CORE JAVA

(Short Notes)

What is JAVA?

=> JAVA is a object oriented programming language.

Who created JAVA and when its released?

=> James Grosling created java, its released on 1995.

Why JAVA considered as platform independent?

=> It is platform independent because it follows right ones and run anywhere protocol.

What is JVM and JRE?

=> JAVA virtual machine & JAVA runtime enviroment

* JVM is an abstract machine (JAVA virtual machine). It is a specification that offers runtime enviroment and allows the execution of java bytecode.

* JRE is a software layer that runs on top of a computer's operating system software and provide

the class libraries and other resources that a specific JAVA program needs to run.

Important keyword (which are reserved for JVM)

⇒ Abstract Assert boolean break byte case

Catch char class continue default do

double else enum extends final finally

float for if implements import instance of

int interface long native new null

package private protected public return short

Static StrictFp super switch synchronized this

throw throws transient try void volatile

While True false

Types of datatypes:-

(1) primitive & non primitive

★ primitive → byte, short, int, long

★ boolean, char

★ float, double

Non primitive → Array & String

Writing first code

```
public class FIRST CODE { // This is public class
    public static void main (String [] args) { // This is function
        System.out.println("Hello World"); // This is method Chaining
    }
}
```

Output : Hello World

Typecasting :- Typecasting is when you assign a value of one primitive datatype to another type.

Types of Typecasting

(i) Implicit Typecasting :- It means conversion of datatype from small to big and which is occuring internally

Ex → `int x => 80; // here int is small datatype`
`float y => x; // here float is big datatype`

`System.out.println (y);`

Output 80.0;

(ii) Explicit Typecasting :- It means conversion of datatype from big to small which is occuring externally

Ex: float y => 80f; // there float is big datatype.

int x => (int) y; // there int is small datatype.

System.out.println (x);

output: 80

Types of operatour in JAVA

☆ Arithmetic decision/ternary comparatour Assignment

What is operatour :- operatours are set of the symbols which is used in JAVA in order to control the flow of programme or implement a logic.

(i) Arithmetic :- + − % . / * < > ==

ex :- Int a => 10

Int b => 5

system.out.println(a+b);

output: 15

(iii) Ternary :- return (logic) ? if : else ;

ex :- int x => 10;

system.out.println (Test (x));

Public Static int Test (int x) {

Return (x > 0) ? 1 : 0;

}

Output: 1

(iii) Comparator :- == && || >= <= !=

ex :- int a => 5
int b => a
System.out.println(a==b);

Output: True

(iv) Assignment :- += -= *= %= /= ==

ex :- int x => 10

x += x; // x = x+x
System.out.println(x);

output: 20

==

☆ Making decision

→ In order to make decision in Java, we use various statements such as :- if else, Nested if else, Switch case, ternary operator.

☆ Syntax

1) if else

```
if (i<n) {
        // logic our print Statement
        System.out.println("Yes");
    }
    else {
```

```
        System.out.println("No");
    }
```

ii) Nested if else

a)
```
if (i<n) {
        System.out.println("Yes");
    }
else if (i==n) {
        System.out.println("Maybe");
    }
else {
        System.out.println("No");
    }
```

b)
```
if (i<n) {
        if (i==1) {
            System.out.println("Done");
        }
        else {
            System.out.println("Done it");
        }
    }
else {
        System.out.println("Ninja");
    }
```

iii) Switch case

```
Switch (n) {
    Case 1:
        System.out.println("Yes");
```

```
        break;
    case 2:
        System.out.println("No");
    break;
    default:
        System.out.println("maybe");
    break;
    }
```

☆ Working with loops

☆ While, do While, for loop, Advance for loop
break & continue, Nested loops

ex ☆

```
for(int i=0; i<10; i++){
    for(int j=0; j<i; j++){
        System.out.println(i);
    }
    System.out.println();
}
```

Nested loops

Ex ☆ Break & continue

```
for(int i=0; i<10; i++){
    if(i%2==0){
        continue;
    }else{
        System.out.println(i);
        break;
    }
}
```

Method Meaning in JAVA

A method / function is a block of code or collection of statements which are set together to perform a certain task it can be of parametised or non parameterised.

Types of Methods

void, return, Static

(i) public void method ( ) { } — Non parameterized void method means will return nothing

(ii) public int method (int i) { return i; } — Parameterized Return method means will return integer.

(iii) public static int method (int x, int y) { return 0; } — Parameterized Static method Means will return int call directly by name.

lambda expression were added in JAVA 8 it is a short block of code which takes parameter and return's a value they work like anonymous class or lets say anonymous function which doesn't need a name.

=> interface a {

    public void add (int x, int y);

}

```
public class lambda {
  public static void main (String [] args) {
    int x -> 10, y -> 25;

    a obj = (int z, int c) -> {
        int sum = x+y;
        System.out.println(sum2;
    };
obj.add(x, y);
  }
}
```

output: 35

What are anonymous class object or method ?

=> It is nothing but a nameless class, method, object and it makes your code more consise & short

```
Class xoxo {

  public void print() {
     system.out.println("hello there");
  }
}
public class anonymous {
  public static void main (String [] args) {

     new xoxo().print(); //anonymous Object
     new xoxo() { //anonymous class

        @override
        public void print() {
```