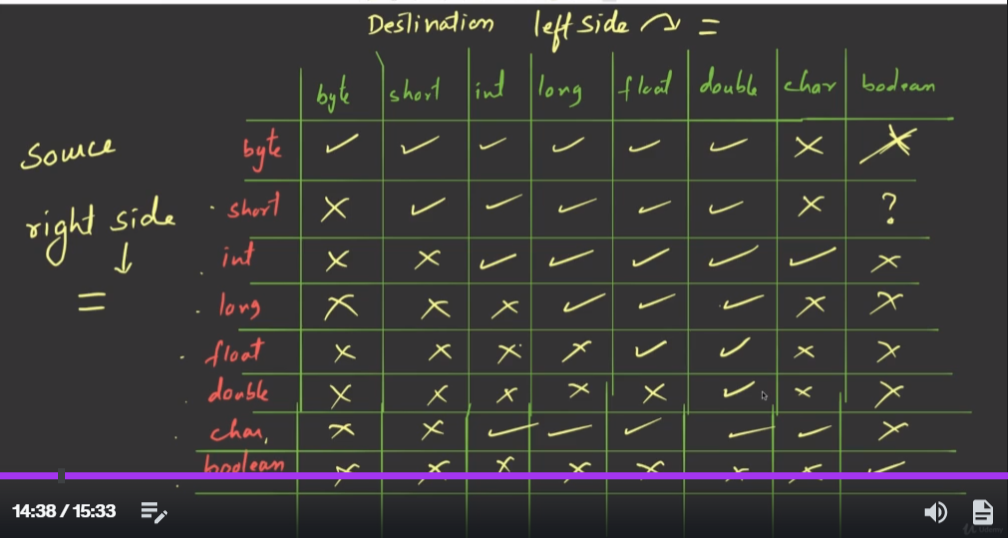
int i=10;

float f=10.5

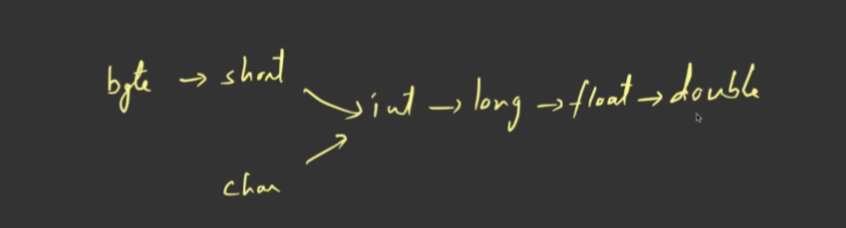
i=f;//not allowed so we do narrowing i.e downcasting

i=(int)f;

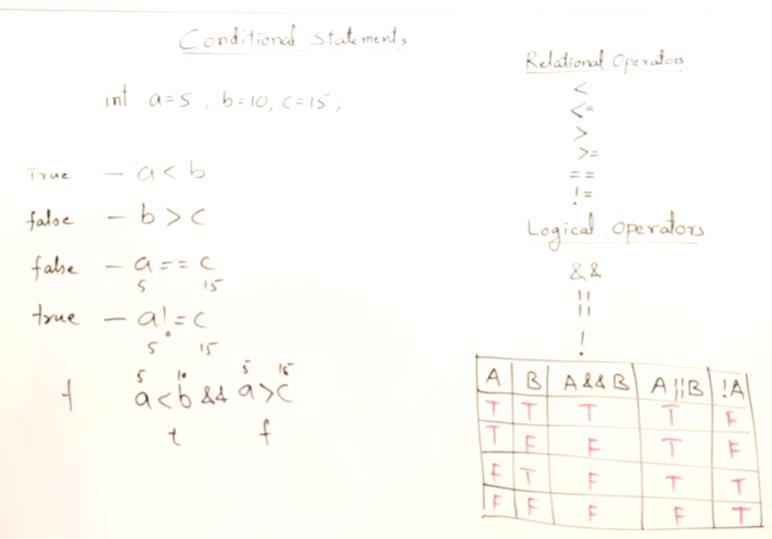
s.o.p(i) data will lose you get 10 not 10.5

