Java...

Java is a general purpose object oriented programming language.

Java was developed by Sun Microsystem in the year 1991 to 1995.

Initially it is known as "Oak" by James Goslings. - one of the inventor of Java.

Note :- The java file extension is .java.

The Reason behind the Name "JAVA"

In Java J stands for James Goslings. and AVA are the three other scientist first character of there names to contribute for developing JAVA.

OOPs :- "Object Oriented programming".

OOPs is an approach that provides modular programming style by creating separate ~~position~~ partition memory area for both Data and Function (Method). and assign such template to web and programmer on demand.

Features of OOPs:—

(1) Class
(2) Object
(3) Data Abstraction
(4) Encapsulation
(5) Inheritance
(6) Polymorphism
(7) Dynamic Binding
(8) Message Passive.

Special Features of JAVA...

(1) JAVA is a "PLATFORM INDEPENDENT" programming language. ("Architecture Free")

(2) JAVA is a two-stage programming language. (compiler and Interperator)
↓
JVM

(3) Auto Garbage collector.

(4) Multi-thread programming language.

(5) Object - Oriented

(6) Dynamic

(7) Secure

(8) Distributed

(9) Web Support.

(10) Simple and Easy to understand

(11) Easy to Extend.

The JAVA Environment ...

The java Environment is surrounded by JDK and API.

(1) JDK (Java Development Kit)

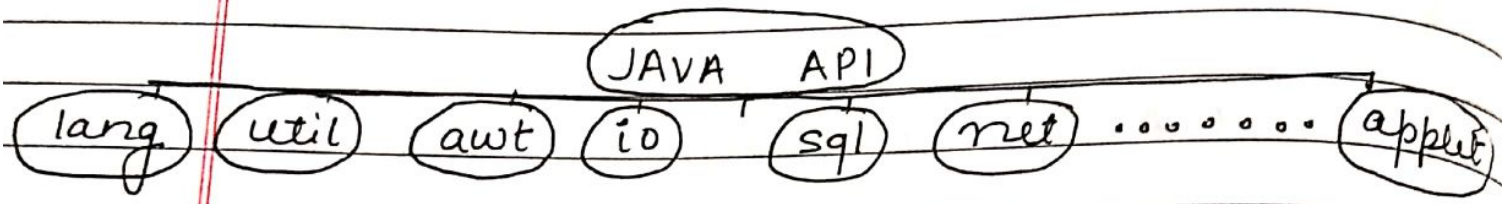
It consist all the necessary tools for compiling and Executing a JAVA program.

Some important JDK components are :-

JDK Components	Purpose
(1) Javac	– Java Compiler
(2) Java	– Java Interperator (JVM)
(3) Java (RMI) (Remote Method Interface)	– Java Remote Method Invocation.
(4) Applet Viewer	– Web Browser.
(5) Javadoc	– Java Document.
(6) Javap	– Java path
(7) Javah	– Java headerfile.

(2) The JAVA API – (Application programming Interface).

The JAVA API consists all the pre-defined Java Package for designing and developing a Java program.



Package... Collection of classes.

Note :—

Default Package	:	lang
Smallest Package	:	Applet
awt	:	Abstract windows tool
lang	:	language
Util	:	
Sql	:	
Net	:	Networking
I/O	:	Input / Output.

(Q) W.A.P in JAVA to display "HELLO JAVA" ?-

```
class a
{
public static void main (String KC])
  {
System. out. println ("HELLO JAVA");
        }
        }
```

Execution steps:-

(1) Open Command prompt.
(2) path = c:| program File Java|jdK 1.8.0-161 |bin
(3) cd desktop
(4) Java c a. Java.
(5) Java a

/* HELLO Java. */.

(Q) How to set the System path as Java Home path?

(1) Open Advance System Setting
↳ Environmental Variable.
↳ Press the new button of either system variable or user variable for core Java but for advance -

(2) JAVA always click the new button of System Variable.

Variable name - Path

Variable Value - c:| Program files |Java |JdK 1.8.0-144 |bin;

and press ok button.

(Q) Why to set the system path as Java Home Path?

⇒ To access all the JDK components (Javac, java, javap, ...) from anywhere to our computer system and execute the Java code from anywhere in our system.

Why the java main method is public and static ?

Public:

It is access specifier and Java main method is public so that it can be access from anywhere (both inside and outside the class)

Static:

It is a keyword and the Java main() method is static to call without creating class object.

void:-

It is predefined (built in) datatype and if a function have void as return type i.e., it doesnot return any value.

PAGE NO. :
DATE : / /

main:

Pre defined method and program execution starts from main()

public static void main (String K[])

* The statement "System. out. println ();"

System...

System is a pre-defined Java class under the Java default package lang.

It is a final class, in Java final is a keyword and a final class never be inherits.

Out...

It is static output object.

Println()...

It is a pre-defined method (function) to perform the write operation on output screen.

Note:— It comes under the printstream class of Java "io" package.

Ex:—
```
import java. io. Print stream;
class a
{
public static void main (string K[])
{
print stream obj = new Print stream (system. out);
obj. Println (" HELLO JAVA");
}}
```

* The difference b/w println() and print() method :—

Both are the predefined java method of PrintStream class and used to write content on output screen.

But println() write with a line Break, whereas print() write without any line break.

Ex: (1) System.out.println("HELLOJAVA");
System.out.println("HELLO JAVA");
Output :— HELLO JAVA
HELLO JAVA

(2) System.out.print("HELLO JAVA");
System.out.print("HELLO JAVA");
Output :— HELLOJAVA HELLO JAVA

(Q) The difference between import.java.io.*; and import java.io.PrintStream;

→ import java.io.*; // import all the classes of in package.

import java.io.PrintStream; // import only Printstream class of io package.

* int a, b, c;
  a = Integer.ParseInt (K[0]);

All the user predefine java classes are starts with uppercase.

All the predefine java methods starts with lowercase but having two or more than two words then After first words all the other word start with uppercase.

Ex: nextInt()
nextLine()
indexOF()
LastIndexOF()

* Pre define method of Integer class to convert String to Integer.

PAGE NO.:
DATE: / /

PLATFORM INDEPENDENCY...

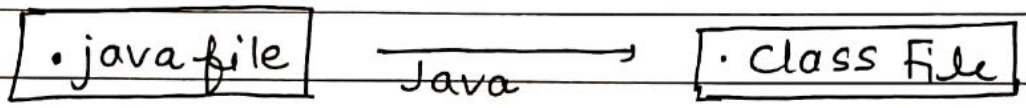
(Q) Why JAVA is platfoorm independency proguamming language?

↳ The term "PLATFORM INDEPENDENCY" is refers to a proguamming language that can be execute in any proguamming language either it can be windows, unix, mac, os etc.

(Q) How can Java acheive platfoorm independency?

↳ Java can acheive it by following two steps:-

Step 1:- COMPILATION



Source Code Byte Code

ex:- java a.java ↵

The java compiler converts the source code (.java file) into byte code (class file)

Step 2:—

.class file (Byte code) → Java (JVM) → Binary code (Machine code)

ex:— java a↵

In this stage the Java Interperator (JVM) converts the byte code to machine code (Binary code) that will be understandable by all the computer operating system. and so java can acheive the platform

AUTO GARBAGE COLLECTOR :—

Another important feature of Java. it means in Java JRE (Java Runtime Enviroment) there is a concept of autogarbage collector.

It is self- executed module to destroy the objects and free the resources.

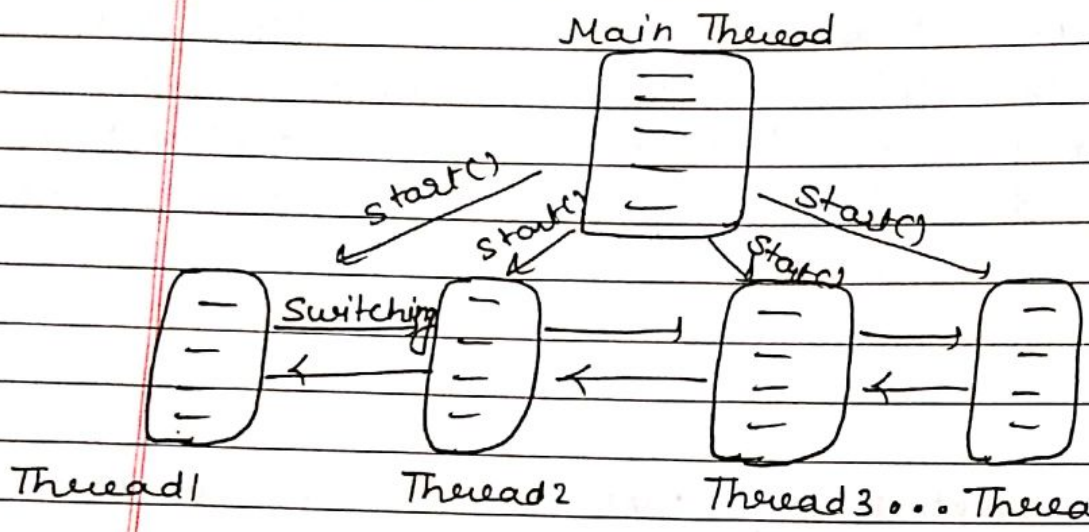
Note:— Java doesnot support user defined destructor (explicit destructor like c++).

It has the implicit destructor that is known as Auto Garbage Collector.

Note 2:— Java can acheive the concept of auto garbage collector by a pre- define method Finalize.

* Multithreading ...

Executing more than one program simultaneously under a single processor system is known as Multithreading or parallel computing.



In the above block diagram of multithreading we are executing 'n' no. of thread under the control of main thread.

Start is a pre-defined Java method to acheive a thread. and switching is a concept to acheive inter-process (interthread) communication.

Thread:- Light weighted process is known as thread. and process is a program in Execution.

★ Different way to providing Input in a Java program :-

(1) Command line Argument
(2) Scanner class
(3) Buffered Reader class

Command Line Argument ...

