**Soft Suave technology Interview Questions**

**Round 1 MCQ:**

Qn 1

String x = "xyz";

x.toUpperCase();

String y = x.replace('Y', 'y');

y = y+"abc";

System.out.println(y);

Answer : xyzabc

Qn 2

A a = new B();

System.out.println("complete");

static class A{

public A(int x){}

}

static class B extends A{}

Qn 3

int i = 1,j = 10;

do{

if(i++>--j){

continue;

}

}while(i<5);

System.out.println("i = "+ i+"and j = "+j);

Answer : i = 5 and j = 6

Qn 4

ExamQn e = new ExamQn();

e.A(0);

System.out.println("j = "+e.j);

public static class ExamQn {

static int j;

static void A(int i){

boolean b;

do{

b = i<10 | B(4);

b = i<10 || B(8);

}while(!b);

}

static boolean B(int i){

j+=i;

return true;

}

}

Answer : 4

Qn 6

String textString = new String("java");

StringBuffer textBuffer = new StringBuffer("java");

stringReplace(textString);

bufferReplace(textBuffer);

System.out.println(textString + textBuffer);

public static void stringReplace(String text){

text = text.replace("j", "c");

}

public static void bufferReplace(StringBuffer text){

text = text.append("c");

}

Answer : javajavac

Qn 6

Tree tree = new Pine();

if(tree instanceof Pine)

System.out.print("Pine");

else if(tree instanceof Tree)

System.out.print("Tree");

else if(tree instanceof Oak)

System.out.print("Oak");

else

System.out.print("Oops");

static class Tree {}

static class Pine extends Tree{}

static class Oak extends Tree{}

Answer : Pine

Qn 7

final StringBuffer a = new StringBuffer();

final StringBuffer b = new StringBuffer();

new Thread(){

public void run(){

System.out.print(a.append("A"));

synchronized(b){

System.out.print(b.append("B"));

}

}

}.start();

new Thread(){

public void run(){

System.out.print(b.append("C"));

synchronized(a){

System.out.print(a.append("D"));

}

}

}.start();

Answer : ACCBAD

Qn 8

App myApp = new App(10);

System.out.println(myApp.index);

static class Super{

public int index;

}

static class App extends Super {

public App(int index){

index = index;

}

}

Answer : 0

Qn 9

try{

System.out.printf("1");

int value = 10/0;

throw new IOException();

}catch(EOFException e){

System.out.printf("2");

}catch(ArithmeticException e){

System.out.printf("3");

}catch(NullPointerException e){

System.out.printf("4");

}catch(IOException e){

System.out.printf("5");

}catch(Exception e){

System.out.printf("6");

}

Answer : 13

Qn 10

C c = new C();

overl(c);

static void overl(A a){

System.out.println("ONE");

}

static void overl(B b){

System.out.println("TWO");

}

static void overl(Object obj){

System.out.println("THREE");

}

static class A{};

static class B extends A{};

static class C extends B{};

Answer : TWO

Qn 11

SubC s = new SubC();

s.meth(123321);

static class SupC{

void meth(Number n){

System.out.println("From Super Class");

}

}

static class SubC extends SupC{

void meth(Double d){

System.out.println("From Sub Class");

}

}

Answer : From Super Class

Qn 12

new A().A();

static class A{

public A(int i){

System.out.println("1");

}

public A(){

this(10);

System.out.println("2");

}

void A(){

A(10);

System.out.println("3");

}

void A(int i){

System.out.println("4");

}

}

Answer : 1243

Qn 13

Test obj = new Test();

obj.set(20);

obj.get();

abstract class demo{

public int a;

demo(){

a = 10;

}

abstract public void set();

abstract final public void get();

}

static class Test extends demo{

public void set(int a){

this.a = a;

}

final public void get(){

System.out.println("a = "+a);

}

}

Answer : Compilation error

Qn 15

Test t = new Test();

static class Test{

public Test(){

System.out.printf("1");

new Test(10);

System.out.println("5");

}

public Test(int temp){

System.out.printf("2");

new Test(10,20);

System.out.printf("4");

}

public Test(int data, int temp){

System.out.printf("3");

}

}

Answer : 12345

**Similar Problems:**

**Here are 15 similar Java practice questions for interview-style preparation. These cover concepts like inheritance, method overloading, threading, static blocks, exception handling, object references, and more—modeled after the questions you shared.**

✅ Qn 1 - String Immutability

String s = "Java";

s.concat("World");

s.toLowerCase();

System.out.println(s);

Options:  
A) javaworld  
B) java  
C) JavaWorld  
D) Java  
Answer: D) Java

✅ Qn 2 - Constructor and Inheritance

class A {

public A(int x) {}

}

class B extends A {

public B() {

super(5);