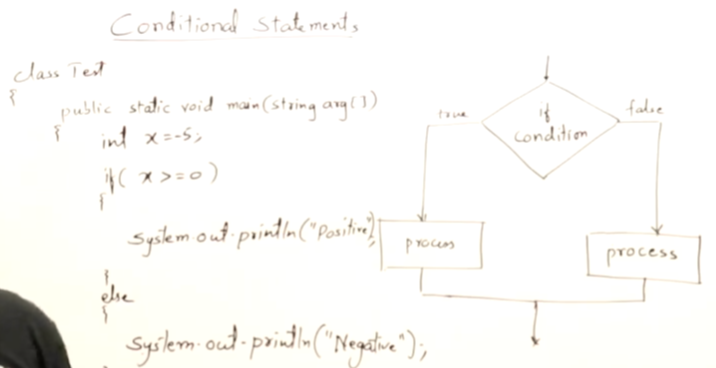
****

**Compatability:**

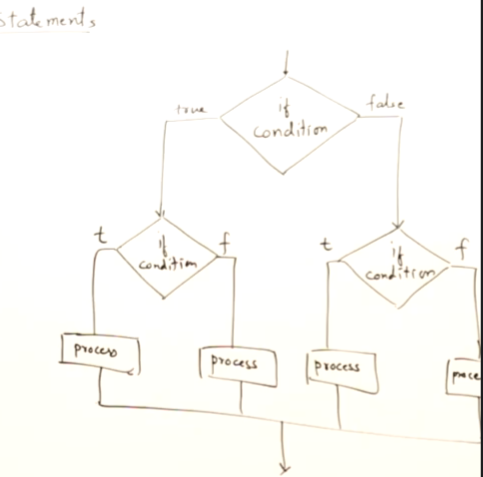
****

**Conditional Statement:**

****

****

**Nested If:**

****

**Eg**

import java.util.\*;

public class oddeven {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int n;

        System.out.println("Enter the number:");

        n = sc.nextInt();

        if (n > 0) {

            if (n % 2 == 0) {

                System.out.println(n + ": is Even");

            } else {

                System.out.println(n + ": is odd");

            }

        } else {

            System.out.println("Invalid inputs");

        }

        sc.close();

    }

}

import java.util.\*;

public class radix {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter the Number:");

        String str = sc.nextLine();

        if (str.matches("[01]+")) {

            System.out.println("Binary radix=2");

        } else if (str.matches("[0-7]+")) {

            System.out.println("octal radix=8");

        } else if (str.matches("[0-9]+")) {

            System.out.println("decimal radix=10");

        } else if (str.matches("[0-9A-F]+")) {

            System.out.println("Hexadecimal radix=16");

        } else {

            System.out.println("invalid input");

        }

    }

}

**Leapyear: (**agar no 4 se divisible hojata agar 4 se hota toh 100 and 400 se bhi divisible ho tab wo leapyear hai)

import java.util.\*;

public class leapYear {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter the Number:");

        int n = sc.nextInt();

        if (n % 4 == 0) {

            if (n % 100 == 0) {

                if (n % 400 == 0) {

                    System.out.println(n + ":is LeapYear");

                } else {

                    System.out.println(n + ":is not leapYear");

                }

            } else {

                System.out.println(n + ":is LeapYear");

            }

        } else {

            System.out.println(n + ": Not leapyear");

        }

        sc.close();

    }

**Loops:**

**Loops are used to do the repeatable task if there are some condition that are repeatable then we use loop to iterate over the process**

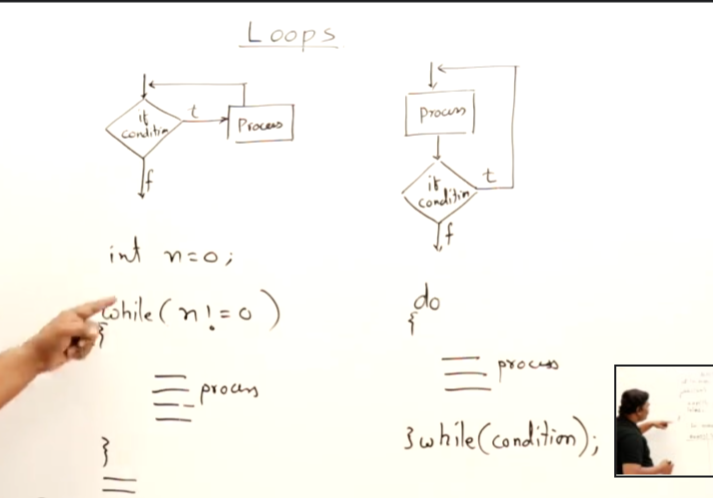
**1.While loop**

**2.For loop**

**3.do while**

**4.foreach loop**

**for loop is used when the number of iterations is known whereas, in the while loop, execution is done until the statement in the program is proved wrong**

****

import java.util.\*;

public class DigitToword {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter the number:");