Passing Argument to main method is known as command line argument. Now consider the Java main method public static void main (String K[]) is always have string class object refrence as argument so java support the concept of Command line Argument.

↳ Advantage of Command line Argument...

To provide input in a java program directly from Command prompt.

(Q) W.A.P in JAVA to add two no using command line argument.

```
class cmd
{
public static void main (String K[])
{
int a,b,c;
```

```
a= Integer.parseInt (K[0]);
b= Integer.parseInt (K[1]);
    c=a+b;
System.out.println("sum="+c);
    }
    }
```

(Q) W.A.P in Java to check to a no is odd or even by command line Argument.

```
class a
{
public static void main(String K[])
 {
int x= Integer.parseInt (K[0]);
   if (x%2==0)
System.out.println("Even");
 else
System.out.println("odd");
    }}.
```

- Integer is a Wrapper class under the default java package lang and parseInt() is a method (Function) to convert string to Integer.

(Q) Check a no is Prime or not?

Prime no:- Prime no is a no that is greater than 1 and divided by 1 or it self only. In other words, prime numbers can't be divided by other number than it self or 1.
For example 2, 3, 5, 7, 11, 13, 17 are the prime numbers.

(Q) Reverse a no?

(Q) Greater between three no. using conditional operator only (?:)

(Q) Check a no. is Armstrong or no.

(Q) W.A.P in Java to get the current system Data time and year?

```
import java.util.*;
  class a
{
public static void main (String K[])
  {
system.out.println (New Data ()); // Data class under until package
  }
}
```

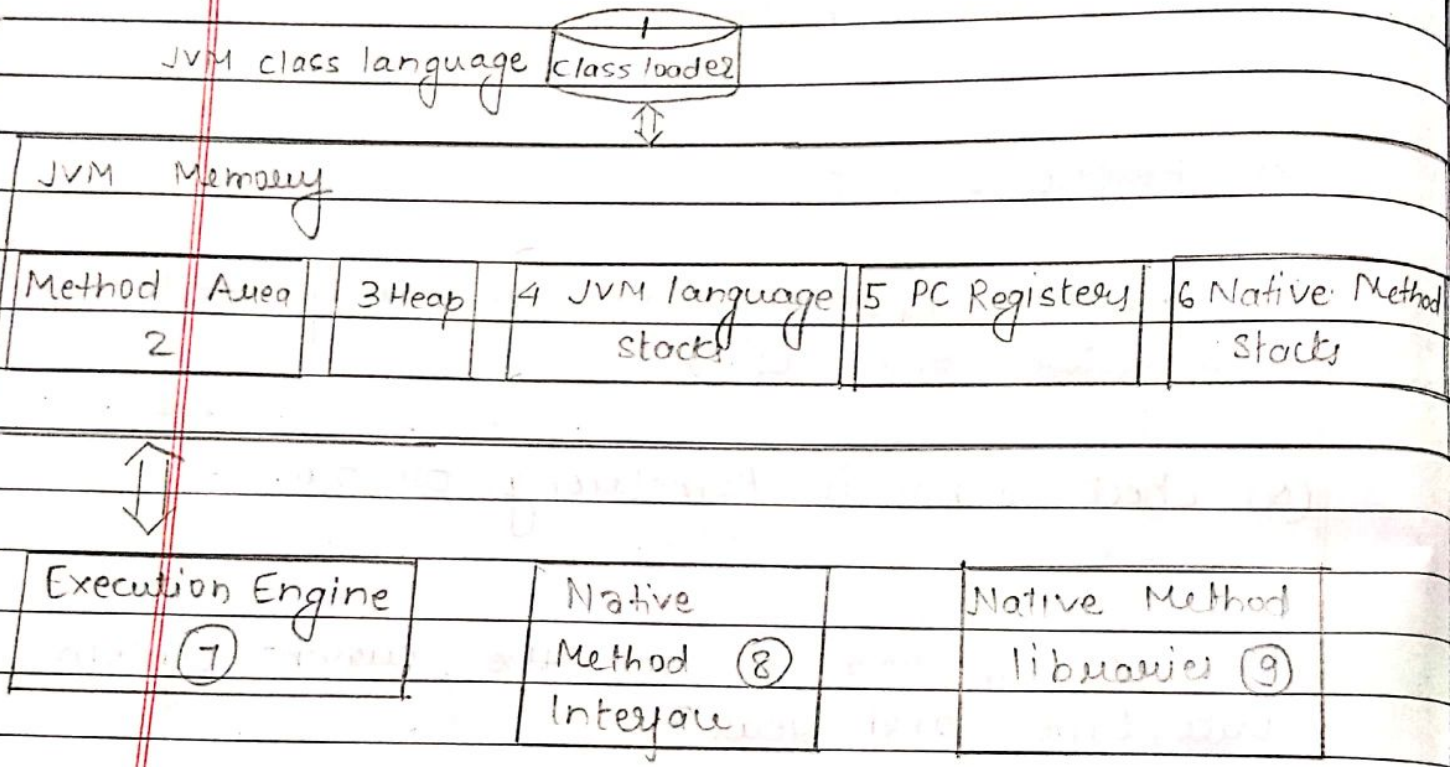
PAGE NO.:
DATE: / /

* JVM

JVM is a java Interperator that stands for " Java Virtual Machine".

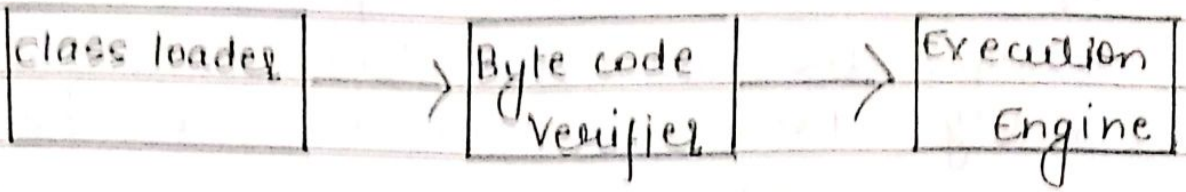
JVM is responsible for making Java a platform independent programming language.

JVM Architecture...



" JVM is a engine that provides runtime enviroment to drive the Java code or applications. It converts Java byte code into machine language. JVM is a part of JRE (Java Run Enviroment). It stands for Java Virtual Machine."

* JVM is responsible for allocating memory space.



Byte code :- Byte code is a special code generated by Java compiler to the Java interpreter (JVM).

(2) Scanner Class in Java ...

Comes under the util package and used to provide input in a Java program.

Methods of Scanner Class :-

Method		Purpose
(1) Next Int()	-	Input integers
(2) Next()	-	input a string without any space
(3) Next float()	-	
(4) Next Double()	-	
(5) Next line()	-	input a string with space.

#NOTE:- Scanner class have particular method for particular data type. So, it doesn't requisied conversion whereas the command line argument always need conversion. bcoz it have only string type.

(Q) W.A.P in Java to check the no. is prime or not by Scanner class.

→
```
import java.util.Scanner;
class a
{
public static void main (String K[])
{
int n, i;
Scanner obj= new Scanner (system.in);
System.out.println ("Enter A no");
  n= obj.nextInt();
for (i=2; i<n; i++)
  {
  if (n%i==0)
  break;
  }
if (i==n)
System.out.println("Prime");
else
System.out.println("Not Prime");
  }
  }
```

Array in JAVA...

(Q) WAP in java to create 'N' size integer array and find :-

(Q) Sum and avg of array element

(Q) Max and 2nd Max value of array.

(Q) Search

(Q) Sort

(Q) Sum of evens

(Q) Sum of primes.

→ Sum and avg of array elements.

```
import java.util.Scanner;
class arr
{
public static void main (String K[])
 {
int n, i, S=0;
Scanner obj= new Scanner (system.in);
System.out.println (" Enter no. of Data u
                    want to store in 1-D
                    array");
```

```
n= obj. nextInt();
int a[] = new int[n]; // n size dynamic integer array.
for (i=0; i<n; i++)
a[i] = obj. nextInt();
System. out. println("In Array are :- In")
for (i=0; i<n; i++)
{
System. out. print (a[i] +"It");
S= s+a[i];
}
System. out. println("In Sum of array:" + s);
System. out. println(" average of array:" + (s/n));
}
}
```

questions:-
(looping)

→ Reverse a no
→ Greatest b/w three no. by conditional operator (?:)
→ Table → Factorial → Prime or not
→ Palindrome or not → Armstrong or not
→ Series of prime between 1 to 100.
→ Series of armstrong between 1 to 10000?

```
String str = a>b && a>c ? "a is greater";
b>c ? "b is greater" : c>b ?
"c is greater" : "All are equal";
```

Vector Class in Java ...

```
import. java. util . vector;
    class  arr
      {
    public  static  void  main (string K[])
{
Vector  V  =  new  vector ();
    V. add (2543.7654);
    V. add (245);
    V. add ("MCR Ranchi");
    V. add ("$");
    System. out. println (V);
          }
          }
```

Q) Check the no. is Armstrong or not ?

(a) Command line argument
(b) Scanner class.
(For any digit no!)

Armstrong no!- 153#

$1^3+5^3+3^3 = 153$

1634

$1^4+6^4+3^4+4^4 = 1634$

```
import. java. util . Scanner;
    Class  arm
      {
public  static  void  main (string K[])
```

```
{
int i, d=0, i, s=0;
scanner obj = new scanner (System.in);
System.out.println(" Enter no");
        n = obj.nextInt();
for (i=n; i>0; i=i/10)
  d++;
For (i=n; i>0; i=i/10)
s= s+(int) (Math.pow ((i/10), d));
  if (s==n)
System.out.println(" Arm");
else
System.out.println ("Not Arm");
        }
        }
```

(Q) How to get the list of methods of a particular class :-

C:|users|Saima Haseeb|Desktop >javap java.util.Scanner

String :- javap java.lang.string.

javap java.awt.Graphics.