}

}

public class Test {

public static void main(String[] args) {

B b = new B();

System.out.println("Done");

}

}

Output:  
Answer: Done

✅ Qn 3 - Do-While + Condition

int i = 0, j = 5;

do {

i++;

j--;

} while (i < j);

System.out.println("i = " + i + ", j = " + j);

Answer: i = 3, j = 2

✅ Qn 4 - Static Method Execution

public class StaticTest {

static int count;

public static void main(String[] args) {

check(5);

System.out.println(count);

}

static void check(int i) {

boolean b;

do {

b = i > 2 | increment();

b = i > 2 || increment();

} while (!b);

}

static boolean increment() {

count++;

return true;

}

}

Answer: 1

✅ Qn 5 - String vs StringBuilder

String s = new String("abc");

StringBuilder sb = new StringBuilder("abc");

modify(s, sb);

System.out.println(s + sb);

static void modify(String s, StringBuilder sb) {

s = s.replace("a", "z");

sb.append("z");

}

Answer: abcabcz

✅ Qn 6 - instanceof Check

Shape shape = new Circle();

if (shape instanceof Circle)

System.out.print("Circle");

else if (shape instanceof Shape)

System.out.print("Shape");

else

System.out.print("Unknown");

class Shape {}

class Circle extends Shape {}

Answer: Circle

✅ Qn 7 - Multithreading

final StringBuilder a = new StringBuilder("X");

final StringBuilder b = new StringBuilder("Y");

new Thread(() -> {

synchronized(a) {

a.append("1");

synchronized(b) {

b.append("2");

}

}

}).start();

new Thread(() -> {

synchronized(b) {

b.append("3");

synchronized(a) {

a.append("4");

}

}

}).start();

Answer: No fixed output (Thread scheduling — possible deadlock or interleaved)

✅ Qn 8 - Shadowing

class Super {

public int val;

}

class Sub extends Super {

public Sub(int val) {

val = val;

}

}

public class Main {

public static void main(String[] args) {

Sub obj = new Sub(5);

System.out.println(obj.val);

}

}

Answer: 0

✅ Qn 9 - Try-Catch Priority

try {

System.out.print("A");

int x = 5 / 0;

} catch (ArithmeticException e) {

System.out.print("B");

} catch (Exception e) {

System.out.print("C");

}

Answer: AB

✅ Qn 10 - Method Overloading

X x = new Z();

check(x);

static void check(Y y) { System.out.println("Y"); }

static void check(X x) { System.out.println("X"); }

class X {}

class Y extends X {}

class Z extends Y {}

Answer: X

✅ Qn 11 - Method Overriding vs Overloading

class Base {

void show(Number n) {

System.out.println("Base");

}

}

class Derived extends Base {

void show(Integer n) {

System.out.println("Derived");

}

}

public class Main {

public static void main(String[] args) {

Derived d = new Derived();

d.show(10);

}

}

Answer: Derived

✅ Qn 12 - Constructor & Method Confusion

new X().X();

static class X {

public X() {

this(100);

System.out.println("A");

}

public X(int x) {

System.out.println("B");

}

void X() {

X(5);

System.out.println("C");

}

void X(int x) {

System.out.println("D");

}

}

Answer: B A D C

✅ Qn 13 - Abstract Method Error

abstract class Alpha {

abstract final void show();

}

class Beta extends Alpha {

void show() {

System.out.println("Beta");

}

}

Answer: Compilation Error: Abstract method cannot be final

✅ Qn 14 - Static Block Execution

public class Test {

static {

System.out.println("Static Block");

}

public static void main(String[] args) {

System.out.println("Main Method");

}

}

Answer:  
Static Block  
Main Method

✅ Qn 15 - Constructor Chain

class Hello {

Hello() {

System.out.print("1");

new Hello(10);

System.out.print("5");

}

Hello(int x) {

System.out.print("2");

new Hello(x, 20);

System.out.print("4");

}

Hello(int x, int y) {

System.out.print("3");

}

public static void main(String[] args) {

new Hello();

}

}

Answer: 12345

**✅ Q17: SELECT employees whose name starts with 'R'**

SELECT \*

FROM Employee

WHERE empName LIKE 'R%';

🧠 **Explanation:**

* LIKE 'R%' means the name starts with 'R'.
* % is a wildcard matching any number of characters.