        // agar kabhi bhi number ko words mai karna hai toh pahele usko string mai banao

        // phir charat se uska char extract karo

        int n, r;

        n = sc.nextInt();

        String str = "";

        while (n > 0) {

            r = n % 10;

            n = n / 10;

            str = str + r;

        }

        char c;

        // System.out.println(str);

        for (int i = str.length() - 1; i >= 0; i--) {

            c = str.charAt(i);

            switch (c) {

                case '0':

                    System.out.print("Zero ");

                    break;

                case '1':

                    System.out.print("One ");

                    break;

                case '2':

                    System.out.print("Two ");

                    break;

                case '3':

                    System.out.print("Three ");

                    break;

                case '4':

                    System.out.print("four ");

                    break;

                case '5':

                    System.out.print("five ");

                    break;

                case '6':

                    System.out.print("six ");

                    break;

                case '7':

                    System.out.print("seven ");

                    break;

                case '8':

                    System.out.print("eight ");

                    break;

                case '9':

                    System.out.print("nine ");

                    break;

                default:

                    System.out.print("Invalid");

            }

        }

        sc.close();

    }

}

**Ap Series**

import java.util.\*;

public class ApSeries {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter the Starting term:");

        int a = sc.nextInt();

        System.out.println("Enter the common difference:");

        int b = sc.nextInt();

        System.out.println("Enter the nth term:");

        int n = sc.nextInt();

        int c = a;

        for (int i = 0; i < n; i++) {

            System.out.print(c + " ");

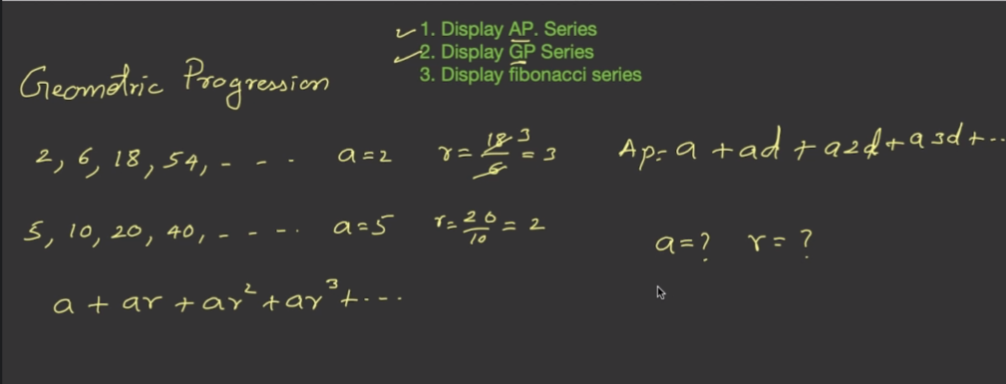
            c = c + b;

        }

        sc.close();

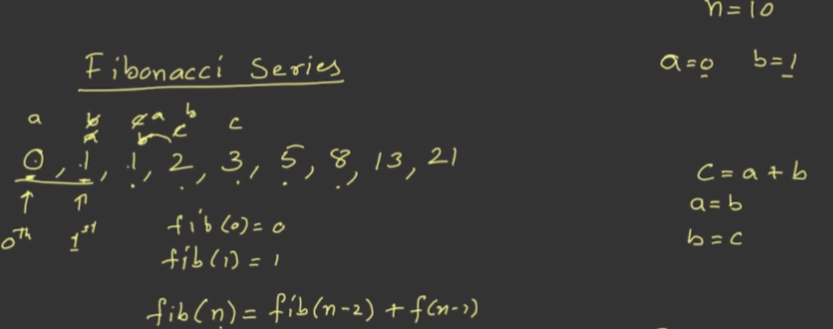
    }

}

****

**In Gp series we get it by multiplying common difference**

**Fibonacci**

****

**In fibo firs t two term is a and b so next term will c and c=a+b but for next iteration prev term should be a and b so replace value as a=b and b=c make a as b and b as c and next term c so by adding a and b we get sum of previous term**

import java.util.\*;;

public class fibonacci {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter the number:");

        int n = sc.nextInt();

        int a = 0, b = 1, c;

        System.out.print(a + " ");

        System.out.print(b + " ");

        for (int i = 0; i <= n; i++) {

            // System.out.print(b + " ");

            c = a + b;

            a = b;

            b = c;

            System.out.print(c + " ");

        }

        sc.close();