(8) Generate the series of Armstrong (any digit no) between a given lower to upper limit.

```
import java.util.Scanner;
class arm
{
public static void main (String K[])
{
int K, d=0, i, s=0, l, u, j;
Scanner obj = new Scanner (System.in);
System.out.println ("Enter lower and upper limit");
l= obj.nextInt();
u= obj.nextInt();
For (i=l; i<=u; i++)
{
For (j=i; j>0; j=j/10)
d++;
For (K=1; K>0; K=K/10)
s= s+(int) (Math.pow (K%.10), d));
if (s==i)
System.out.println (i+ "It");
d=0;
s=0;
}
}
}
```

RND - Research and Development ...

Ques:- Enter a pair of no and check whether they are co-prime or not?

co-prime :- 14 and 15
(1,2,7) (1,3,5)

Ques:- W.A.P in Java to check a no. is unique or not?

unique :- 123, 143
not unique, 122, 778, 777, 121.

(Q) W.A.P in Java to find the Greatest between three no. using Scanner class.

(Q) Factorial by Buffered Reader

(Q) Check a no. is palindrome by command line argument.

1. import java.util.Scanner;
class check
{
public static void main (String K[])
{
Scanner S= new Scanner (System.in);
int a, b, c;
System.out.println (" enter 3 no");
a= S.nextInt();
b= S.nextInt();
c= S.nextInt();
if (a>b && a>c)
{
System.out.println (" a is greter");
}
else if (b>c)
{
System.out.println (" B is greatest");
}
else
System.out.println (" C is Greatest");
}
}

3. class pald

```
{
public static void main (String K[])
 int n, r=0;
   n = Integer.parseInt (K[0]);
for (int i = n; i>0; i = i/10)
 {
  r = r*10 + (i%10);
    }
 if (r==n)
System.out.println (" Pald");
else
System.out.println (" Not pald");
    }
    }
```

Vector Class in Java ...

The vector class comes under java util package. " Vector is a any size type dynamic generic array".

Some Important method of Vector class...

Method	Purpose
(1) V.size()	It return the current size of vector class.
(2) V.addElement()	Add a new element in vector

(3) v.ElementAt(i) Return the ith element of vector. where v is an object of class vector.

- Advantage of Vector Class...

(1) Vector provides Generic Programming. in Java.

* Generic Programming is a special type of programming that is not type casted to a individual data type.

(2) Using vector we can create a dynamic array in Java.

A Dynamic array which size is not pre-defined, it will be grow at runtime

↳ Ex:

```
import java.util.vector;
  class vec
{
public static void main (String K[])
{
Vector v= new vector ();
System.out.println (v.size());
  v.addElement (88);
System.out.println (v.size());
v.addElement (7656.7656);
```

```
System.out.println (v.size());
v.addElement ('$');
System.out.println (v.size());
v.addElement ("Java is an Oops");
System.out.println (v.size());
System.out.println ("In Vector Elements are:-
                     In");
for (int i=0; i<v.size(); i++)
System.out.print (v.ElementAt (i)+"\t");
    }
}
```

Ques:- W.A.P in Java to search an element inside a vector :-

2-D Array in Java :-

Example:-

```
int a[][] = new int [3][3];
```

i.e., 'a' is an integer matrix of three row and three cols, so it can store 3*3=9 integers.

(Q) W.A.P in Java to create m*n size integer matrix (2-D array) and display the array elements.

→
```
imports java.util.Scanner;
class vec
{
public static void main (String K[])
{
Scanner S= new scanner ( System. in);
int r, c, i, j;
System. out. println(" Enter no. of rows and
                        cols");
   r =  S. nextInt ()
   c =  S. nextInt();
int  a[][]= new int [r] [c];
for (i=0; i<r; i++)
  {
for (j=0; j<c; j++)
  {
a[i][j]= S.nextInt ()
  } }
```

```
System.out.println("In Matrix Element : \n");
for (i=0; i<r; i++)
    {
    for (j=0; j<c; j++)
    {
    System.out.print (a[i][j] + "\t");
    }
    System.out.println();
    }
    }
```

Assignment on Array ...

(1) 1-D Array
- (a) Searching
- (b) Sorting
- (c) Find the max value of array
- (d) Find the 2nd max value of array
- (e) Sum and average of array
- (f) find the sum of evens.

(2) 2-D Array
- (a) Matrix multiplication
- (b) Addition of two matrix
- (c) Display the dialog element of matrix
- (d) Transposition of Matrix

(Q) Display right, left and both diagonal of a square matrix.

```
class mat
{
public static void main (String K[])
 {
 int i,j ;
int a[][]= {{1,2,3} {4,5,6} {7,8,9}} ; // static
                                          matrix
System.out.println ("\n left Dioa :-\n");
 for (i=0;  i<3; i++)
{
for (j=0; j<3; j++)
 {
if (i==j)
System.out.print (a[i][j]+"\t");
  else
System.out.print ("\t");
     }
System.out.println 0 ;
      }
 System.out.println ("\n Right Diog :-\n");
int r=2 ;
 for (i=0;  i<3;  i++)
   {
for (j=0 ; j<3 ; j++)
  {
if (j==y || i==j)
```

```
System.out.print(a[i][j]+"\t");
else
System.out.print("\t");
}
System.out.println();
y--;
}
}
}
```

Final Keyword in Java :—

Using the final Keyword, we can create

(1) Final Data
(2) Final class
(3) Final Method

(1) **Final Data** ... A final data never be modify (change).

example :— final int x = 190;
x++; // error can not assign value to final variable 'x'.

(2) **Final class** ... A final class never be inherits.

example.. final class a
{
}

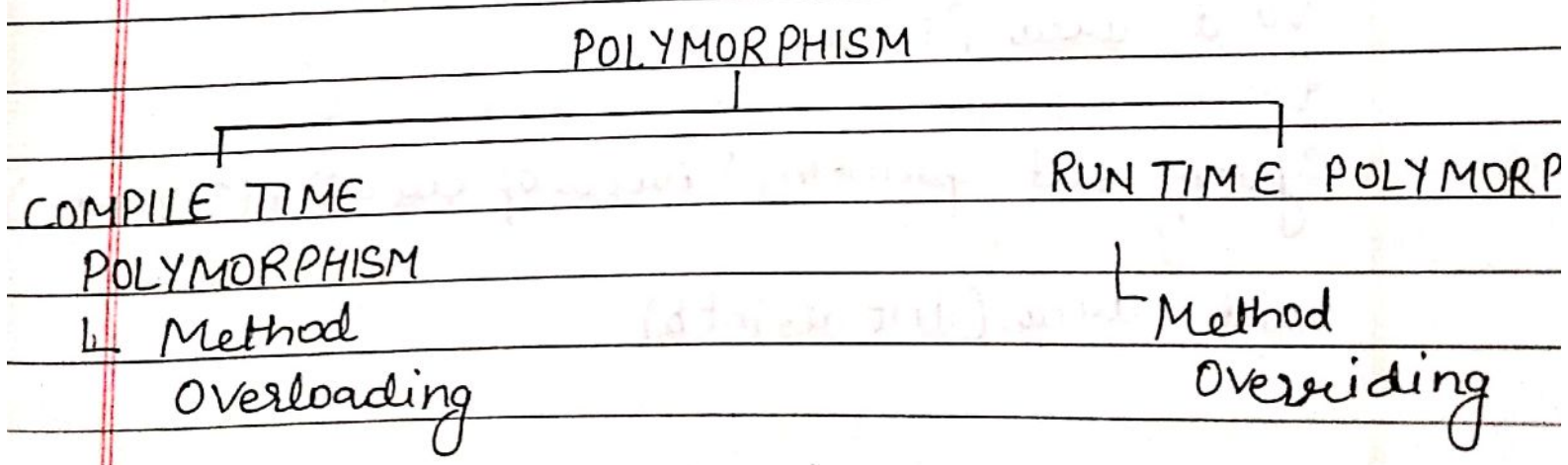
```
class b extends a // error, cannot inh.
                     from final class a.
{
}
```

(3) Final Method .. A final method never be override.

example .. Final void show()
```
{
.......
.......  // body of Final method.
.......
}
```

Polymorphism in JAVA ...

* Polymorphism ... It is an important concept of OOPs. It is a Greek term which english equivalent is "one name different form for different purpose".



(Q) Difference between method Overloading and method Overriding.

Answer... Both are the e.g. of polymorphism but there are some major differences between method overloading and method overriding.

Method Overloading	Method Overriding
(1) It is an example of compile time polymorphism.	(1) It is an example of runtime polymorphism.
(2) In method overloading, a group of method with same name but different argument list.	(2) In method overriding a group of method with same name and also same argument list.
(3) It doesnot need inheritance.	(3) It always have inheritance.

Prog. e.g:- Method Overloading ...

```
import java.util.*;
class a
{
void area (int a)
{
System.out.println("Area of cir = " + (3.14*a*a))
    }
void area (int l, int b)
```

```
{
System.out.println("Area of rect" + (l*b));
}
public static void main (String K[])
{
Scanner S= new Scanner (System.in);
a obj = new a();
System.out.println ("Enter radius");
obj.area (S.nextInt());
System.out.println ("Enter l and b");
obj.area (S.nextInt(), S.nextInt());
}
}
```

Method Overriding ...

```
class a
{
void show()
{
System.out.println("In Super class");
}}
class b extends a
{
void show()
{
System.out.println ("In Sub class");
super.show(); // Super is a keyword that works
                inside sub class to access
                super class method.
}
```

```
public static void main (String K[])
  {
b. obj = new b();
obj.show();
  }
}
```

Wrapper Class in Java...

A Group of classes in Java is known as wrapper classes like :-

Wrapper class	Data type
Integer	int
float	float
Double	Double
character	char
Boolean	Boolean.

note :- Wrapper classes are used to convert simple premitive data type (int, float, char, ...) into object type that is also known as Autoboxing. whereas converting object type is known as Unboxing.