**✅ Q18: SELECT id, age, name in both ascending and descending age order**

**Ascending order of Age:**

SELECT empld, Age, empName

FROM Emp

ORDER BY Age ASC;

**Descending order of Age:**

SELECT empld, Age, empName

FROM Emp

ORDER BY Age DESC;

**✅ Q19: Total salary (assuming by each employee or department)**

Since the question says:

*"to calculate the total amount of salary on each from the below Emp table"*

Assuming you mean **total salary** (overall):

SELECT SUM(Salary) AS TotalSalary

FROM Emp;

If you meant **grouped by a category**, e.g., Age or Department (not present in table), here's how you'd write it (if department column existed):

SELECT Department, SUM(Salary) AS TotalSalary

FROM Emp

GROUP BY Department;

**🔁 Similar Practice Questions**

**1. Display names that end with the letter 'a'.**

SELECT empName

FROM Employee

WHERE empName LIKE '%a';

**2. Display all employee details whose age is greater than 28.**

SELECT \*

FROM Employee

WHERE Age > 28;

**3. Display the top 3 highest salaried employees.**

SELECT \*

FROM Emp

ORDER BY Salary DESC

LIMIT 3;

**4. Count how many employees are older than 30.**

SELECT COUNT(\*) AS OlderThan30

FROM Employee

WHERE Age > 30;

**5. Find the employee with the minimum salary.**

SELECT \*

FROM Emp

ORDER BY Salary ASC

LIMIT 1;

Perfect! Let’s go through each topic step by step with the following structure:

✅ 1. Age Problems

🔹 Core Concept:

Age problems are based on forming equations using current age, past/future age differences, and ratios. The key idea is to express all ages in terms of a single variable.

🔹 Tricks:

* Define the present age as x.
* Use phrases like:
  + “twice the age” → 2x
  + “age after 5 years” → x + 5
  + “age before 10 years” → x - 10
* Make sure both sides of the equation refer to the same time frame.

✏️ Example Q1:

A mother is twice as old as her son. 20 years ago, she was 10 times older than her son. What is the mother's present age?

Step 1: Let the son’s current age be = x.  
Then, mother’s current age = 2x

Step 2: 20 years ago  
Mother's age = 2x - 20  
Son's age = x - 20

Given: 2x - 20 = 10(x - 20)

Step 3: Solve  
2x - 20 = 10x - 200  
200 - 20 = 10x - 2x  
180 = 8x  
x = 22.5  
Mother’s age = 2x = 45

✅ Answer: D. 45 years

✏️ Practice Tip:

Start from the current age, then go backward or forward based on what the question asks.

🚤 2. Boats and Streams

🔹 Core Concept:

* Boat in Still Water (B)
* Stream speed (S)
* Downstream Speed = B + S
* Upstream Speed = B - S

Use the formula:

Speed = Distance / Time  
Time = Distance / Speed

✏️ Example Q2:

A boat can travel 36 km downstream in 3 hours and the same distance upstream in 6 hours. What is the speed of the boat in still water?

Downstream speed = 36/3 = 12 km/h  
Upstream speed = 36/6 = 6 km/h

Boat speed = (12 + 6) / 2 = 9 km/h

✅ Answer: A. 9 km/h

✏️ Practice Tip:

* First find downstream and upstream speeds using time and distance.
* Average them to get boat’s still water speed.
* Subtract them and divide by 2 to get stream speed.

💰 3. Interest Calculations

🔹 Core Concept:

A. Simple Interest (SI):

SI = (P × R × T) / 100

Where:

* P = Principal
* R = Rate (%)
* T = Time (years)

B. Compound Interest (CI):

A = P(1 + R/100)^T  
CI = A - P

✏️ Example Q3:

Find simple interest on ₹5000 at 5% for 3 years.

SI = (5000 × 5 × 3) / 100 = ₹750

✅ Answer: A. ₹750

✏️ Example Q4:

Compound interest on ₹4000 at 10% for 2 years

A = 4000 × (1 + 10/100)^2  
= 4000 × (1.1)^2 = 4000 × 1.21 = ₹4840

CI = 4840 - 4000 = ₹840

✅ Answer: C. ₹840

✏️ Practice Tip:

* Use SI for faster calculation.
* CI builds interest on top of previous interest — use formula with powers.

🧠 4. Logical Reasoning

🔹 Core Concept:

Based on patterns, sequences, coding-decoding, odd-one-out, and critical thinking.

✏️ Example Q5:

If FLOWER = GMPXFS, how is GARDEN coded?

Observe letter shift:  
F → G (+1), L → M (+1), O → P (+1), ...

Same pattern applies:  
G → H  
A → B  
R → S  
D → E  
E → F  
N → O

So, GARDEN → HBSFOM

✅ Answer: B. HBSFOM

✏️ Example Q6:

Find the odd one out: 2, 4, 8, 16, 31

Pattern: Multiply by 2  
2 × 2 = 4  
4 × 2 = 8  
8 × 2 = 16  
16 × 2 = 32  
⛔ 31 doesn’t follow the pattern.

✅ Answer: 31 is the odd one.

✏️ Practice Tip:

* Use elimination: Remove options that match a clear pattern.
* In coding, check for alphabet positions or shifts.