

Java Introduction:

- The Project of Developing a new language called java started in the year 1990 by sun Microsystems.
- Project was lead by James Gosling & his 4 others team members.
- The Version jdk 1.0 was released in 1996 for public use.
- Now currently JAVA was taken control by Oracle.
- you can give certification exams in java to have good future ahead in java software Development & they are OCPJP - oracle certified Professional java programmers. (Core)
OCWCD - Oracle Certified Web Component Developers. (Advanced)

Features of Java:

Simple : having syntax similar to C++ code.

Object Oriented : Java is Pure OOP.

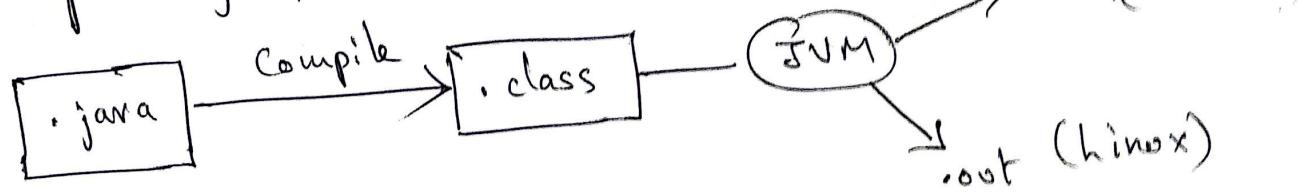
Distributed : using java we can develop network based Applications.

Secure: We can design virus-free Applications 2/8
using java.

Robust: Java has better Exception Handling }
Memory Management }

Platform Independent: Java code will be converted
into .class file (byte code file which is platform
Independent)

JVM helps to convert .class file to specific
Operating System code (Machine Dependent code)



Introduction to OOP:

1) Encapsulation: Binding variables, Methods in a single unit & providing them a better security using Access Specifiers.

Example: class Sample
{
 int a;
 private data()
 {
 //
 }
}

2) Poly morphism: 'Poly' means many 'Morphism' means forms which means in existing Many forms.

Poly morphism in JAVA

Static

Function Overloading

Constructor Overloading

Def: Binding of Methods to an object at compile time

also known as early binding

Dynamic

function overriding

Def: Binding of Methods to an object in the Run time
also known as Late Binding

3) Abstraction: Is Hiding Implementation Details.

will be existing in JAVA using Abstract classes & Interfaces

4) Inheritance: Accessing variables & Methods from another class. Donor is parent class & Receiver is Super class Sub class



Is Inheriting A

ex:

class A

{

{

class B extends A

{

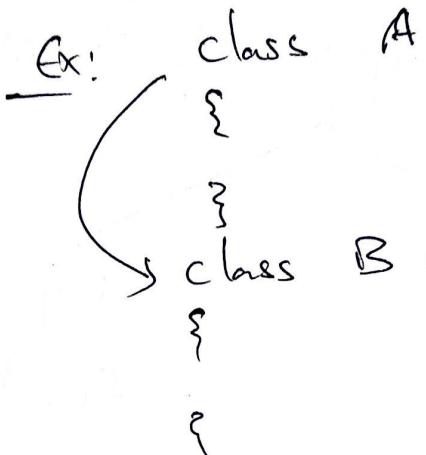
{

Diff forms of Inheritance:

Single level:



Single sub class inheriting Single Super class



* All variables } of A
Methods }

will be copied to B.

Multi level:



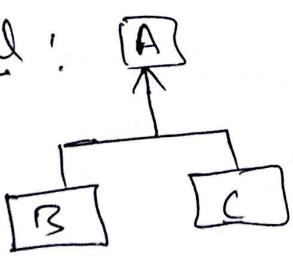
Ex: class A

class B extends A

class C extends B

}

Hierarchical:



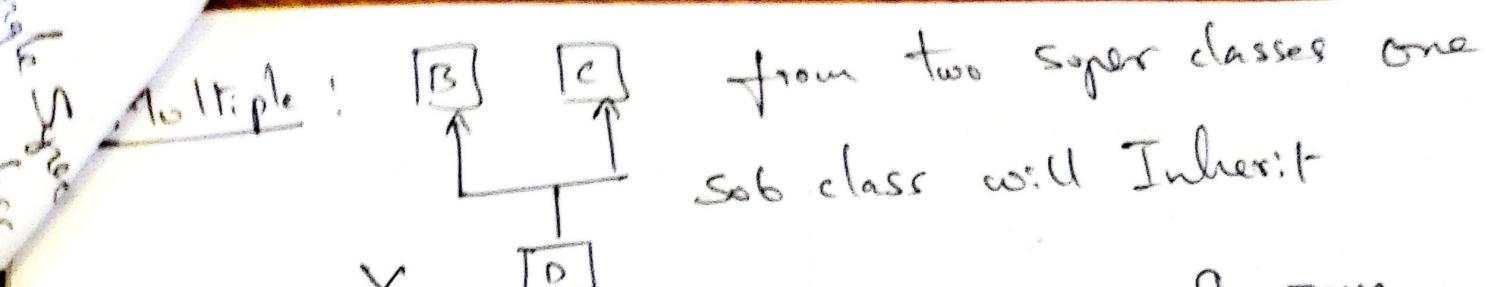
from Single Super class Two Sub classes inheriting

Ex: class A

class B extends A

class C extends A

}



from two super classes one
sub class will inherit

but it is prohibited in JAVA

Note: In JAVA we can't extend multiple classes

class B

{

}

class C

{

class D extends B, C X is not allowed.