(Q) W.A.P in Java to add two no, but without using 'int' as a datatype?

e.g of Autoboxing.

```
import java.util.Scanner;
class a
{
public static void main(String K[])
{
Scanner S= new Scanner (System.in);
System.out.println(" Enter two no");
  Integer obj1 = New Integer (S.nextInt());
                          // autoboxing
Integer .obj2= New Integer (S.nextInt());
System.out.println(" SUM = " (obj1.int value()+
                    obj2. int value()));
```

9-07-19

Unboxing :— It is opposite to the autoboxing so in unboxing we can convert the object type to simple primitive data type.

```
Class a
{ public static void main (String K[])
{ int x = 10;
Integer obj= new Integer (x); // auto boxing
System.out.println( obj. int value());
   x= obj; // unboxing
System.out.println (obj);
}}
```

Static Keyword in Java ...

Using the Keyword static we can declare:-

(1) Static data (class data)
(2) Static method (class method)

Static Data	Non- Static Data.
(1) Static Data is known as Class Data	(1) Non-static data is also known as instance data. (object - data)
(2) Static data is associated with the entire class.	(2) Non- static data is associated to the individual object.
(3) A single copy of static data is created and shared by all the object of same class.	(3) Multiple copies of non-static data is created for each individual object.
(4) Static data is directly access by class name.	(4) Non- static data is ~~directly~~ always access by object.

NOTE :- In Java both static and non-static data are default initialize by zero.

```
class a
{
static int x;
void count ()
 {
x++;
 }
void display()
{
System.out.println ("x= "+x);
 }
public static void main (String K[])
 {
public s.
 a obj1 = new a ()
a obj2 = new a()
a obj3 = new a()
obj1.count();
obj2.count();
obj3.count();
obj1. . . .
obj2.display();
obj3.display();
 }
}
```

O/P :-
x = 3
x = 3
x = 3

And if x is non-static (int x;) then
O/P
x = 1
x = 1
x = 1

(2) Static method:-
(class method)

Static method	Non-Static Method
(1) It is also known as class method.	(1) It is also known as instance (object) method.
(2) Directly call by class name without object.	(2) Always call by object without dot (.) operator.
(3) Always work with static data.	(3) Works with both static and non-static data.

```
class a
{
int x=10;
static void show
 {
int y=10;
System.out.println(y);
System.out.println
("Hello =" + x); //error, because x,
                  a no static data
                  member and.
 }
public static void main (String & CJ)
```

```
{
a.show();
}
}
```

NOTE .. The most famous static method in java is main method.

Array list... Array list is a predefined Java class under the util package. "The array list class is a logical extension of vector class so it is also a dynamic generic Array".

(Q) W.A.P in Java to implement Array list class:-

```
import java.util.*;
Class a
{
public static void main (String K[])
{
Arraylist obj = new Arraylist (.);
    obj.add (89);
    obj.add (76.876);
    Obj.add ('$');
    obj.add ("Java is OOPs");
    System.out.println (obj);
        }
        }
```

(Q) Difference between array list and Vector class.

Constructor in java...

Constructor is a special member method in java and it is special due to its following properties...

(1) The name of constructor is same as class name.
(2) It always declare in public.
(3) Never return any value even void.
(4) Auto executed, when the class objects created.
(5) A constructor construct or initialize the class object.

Types of Constructor...

(1) Default Constructor
(2) Parametrised "
(3) Copy "

Default Constructor... A constructor without any parametrel is known as Default constructor, and it is also known as non-parametrised constructor.

Parameterised Constructor ... This constructor should have some parameter.

Copy Constructor ... Copy Constructor is also a type of parameterised constructor with a limitation, i.e., it always have the class object reference as argument.

Note :- (1) So, all the copy constructor are parameterised constructor but all the parameterised constructor are not copy constructor.

(2) Constructor can easily overloaded

now consider the following example :-

```
Class a
{
a()
{
System.out.println("Default Constructor");
}
a(int x)
{
System.out.println("Parameterised Constructor
+x);
}
a(a obj)
{
System.out.println("copy constructor");
}
```

```
public static void main (String K[])
{
a a1 = new a(); // call the default
a a2 = new a(5); // call the parameter
a a3 = new a (new a()); // First call the
                          default then
                          copy.
}
}
```

(Q) W.A.P in Java to add two no in copy constructor but the no must be the class data member?

```
import java.util.*;
    class a
    {
Scanner s= new Scanner (System.in);
 int x,y; // data member
   a()
  {
 S.O.P ("Enter two no");
      x = s.nextInt();
      y= s.nextInt();
    }
a(a obj)
  {
S.O.P ("sum=" + (obj.x + obj.y));
    }
public static void main (String K[])
```

```
{
a a1 = new a();
a a2 = new a(a1);
    }
}
```

```
/*
a1    x = 1     & y = 2
a2    x = 0     y = 0        Sum = 0+0 = 0

a1    x = 1     y = 2
a2    obj.x = 1    obj.y = 2      1+2 = 3.

*/
```

(Q) W.A.P in Java to generate the series of prime b/w a given limit but the limits should be the parameters of parametrised constructor.

Lower to upper

(Q) Reverse a no. by default constructor.

(Q) Destructor in Java...

Java does not support explicit user define destructor like C++.

In Java JRE there is an implicit destructor known as AUTO GARBAGE COLLECTOR.

String handling in Java...

Sequence of data is known as String. In Java for string handlings we have four different classes.

1) String
2) String Buffer
3) String Builder
4) String Tokenizer

NOTE :-

The first three classes are under the java default package lang whereas the string tokenizer under the java util package.

(1) String class...

String class is a immutable class, It means it cannot be modify and suppose we do new object will be created.

Some important methods of String class.

Methods	Purpose
1) Str. length()	
2) Str. charAt (i)	
3) Str. 2ndex of ('a')	
4) Str. lastIndex of ('a')	

(5) Str. to upper case()
(6) Str. to lower case()
(7) str. get bytes()
(8) Str. trim()
(9) Str. sub String()
10) Str. start with()
11) Str. equals(str1)
12) Str. ends with()

(Q) W.A.P in Java to reverse a string and also check the string is palindrome or not.

```
import java.util.Scanner;
class a
{
public static void main (String K[])
{
Scanner S= new Scanner (System.in);
String str, str1 = " ";
S.O.P (" Enter a string");
 Str = S.next;
for (int i = str.length()-1; i>=0; i--)
 str1 = str1 + str.charAt(i);
S.O.P (str1);
if (str.equals(str1))
 S.O.P (" Pald string");
 else
 S.O.P (" Not Pald string");
 }
```

* String is pald or not ...

```
import java.util.*;
class a
{
public static void main (String K[])
Scanner s= new Scanner (System.in);
String str ;
int i, j, c=0 ;
S.O.P (" Enter a string");
   str = s.next();
For (i=0; j = str.length()-1 ; i< str.length();
          i++, j--)
if (str.charAt(i) == str.charAt(j))
   c++ ;
if (c== str.length()/2)
 S.O.P (" pald String");
   else
S.O.P (" Not pald string");
          }
          }.
```

Assignment on String handlings ...

(1) Count the no. of vowels in a strings.
(2) Merge two string into a third strings.
(3) Copy one strubg to other
(4) Convert lower case string to upper case without using touppercase() method.
(5) Enter a sentence and display the pald string only.

input : mom and dad
output : mom
dad

(6) Enter a sentence and display the largest word.
input : Java is a programming language.
output : programming.

(7) Enter a sentence and display the words start and ends with vowels.

input : ijKho hjghg ejhi hgfd
output : ijkho
ejhi

(Q) WAP in Java to convert lowercase string to uppercase without using to uppercase function.

```
import java.util.Scanner;
  class a
  {
public static void main (string Sai[])
  {
Scanner s= new Scanner (System.in);
System.out.println (" Enter a string");
      String str2 = " ";
  String str = new string (s.nextLine());
For (int i= 0; i< str.length(); i++)
str2 = str2+ (char) (((int) (str.charAt (i)) - 32))
  s.o.p (str2);
    }}
```

(2) String Buffer class...

It is pre-define java class under the default java package lang.

NOTE:

The main difference between String and string buffer class is that "String" is an immutable class (never (be modify) whereas String Buffer is a mutable class (easily modify)

↳ Difference:-

String	String Buffer
(1) It is immutable	(1) It is mutable.
(2) It is slower during concatination.	(2) It is faster during concatination.
(3) The length of string object is fixed.	(3) The length of the string Buffer can be increased.
(4) Consumes more memory.	(4) Consumes less memory.
(5) The object of String class is Stored in constant pool.	(5) Its object stored in heap memory.

* Some important method of String Buffer class :-

method	Purpose
(1) str. append()	Writing content from the end of existing string.
(2) str. insert (int, str1)	insert new content on given index.
(3) str. delete charAt (i)	delete the ith char of string.
(4) str. replace (i1, i2, str1)	replace sub string between i1 to i2 index by str1.

(Q) W.A.P in Java to justify the statement that string Buffer is a mutable class.

```
import java.util.scanner;
  class a
{
public static void main ( String K[]);
{
Scanner s= new scanner (System.in);
 s.o.p (" Enter a string");
String Buffer str= new stringBuffer ( s.next-Line
                                      ());
 s.o.p (str);
s.o.p (str. append (" initially Known as Oak")
```

```
S.O.P (str. insert (5, "MCR") ;
S.O.P (str. deleteCharAt (0)) ;
S.O.P (str. replace (0, 6, "ABCDE")) ;
        }
    }
```

(Q) W.A.P in Java to display the current system date and time without import util package.

```
System.out.println (new java.util.Date());
```

* (1) class a
```
{
{
float pi = 3.14f ;
    }
}
```

(2) class b
```
{
{
float pi = 3.14 ;
    }
}
error;
```

* Providing character input in Java at run time.

```
import java.util.*;
   class a
   {
public static void main (String K[])
 {
 char x;
Scanner s= new Scanner (System.in);
System.out.println ("Enter a charter");
     x= s.next().charAt(0);
     S.O.P (x);
      }
     }
```

Example :

```
System.out.println ((char) 95); // casting
                                   cast int to
                                   character
```

(Q) Enter a string and display the string without repeatation of character in sorted order.

```
class a
{
public static void main ( String K[])
 {
 int i, j;
  S.O.P (" Enter a string");
string str= new java.util scanner
          (System.in) .next();
```

```
for (i = 97 ; i <= 97+25; i++)
  {
for (j=0; j< str. length ; j++)
  {
char x = str. charAt (j) ;
int x+ = (int x) ;
    if (x1 = = i)
    {
S.O.P ( str. charAt (j)) ;
      break ;
        }
      }
    }
```

Recursion in Java ...