    }

}

// 0 1 1 2 3 5

**N =5 so value 5 baar he iterate hona chaiye**

**Enter the number:**

**5**

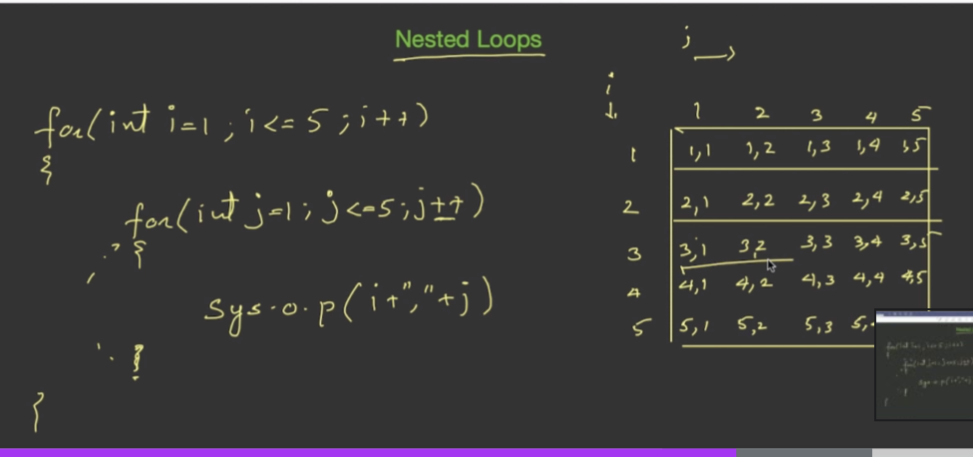
**0 1 1 2 3 5**

**Enter the number:**

**10**

**0 1 1 2 3 5 8 13 21 34 55**

**Nested Loop**

****

public class pattern {

    public static void main(String[] args) {

        int count = 0;

        for (int i = 1; i <= 5; i++) {

            for (int j = 1; j <= 5; j++) {

                count++;

                System.out.format("%02d ", count);

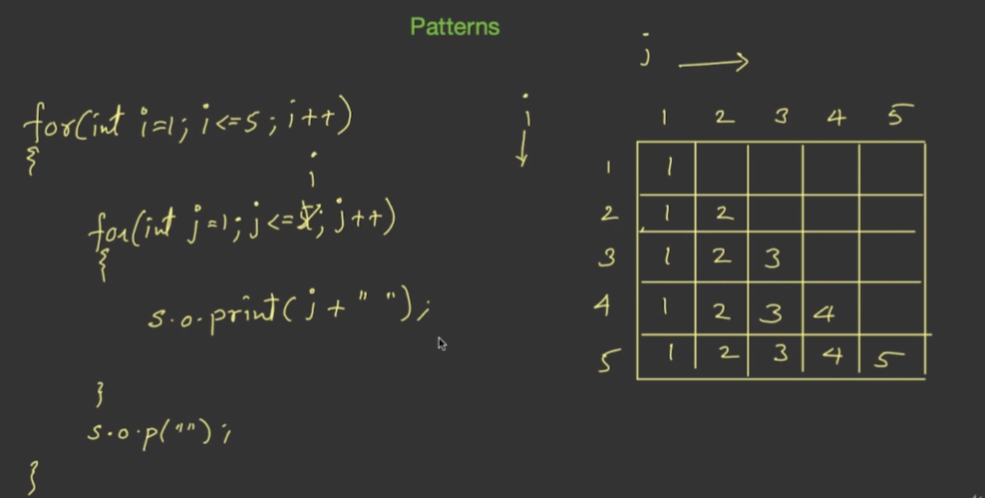
            }

            System.out.println(" ");

        }

    }

}

****

**Array:**

import java.util.\*;

public class largest {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter the size of array");

        int n = sc.nextInt();

        int arr[] = new int[n];

        // int c;

        for (int i = 0; i < n; i++) {

            System.out.println("Enter the " + i + " element ");

            arr[i] = sc.nextInt();

        }

        int max = arr[0];

        for (int i = 0; i < arr.length; i++) {

            if (arr[i] > max) {

                max = arr[i]; //

            }

        }

        // smallest number from array

        int min = arr[0];

        // 3 4 6 8

        // min=3

        // inside loop min<arr[i] 3<=3 min =arr[i] 3>=2

        System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

        for (int i = 0; i < arr.length; i++) {

            if (min > arr[i]) {

                min = arr[i];

                // arr[i]>min 23>0 23 45>23

            }

        }

        System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

        System.out.println("Smallest Value:" + min);

        System.out.println("Largest Value:" + max);

        System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

        for (int i = 0; i < arr.length; i++) {

            System.out.println(arr[i]);

        }

        sc.close();

    }

}