{

}

Standards of JAVA:

- Create a project first in Net beans
- Create a package & then create java class file
- Always name a program with a class name in which main() function exists.

Example: Sum.java // program Name

```

class Sum
{
    public static void main (String args)
    {
    }
}

```

✓

A
E
C

— Class names should be defined as follows	
Ex: Sum ✓	<u>Wrong:</u> X
Student	sum
StudentDetails	student
Factorial	student
	Details //space not allowed.
	factorial

Note: First letters of each word should be Capital

Ex: Sum String System Scanner

— Methods should be as.

read()	{	if single word all in small case
readDet()		
findFact()		

read()	if two words first word in small
readDet()	
findFact()	

read()	Second word starting letter Cap.
readDet()	
findFact()	

Variables:

Predefined: short, int, long, double, float

char

for strings

String

(which is not a datatype
its a class)

Ex: int a;

short b;

float Avg;

double distance;

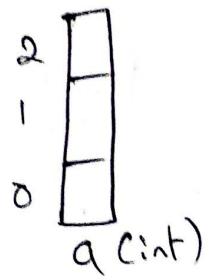
char ch;

String sname;

* (here sname is an object of

String class)

Arrays: $\text{int } a[3];$ $\text{int}[3] \text{ a};$



Static variable allocation:

✓ $\text{int } a[3] = \{13, 14, 15\};$

Array index start from 0 to n-1

a	0	13	$a[0]$
	1	14	$a[1]$
	2	15	$a[2]$

2D Arrays:

$\text{int } \{2\}\{3\} \text{ a};$ (or) $\text{int } a[2][3]$

→ means 2 row 3 column (2×3)

0	1	2	
0	$a(0,0)$	$a(0,1)$	$a(0,2)$
1	$a(1,0)$	$a(1,1)$	$a(1,2)$

Static allocation:

✓ $\text{int } a[2][3] = \{\{13, 14, 15\}, \{16, 17, 18\}\};$

Dynamic Allocation of Arrays:

2D

$\text{int } a[2][3];$

```
for(int r=0; r<2; r++)
{
    for (int c=0; c<3; c++)
    {
        a[r][c] = 2;
    }
}
```

Ex:	0	1	2
	0	2	2
	1	2	2

1D

$\text{int } a[3];$

```
for (int i=0; i<3; i++)
{
    a[i] = i;
```

}

a	0	1	2
Ex:	0	1	2

First Program :-

First.java // may name

```
public class First  
{
```

```
    public static void main (String ars)
```

```
    {  
        System.out.println ("Welcome to JAVA");  
    }
```

{about Main Method of JAVA}

Here public — everywhere available

static — can be accessed directly from
class reference , no object needed

void — returns nothing

& if main takes String array as Input

Comments in JAVA:

// single line comment

/*
First line;
Second line;
*/

(group comment)

Basic Programs :

11 Factorial . java

```
public class Factorial
{
    public static void main (String args)
    {
        int num, res;
        Scanner sc = new Scanner (System.in);
        System.out.println ("Enter Num:");
        num = sc.nextInt();
        res = 1;
        for (int i=1; i<=num; i++)
        {
            res = res * i;
        }
        System.out.println ("Factorial = " + res);
    }
}
```

Output: Enter Num:

5 ↴

Factorial = 120

Q) // Sum.java.

```
public class Sum
{
    public static void main (String args)
    {
        int a, b, c;
        Scanner sc = new Scanner (System.in);
        a = sc.nextInt();
        b = sc.nextInt();
        c = a + b;
        System.out.println ("Sum = " + c);
    }
}
```

Output: 3 ↴
 4 ↴
Sum = 7

```
// Bigger.java  
public class Bigger  
{  
    public static void main (String args)  
    {  
        int a,b,c;  
        Scanner sc = new Scanner (System.in);  
        a = sc.nextInt();  
        b = sc.nextInt();  
        c = sc.nextInt();  
        if ((a > b) && (a > c))  
        {  
            System.out.println ("A is Bigger");  
        }  
        else if (b > c)  
        {  
            System.out.println ("B is Bigger");  
        }  
        else  
        {  
            System.out.println ("C is Bigger");  
        }  
    }  
}
```

④ II Fibonacci.java

```
public class Fibonacci
{
    public static void main (String a[])
    {
        int range = 20;
        int x = 0;
        int y = 1;
        int z;
        System.out.println (x);
        System.out.println (y);
        while (z <= range)
        {
            z = x + y;
            System.out.println (z);
            x = y;
            y = z;
        }
    }
}
```

Output:

0
1
1
2
3
5
8
13
21