A function that call it self is called recursive function.

```
class a
 {
int i = 1;
 void d()
 {
if (i<=10)
 {
 S.O.P ("i=" +i) ;
    i++;
   d() ;
    }
  }
public static void main ( string k [])
```

```
{
a obj= new a();
}
}
```

* Factorial by Recursion...

```
class a
{
int i= 1, f=1;
void d (int n)
{
if (i<=n)
{
P= f*i;
i++;
d(5); // call itself so it is a recursive
        function.
}
else
System.out.println (f);
}
public static void main (String K[])
{
a obj =new a();
}
}
```

* Vowel check...

```
import java.util.*;
class a
{
public static void main (String K[])
{
Scanner S= new Scanner (System.in);
String str1, str2 = " ";
int i, j;
S.O.P (" Enter a Line");
str1 = S.nextline();
For (i=0; i<str1.length(); i++)
{
char x = str1.charAt (i);
if (x= ' ')
{
char x1 = str2.charAt (0);
char x2= str.CharAt (str2.length()-1);
if ((x1=='a' || x1=='e' || x1='i' || x1='o' ||
    x1='u')
or (x2='a' || x2='e' || x2='i' || x2='o' || x2='
   S.O.P (str2);
      str2 = " ";
    }
    else
    str2 = str2+x;
        }
        }
        }
```

String Tokenizer class :-

It is an inbuilt java class under the util package to break the string in tokens (words without space)

Method	Purpose
(1) nextToken()	move to the next word (token)
(2) hasMoreTokens()	return true until it got the next word (token) in line.

Ex:
```
import java.util.*;
class a
{
public static void main (String K[])
{
StringTokenizer obj = new StringTokenizer ("Java is an oops programming language");
while (obj.hasMoreTokens())
S.O.P (obj.nextToken());
}
}
```

O/P :-
```
Java
is
an
```

String Builder Class...

The string Builder class comes under Java default package lang. It is also a mutable class (class that can easily modify)

Some important method...

(1) Insert ()
(2) Append ()
(3) Char At ()
(4) delete (int, int)
(5) replace (int, int, string)

Ex:-

```
Class str
{
public static void main (String K[])
{
StringBuilder str= new StringBuilder
        (new java.util.Scanner (System.in)
                nextline ());
S.O.P (str.insert (0," Hello "));
S.O.P (str.append ("hello"));
S.O.P (str.Char At (0));
S.O.P (str.delete (0,5));
S.O.P (str.replace (2, 6," object "))
}
}
```

Note: The string class is immutable class whereas string Buffer and string Builder are mutable, but string Buffer is thread safe and unsynchronized but string Builder is thread unsafe.

Note:- The Kyword synchronized is used in Java multithreading and when a synchronized method or block is in execution no other thread can execute the same piece of code at time even we are in multi thread system.

INHERITANCE AND INTERFACE IN JAVA...

Inheritance is another important concept of OOPS, where a sub class can inherits some or all properties of its associated super class.

Advantage:

(1) Reuseability.. Using Inheritance we can achieve reusibility in Oops.

(2) By Reuseability the length of this program is reduced so it compiles and execute much faster.

Types :-

(1) Single
(2) Multiple
(3) Hierarchical
(4) Multilevel
(5) Hybrid

NOTES:-

JAVA does not support multiple Inheritance directly but it will achieve by a new concept "Interface."

* Single Inheritance

(Q) W.A.P in Java to implement Single Inheritance?

```
import java.util.*;
  Class a
 {
Scanner S= new Scanner (System.in);
  int l;
   void getl()
   {
 SOOP ("Enter l");
   l= S.nextInt();
       }
       }
Class b extends a
 {
int b1;
```

```
void getb()
  {
S.o.p ("enter b");
b1 = S.nextInt();
    }
void area()
{
S.o.p (" Area= " + (l*b1));
  }
PSVM (String S[]).
{
b1 = new b();
b1.getl();
b1.getb();
b1.area();
  }
  }
```

) Multilevel Inheritance...

A sub class derives another sub class.

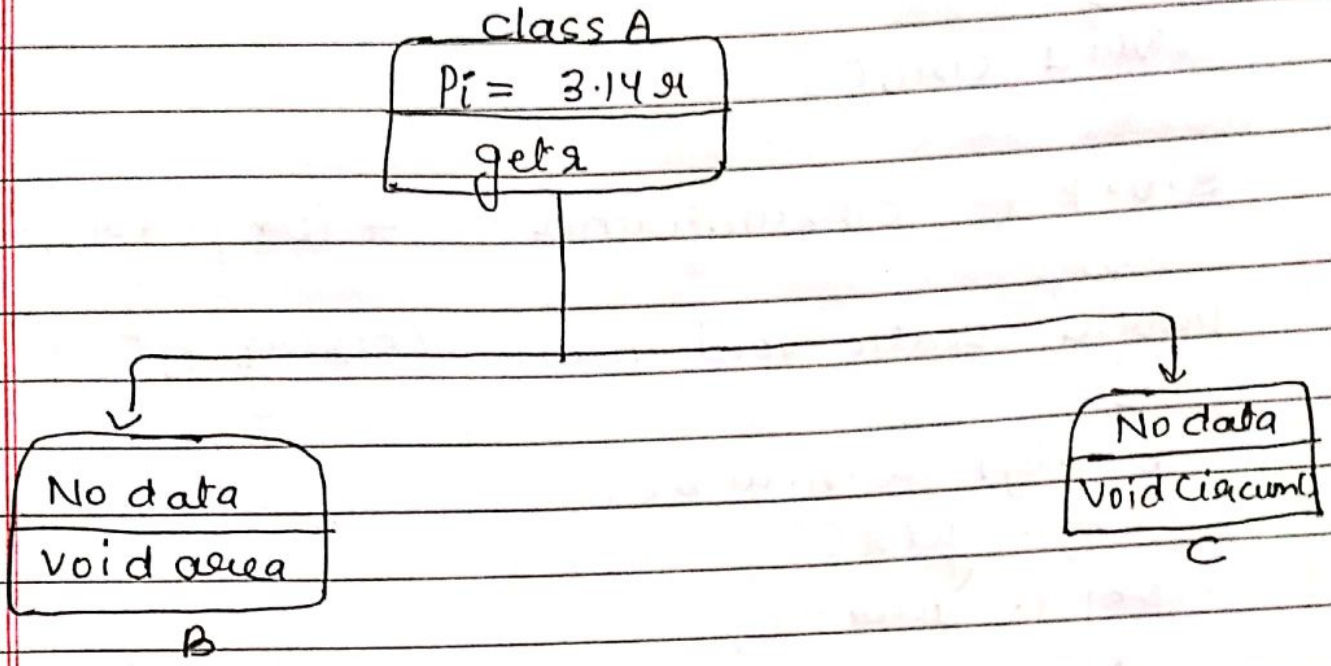
(8) W.A.P in Java to implement Multilevel Inheritance?

```
import java.util.*;
class a
{
Scanner s= new Scanner (System.in);
int l;
void getl()
```

```
{
s.o.p ("enter l");
l= s.nextInt();
}
}
class b
{
int bl;
void getb()
{
s.o.p ("Enter b");
bl = s.nextInt();
}
}
class c
{
void area()
{
s.o.p ("Area=" +(l*bl));
}
public static void main (String K[])
cbl= new c();
bl.getl();
bl.getb();
bl.area();
}
}
```

(3) Hierarchical Inheritance...

A super class derives atleast two sub class.




```
import java.util.*;
   class a
   {
Scanner sc= new scanner (system.in);
 float pi = 3.14f;
  int r;
 void getr ()
  {
System.out.println("Enter radius");
 r= sc.nextInt ();
    }
  }
class b extends a
 {
 void  area()
{
S.O.p ("area = " + (pi * r * r));
```

```
    }
}
class c extends a
{
void circ()
{
S.O.P ("circumference =" + (2* pi * r));
}
public static void main (String K[])
{
b obj1 = new b()
obj1. getr();
obj1. area();
c obj = new c();
obj. getr();
obj. circ();
    }
}
```

* Assignment on loop :—

(Q) Reverse a no.
(Q) Greatest between three by conditional operator
(Q) Table
(Q) Factorial
(Q) Prime
(Q) Armstrong
(Q) Pald
(Q) Series of Prime between 1 to 100
(Q) Series of Armstrong between 1 to 100.

(1) Reverse :

```
import java.util.Scanner;
class rev
{
public static void main (String K[])
{
Scanner sc= new Scanner (System.in);
int n, r=0;
System.out.println ("Enter a no.");
      n=  sc.nextInt ();
for (int i=n ; i>0; i=i/10)
      r= r*10 + (i%10);
System.out.println (" Reverse = " +r);
        }
        }
```

(2) Ge Greatest b/w three :

```
import.java.util.*;
class check
{
public static void main (String K[])
{
Scanner sc= new Scanner (system.in);
int a, b, c;
System.out.println (" Enter three no ");
   a= sc.nextInt ();
   b= sc.nextInt ();
   c= sc.nextInt ();
System.out.println ((a>b&& a>c) ? a + " is greatest no
   : ((b>c) ? (b:c) +" is greatest ");
                }}
```

(3) Mable :

```
import java.util.Scanner;
class Mable
 {
public static void main (String K[])
 {
Scanner S = new Scanner (System.in);
int n, i;
System.out.println("Enter no");
 for (i=1; i<=10;  i++)
System.out.println(n+"*"+i+"="+(n*i));
    }
    }
```

(4) Factorial :

```
import java.util.*;
 class Fact
  {
public static void main (String K[])
{
Scanner SE = new Scanner (System.in);
int n, i, F= 1;
System.out.println(" enter a no");
      n= sc.nextInt();
For (i= 1; i<=n; i++)
     F=F*i;
   System.out.println("Factorial = "+F);
        }
    }
```

(5) Prime:

```
import java.util.*;
class prime
{
public static void main (String K[])
{
Scanner sc = new Scanner (System.in);
int n, i, count = 0;
System.out.println ("Enter a no");
    n = sc.nextInt ();
for (i=2; i<n; i++)
{
if (n%i == 0)
{
count ++ ;
break;
}
}
if (count == 0)
System.out.println (" Prime");
else
System.out.println (" not a Prime");
}
}
```

(6) Pald:-

```
import java.util.*;
class pald
{
public static void main (String K[])
{
Scanner sc= new Scanner (System.in);
int n,i, s=0;
System.out.println ("Enter a no.");
    n= sc.nextInt ();
For (i=n; i>0; i=i/10)
{
s= s*10 + (i%10);
}
if (s==n)
System.out.println(" pald no");
else
System.out.println(" Not Pald");
}
}
```

* INTERFACE...

INTERFACE is same as class so it also have data and members but there is two major difference b/w class and interface that is :—

a) All the data members of interface are by default final.
b) All the method of an interface are by default abstract.

Final data: never be modify

Abstract method: Just declare inside a class or interface but there body must be implemented in derived or sub-class

example:-

```
interface a
{
double pi = 3.14 ; // Final data
double compute (); // Abstract method
}
```

* Advantage of Interface...

(1) Multiple and Hybrid inheritance will acheived by using interface in Java.

(2) Interface is a pure Abstract class.

* The difference between Interface and Abstract class...

Abstract class	Interface
(1) Abstract class can have abstract and non-abstract method.	(1) It can have only Abstract method.
(2) It may contain non-final variable.	(2) Variable declared in Java Interface is by default final.
(3) It can provide the implementation of interface.	(3) It can't provide the implementation of abstract class.
(4) It can have class member like private, protected etc.	(4) Member of a Java Interface are public by default.
(5) It can be extended using Keyword extends.	(5) It can be implemented using Keyword implements.

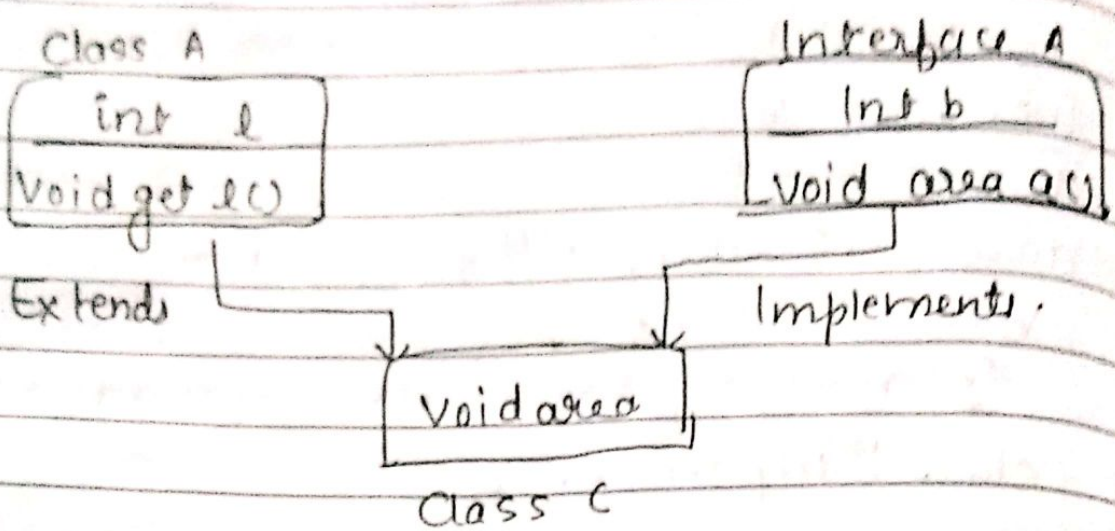
What is the neccessity of INTERFACE in Java and why interface having default final data default Abstract class.

(8) W.A.P in Java to Implement INTERFACE...

Hierarchical Inheritance:

```
Interface a
{
double pi = 3.14 ; // final data
 void area() ;
  }
class b implements a
 {
public void area()
  {
S.O.P ("Area =" + (pi * 8*3));
      }
    }
Class b implements a
 {
public void area()
  {
S.O.P (" circf =" + (2*pi *3));
    }
public static void main (String Sj[])
  {
b b1 = new b();
   b1. area();
c c1 = new c();
   c1 . area();
    }
   }
```

(Q) W.A.P in Java to implements multiple inheritance using interface;




```
import java.util.*;
class a
{
int l;
Scanner s= new Scanner (system.in);
void getl()
 {
s.o.p ("Enter l");
 l= s.nextInt();
   }
   }
interface b
 {
int b1=9; // final data
void area (); // abstract method
   }
Class c extends a implements b
```

```
{
public void area()
{
S.O.P ("Area =" + (l * bl));
}
public static void main (string aj[])
{
c obj = new c();
obj.get l();
obj.area();
}
}
```

(Q) W.A.P in Java to implement multiple inheritance where both the super class must be interface.

```
interface a
{
int l=8;
}
interface b
{
int bl=9;
void area();
}
class c implements a, b
{
public void area()
{
S.O.P ("Area = "+ (l * bl));
}
```

```
public static void main (String s[])
{
c obj = new c();
obj.area();
}
}
```

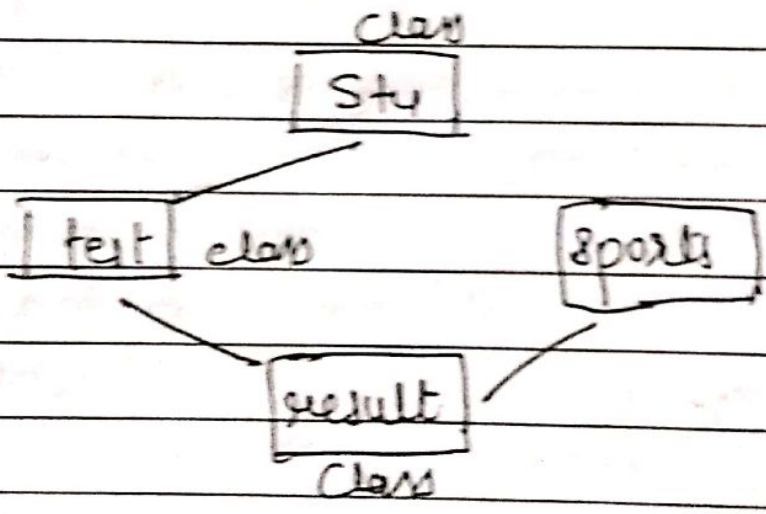
NOTE:-

We never create object of interface.

Note 2:-

When a class inherits the class the keyword extends is used and whenever a class inherits an interface the keyword should be implements.

8) W.A.P in Java to implement Hybrid Inheritance.



Package ...

Package: Collection of classes, interfac and also the sub packages is known as Package..

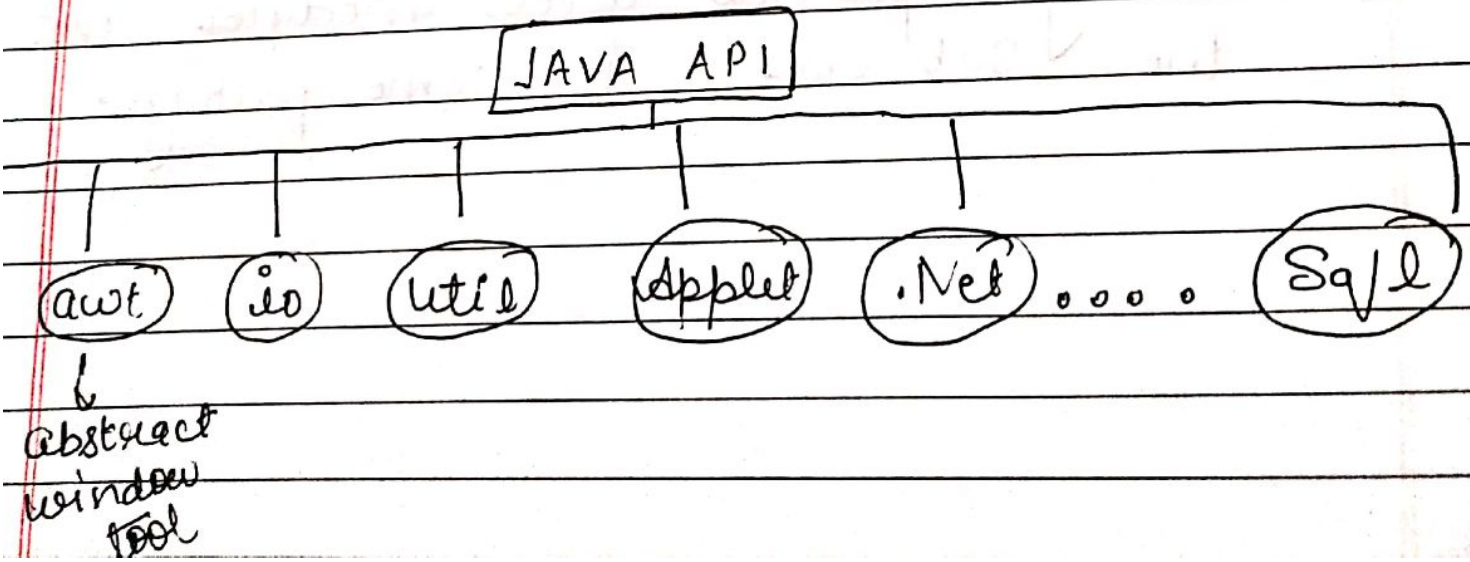
Package is a java way to organize thousands of Java Package according to the common functionality. So that retrival becomes easier.

Types of Java Package...

(1) System define Java Package (Pre-define)
(2) User defined Java Package.

(1) System Define Java Package...

These packages are the predi -d Java package API.



(2) User-defined Java Package...

These package are created by the user as per their requirement.

* Advantage of Java Package...

(1) Package provides an organised structure for thousand of Java classes.

(2) Using Package we can achelve reusability in Java.

System.out.println (Math.pow (2,3));

o/p = 8.0

The class math comes under the Java lang Package.

(3) As per requirement a user can also create own package.

(4) Package provide access modifier inside the sub class in same package.

(Q) Write th down the step for creating our own Java Package:-

Step1:- Create a folder (package) anywhere in your system.
Suppose D:| myPack

Step2:- Create a class and save inside your own package.

Ex:
```
package mypack;
public class my
{
public void show()
{
System.out.println("It is my Package");
}
}
```

Save on the location...

D:| mypack |my.Java

Where package is a keyword in Java that specify that my is a class inside our own package "mypack".

Step3:- Compile the above file by command prompt.

steps: Open cmd
D: ↵
cd MyPack ↵
javac my.Java ↵

Step4:- Now import our own Java package by writing the following Java code.

```
import mypack.*;
class we
{
public static void main (String K[])
{
my obj = new my();
obj.show();
}
}
```

O/P :→ It is my package.

8) abc → package

└→ XYZ → class

└→ int fact (int) → method

// The method fact should be static.

* Access Specifier / Modifier in Java :

The access modifier defines the visibility of the variables, class, constructor, method and also the data members.

In Java we have following four access modifier.

(1) Public
(2) Private
(3) Protected
(4) Default.

	Default	Private	Protected	Public
Same class	Yes	Yes	Yes	Yes
Same package sub class	Yes	No	Yes	Yes
Different package sub class	No	No	Yes	Yes
Different package non subclass	No	No	No	Yes
Same package non sub class	Yes	No	Yes	Yes
	Yes		Yes	

NOTE:-

In Java the default visibility mode is officially default (No keyword required).

But the difference b/w public and default is so thin so in general public is also consider as the default visibility mode of java.

* Hide class in Java ...

A class inside a package without public access modifier is known as hide class.

Example:

```
package mypack;
class a // Hide class
{ ---
---- // body of hide class.
---
}
```

NOTE :—

A hide class is not accessible outside the package.

Purpose of Hide class...

(1) All the classes inside the package are not created for the purpose to use outside the package.
Some classes are also created as a utility class to other classes in same package.
These classes should be hide class.

(2) Super class inside the package are also hide class.

* Static import : It is a concept to avoid writing the class name in case of access the static member of that class.

Ex:- import-static.
Out is a static output object of system class so if we import system as static then we just have to write : out.println()
in place of system.out.print

Ex:— import static . java.lang.system.*;
Class a
{
public Static void main (string s[])
{
out.println("Hello");
}
}

Exception Handling

Exception handling ...

Exception is also a type of error that creati problems at program execution.

Error ...

Any type of programming mistake known as error.

Types of Error ...

1. Compile time error...

Syntax related error that can be easily identify by the compiler.

2. Run time error...

logical error, These error never be identified by compiler and creates problems at runtime.

Exception Super class in Java...

It comes under the Java default package and consists all the necessary subclass to handle different type of exception.

Objects

Error

Throwable ..... Exception

IO Exception
Socket Exception
Server Exception
File Not Found Exception
Interrupt Exception
.
.
.
.
.
.

Run time Exception
Arithmatic Exception
Null Pointer Exception
Number Format Exception
ArrayIndexOut of Bound- - Exception
String Index Out of Bound- - Exception.
.
.
.

* **Note:-**

IO Exception is also known as checked Exception.

Run time Exception is also refer as unchecked Exception.

* Advantage of Exception Handling...

Due to Unhandled exception we have to face following problems:-

(1) Unusual termination of program at the point of exception occur.
(2) Sometimes we also get the strange output like infinite, garbage values, stack overflow etc.

All these problems will be solved by applying Exception handlings.

NOTE :

Exception handlings is also useful for error debugging.

* How to handle Exception...

Step 1-

Identify the statement that is responsible for generating the Exception.

Step 2-

Put that statement inside the try block.

Step 3-

After the end of the try block there must be a catch block to handle the exception.

Syntax:

```
try
{
....
.....  // try block, statement
...
catch (Exception obj)
{
------
--------  // catch block to handled
--------  the exception.
}
```

(5) W.A.P in Java to explain exception Handling.

```
Class a
{
public static void main (String K[])
  {
  int x = 9/0; // Arithmatic exception ! by zero.
  S.O.P ("Hello Java");
  }
}
```

The above program compiled successfully but at the time of execution (Runtime). it immidia terminate at the point of exception create due to unhandled exception.

Now, we rewrite the same program with exception Handling.

```
class a
{
public static void main (String K. [])
{ try {
int x = 9/0 ; // Arithmatic Exception by zero
}
catch (Exception obj)
{
S.O.P (" exception handelled !!");
}
S.O.P {" Hello Java");
}
}
```

(B) W.A.P in Java to explain ArrayIndexOutOfBound Exception.

```
Class a
{
PSVM ( String K [])
{
int x[] = {1,3};
S.O.P (x[9]) ; // ArrayIndexOutOf. Bounds-Exception
}
}
```

(Q) W.A.P in Java to explain Null pointer Exception.

It arise when we use an object without creating.

```
java.util.*;
Class a
{
Static Vector V;
PSVM (String K[])
{
V.addElement (10); // Null pointer Exception
                   because we just create
                   the reference of vector
                   not the object.
}
}
```

(Q) How to create our own exception sub class :—

In Java it is possible to create own exception subclass by extending Exception super class.

(Q) WAP in Java to enter a number if the number is 10 then a general message 10 otherwise an Exception message by your own exception subclass its not 10.

```
Class myexception extends Exception { }
  class a
  {
public static void main (String k[])
    {
    int n;
System. out. println(" Enter no");
try
{
n= new java. util. Scanner ( System. in)
     next Int ();
if (n==10)
System. out. println(" It is 10");
else
throw new myexception ();
      }
  catch (myexception obj)
System. out println (" It is not 10");
      }
    }
  }
```

(Q) WAP in Java to enter a string if the string is INDIA then a normal message It's India otherwise an exception message by our own exception sub class It's not India.

* The difference b/w checked and unchecked Exception...

Checked Exception	Unchecked Exception
(1) These checked Exception must check by compiler that either we handelled or not, if not then an exception message "Unreported 90Exception" must be catch or declare.	(1) These exception never be checked by the compiler that you handeled or not and create error at runtime.
(2) Examples are all the Java IOException like; Seaver Exception, Socket Exception, File Not Found Exception etc.	(2) Examples are all the Java 'RunTimeException like: NumberFormat Exception, Null Pointer Exception, Arithmatic Exception etc.

(3) Perogramming example:
import java.io.*;
class a
{
PSMV (String S[])
{
FileWriter F= newFileWriter ("K.txt");
F.write ("Hello Java");
F.close();
}
}
When we compile the perogram
	(3) Perogramming example:
	{
	int a = 9/0;
	S.O.P = ("Hello");
	}
	}
	perogram execute successfully even it have Arithmatic Exception that is unchecked Exception

throw an compile time Exception " Unreported IO Exception must be catch or declare" and program terminates due to unhandeled Exception.

Now, we can rewrite the following code with exception handling like;

```
import java.io.*;
class a
{
PSMV (String S[]) throws
                IOException
{
FileWriter F= new FileWriter
            ("K.txt");
F. write (" Hello Java");
 F. close();
   }
  }
```

Now we can rewrite the following code with exception handling like;

```
import .java.io.*;
  class a
{
PSMV (String K[])
   {
  try
  {
  int    a=9/0;
   }
catch (Exception obj);
{
S.O.P (" Exception handled
   }
S.O.P (" Hello Java");
   }
  }
```

Multiple catch Block for a Single try Block :

We can easily create multiple catch block to a single block by using the following syntax:

```
try
{ ...
  ------
  ------
  ------
}
catch (Exception Subclass1 obj1) { }
catch (Exception Subclass2 obj2) { }
   :
   :
catch (ExceptionsubclassN objN) { }
finally
{
---------
---------  // Finally block
---------
}
```

Finally :- It is a Keyword used in Java exception handling and a finally block is always executed either the exception is handled or not.

* Difference b/w:-

(1) final
(2) Finally
(3) Finalize

(8) W.A.P in Java to explain the multiple catch blocks for a single try block?

```
class a
{
psvm (String obj[])
{
try
 {
  int x= 9/0;
 }
catch (Arithmatic Exception K1)
 {
 S.O.P ("AE");
 }
catch ( Null Pointer Exception K2)
 {
 S.O.P (" AIOOBE");
 }
finally
 {
S.O.P ("I am always with u");
 }
 }
}
```

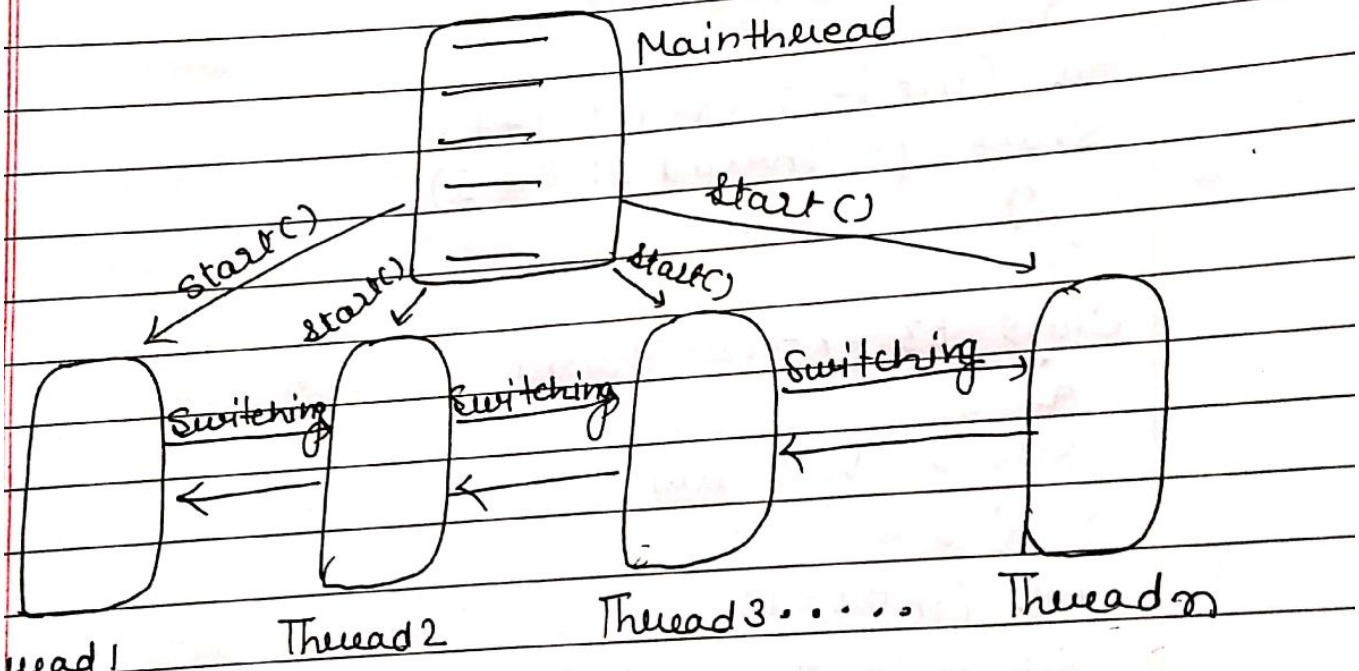
DATE: / /

Multi-Threading...

Multithreading...

Executing more than one program simultaneously under a single processor system is known as Multithreading or Multitasking.

For example:
All the modern day os like unix, windows, MAC-os etc.



In the above example we are executing more than one thread simultaneously under the control of main thread where start is a method to active a thread and switching is a

concept to acheive interthread (process) communication.

(8) W.A.P in Java to implement multi-thread

```
class t1 extends Thread
  {
public void run()
 {
for (int i=1; i<=10; i++)
S.o.p (" Thread 1: "+i);
    }
  }
class t2 extends Thread
  {
public void run()
  {
for (int i=1; i<=10; i++)
S.o.p (" Thread 2: "+i);
  }
}
class t3 extends Thread
  {
public void run()
  {
For (int i=1; i<=10; i++)
S.o.p (" Thread 3: "+i);
    }
public static void main (String K[])
```

```
{
    t1 ob1 = new t1();
    t2 ob2 = new t2();
    t3 ob3 = new t3();
    ob1.start();
    ob2.start();
    ob3.start();
}
```

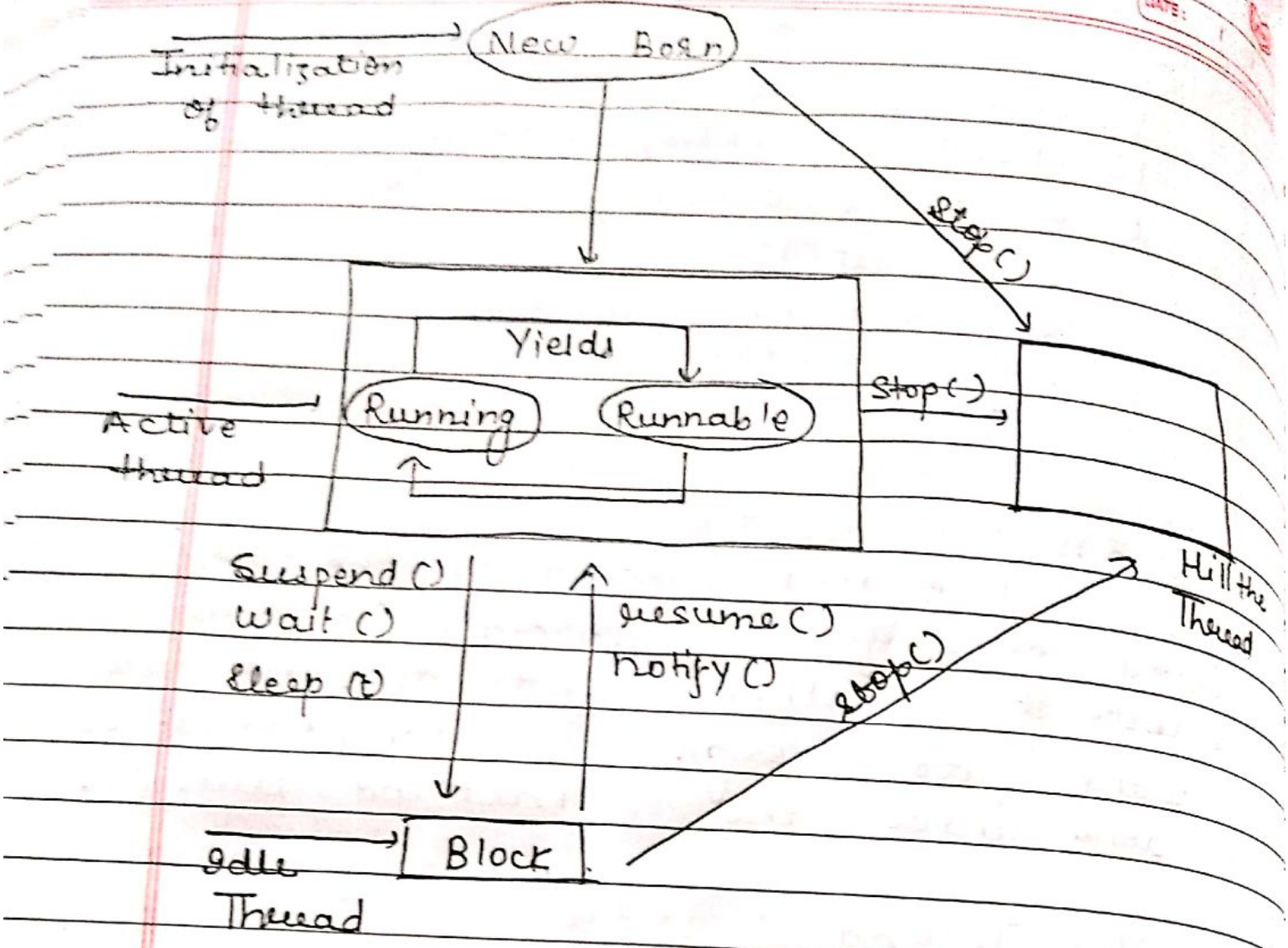
NOTE ..

The run method is the heart and soul of any thread. It is an abstract method and any task you want to assign on thread must be done inside the method run().

* The Thread lifecycle:

A thread goes through different stage in its life cycle, it includes

(1) New Born
(2) Running
(3) Runnable
(4) Block
(5) Dead.



* How to set the thread property :-

We can easily set thread priority by calling the method setPriority() of thread super class.

example :-

```
ob1. setPriority(10); // max Priority
ob2. set Priority(5); // avg Priority
ob3. set Priority(1); // min priority.
```

NOTE:

The method setPriority() consist an integer argument b/w 1 to 10.

NOTE:

If two thread have same priority then FCFS will be applied.

(Q) Create three sub thread and assign the last thread max priority and 2nd last have the min priority and the 1st thread have the average priority.

NOTE:

The Run is an abstract method of Runnable Interface that also overrides in Thread super class so we also implement run() method in thread class.

```
* class t3
  {
  PSVM (String a[])
    {
StringBuffer obj = new StringBuffer ("abc");
StringBuffer obj1 = new StringBuffer ("abc");
if (obj1.toString().equals (obj1.toString ()))
   S.O.P ("SAME");
   else
```

```
S.o.p ("NOT SAME");
    }
}
```

Synchronized Keyword in Java Multithreading

In Multithreading there are some situation when we have to prevent the parallel execution.

For example consider the famous reading-writing problems in a file where reading always be simultaneous but writing always be exclusive (one at a time).

Syntax:

```
synchronized (obj)
{
- - - - - -
- - - - - - - //synchronized block.
- - - - - -
}
```

When a synchronized block is in execution no other thread can execute the block at same time even we are in multithr-ading.

(Q) W.A.P in Java where one thread generate the series of prime b/w 1 to 100 and another thread generate the series of even b/w 1 to 100 both thread must synchronized?

```
class a
{
int i, j ;
 void prime()
{
For (i=2; i<=10 ; i++)
 {
For (j=2; j<i; j++)
  {
  if (i%.j==0)
   break;
   }
if (i==j)
s.o.p (i);
    }
  }
void even()
  {
For (i=2 ; i<=10; i=i+2)
  s.o.p(i);
   }
}
Class t1 extends thread
  {
String str;
```

```
a ob;
t1 (String s, a obl)
{
str=s;
ob= obl;
}
public void run()
{
Synchronized (ob)
{
Ob. prime();
S.O.P (str);
Ob. even();
}
}
PS VM (String Sj[])
{
a obj = new a();
t1. obj1 = new t1 ("Thread 1 Exit", obj);
obj1. start();
}
}
```