

# Java for Testers – Interview Questions and Answers Part-1

## 1) Write a simple Java program?

```
public class Simple{  
    public static void main(String[] args){  
        System.out.println("Hello World");  
    }  
}
```

## 2) Write a Java program to print an integer value?

```
public class PrintInteger{  
    public static void main(String[] args){  
        int i = 1; System.out.println(i);  
    }  
}
```

## 3) Write a Java program to print the command line arguments?

```
public class Sample {  
    public static void main(String[] args){  
        if (args.length > 0) {  
            System.out.println("The command line" + " arguments are:");  
            for(String val:args)  
                System.out.println(val);  
        } else  
            System.out.println("No command line " + "arguments found.");  
    }  
}
```

Command Line:

Bijans-MacBook-Pro:java bijan\$ javac Sample.java

Bijans-MacBook-Pro:java bijan\$ java Sample Java Interview Questions

The command line arguments are: Java Interview Questions

#### 4) Write a Java program to print the input from scanner?

```
public class PrintScannerInput {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        String name = sc.nextLine();  
        System.out.println("Name: "+name);  
    }  
}
```

#### 5) Write a Java program to convert from Fahrenheit to Celsius?

```
public class ConvertToCelsius {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        System.out.println("Input temperature in Fahrenheit: ");  
        double fahrenheit = sc.nextDouble();  
        double celsius = ((5 * (fahrenheit - 32.0)) / 9.0);  
        System.out.println(fahrenheit + " degree fahrenheit is equal to " + celsius + "in celsius");  
    }  
}
```

#### 6) Write a Java program to swap two numbers?

```
public class SwapNumbers {  
    public static void main(String[] args) {  
        int x, y, z;  
        Scanner sc = new Scanner(System.in);  
        x = sc.nextInt(); y = sc.nextInt();  
        System.out.println("Before swapping\n x = " + x + "\n y = " + y);  
        z = x; x = y; y = z;  
        System.out.println("After swapping\n x = " + x + "\n y = " + y);  
    }  
}
```

#### 7) Write a Java program to swap two numbers without using third variable?

```
public class SwapNumberWithoutVariable {  
    public static void main(String[] args) {  
        int x = 10;  
        int y = 5;  
        x = x + y;  
        y = x - y;
```

```

x = x - y;
System.out.println("After swapping: " + "x is " + x + " and " + "y is " + y);
}
}

```

### 8) Write a Java program to add two numbers?

```

public class AddNumbers {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int a = sc.nextInt(); int b = sc.nextInt();
        int sum = a + b;
        System.out.println("Sum of two numbers is: " + sum);
    }
}

```

### 9) Write a Java program to find the largest number?

```

public class FindLargestNumberInArray {
    static int array[] = {21,98,13,9,34};
    public static void main(String[] args){
        int maxNumber = findLargestNumber();
        System.out.println("Maximum number in the array: " + maxNumber);
    }
    private static int findLargestNumber(){
        int max = array[0];
        for(int i = 0; i < array.length; i++){
            if(array[i] > max)
                max = array[i];
        }
        return max;
    }
}

```

### 10) Write a Java program to demonstrate if..else statement ?

```

public class FindNumbersIsPrime {
    public static void main(String[] args){
        boolean flag = false;
        System.out.println("Enter input number");
        Scanner sc = new Scanner(System.in);
        int num = sc.nextInt();
        for(int i = 2; i <= num/2; i++){

```

```

if(num%i==0){
flag = true;
break;
}
}
if(!flag){
System.out.println(num + " is a prime number");
} else{
System.out.println(num + " is not a prime number");
}
}
}
}

```

## 11) Write a Java program to demonstrate nested if ... else if .. statement?

```

public class IfElseDemo {
public static void main(String[] args){
int i =20;
if(i == 10)
System.out.println("i is 10");
else if(i == 15)
System.out.println("i is 15");
else if(i == 20)
System.out.println("i is 20");
else
System.out.println("i is not present");
}
}

```

## 12) Write a Java program to demonstrate nested if ...else statement?

```

class Ladder {
public static void main(String[] args) {
int number = 0;
if (number > 0) {
System.out.println("Number is positive.");
} else if (number < 0) {
System.out.println("Number is negative.");
} else {
System.out.println("Number is 0.");
}
}
}

```

```
}
```

### 13) Write a Java program to find odd and even numbers ?

```
public class FindOddEvenNumbers {  
    public static void main(String[] args){  
        Scanner sc= new Scanner(System.in);  
        System.out.println("Enter a number: ");  
        int inputNumber = sc.nextInt();  
        findNumber(inputNumber);  
    }  
    public static void findNumber(int number){  
        if(number%2!=0){  
            System.out.println("Number is odd");  
        }  
        else {  
            System.out.println("Number is even");  
        }  
    }  
}
```

### 14) Write a Java program to compare two strings?

```
public class CompareTwoStrings {  
    public static void main(String[] args){  
        Scanner sc = new Scanner(System.in);  
        System.out.println("Enter first string");  
        String first = sc.next();  
        System.out.println("Enter second string");  
        String second = sc.next();  
        compare(first,second);  
    }  
    public static void compare(String s1, String s2){  
        if(s1.compareTo(s2)==0) {  
            System.out.println("Strings are equal");  
        } else {  
            System.out.println("Strings are not equal");  
        }  
    }  
}
```

### 15) Write a Java program to demonstrate for loop ?

```
public class ReverseSentenceWordByWord {
    public static void main(String[] args){
        String sentence = "Java Interview Questions";
        String reversedSentence = reverseSentence(sentence);
        System.out.println(reversedSentence);
    }
    public static String reverseSentence(String sentence){
        String reverse = "";
        String[] words = sentence.split("\\s");
        for(int i=words.length-1;i>=0;i--){
            reverse = reverse + words[i] + " ";
        }
        return reverse;
    }
}
```

### 16) Write a Java program to print stars using for loop, where the number of stars printed should be equal to the row number?

```
public class PrintStarPatternInRow {
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the number of rows");
        int rows = sc.nextInt();
        printStars(rows);
    }
    public static void printStars(int n){
        for(int i=0;i<n;i++){
            for(int j=0;j<=i;j++){
                System.out.print("* ");
            }
            System.out.println();
        }
    }
}
```

### 17) Write a Java program to demonstrate while loop?

```
class WhileLoopExample {
    public static void main(String args[]){
        int i=10;
```

```

while(i>1){
System.out.println(i);
i--;
}
}
}

```

### 18) Write a Java program to print the entered number in reverse?

```

public class ReverseNumber {
public static void main(String[] args){
Scanner sc = new Scanner(System.in);
System.out.println("Enter the number");
int num = sc.nextInt();
reverseNumber(num);
}
public static void reverseNumber(int number){
int reverse = 0;
while(number!=0){
int digit = number % 10;
reverse = reverse * 10 + digit;
number = number/10;
}
System.out.println("Reversed number " + reverse);
}
}

```

### 19) Write a Java program to demonstrate the usage of break statement inside while loop?

```

class BreakExample {
public static void main(String[] args) {
Double number, sum = 0.0;
Scanner input = new Scanner(System.in);
while (true) {
System.out.print("Enter a number: ");
number = input.nextDouble();
if (number < 0.0) {
break;
}
sum += number;
}
}

```

```

System.out.println("Sum = " + sum);
}
}

```

## 20) Write a Java program to demonstrate the usage of break and continue statements inside while loop?

```

public class ContinueExample {
    public static void main(String args[]) {
        int [] numbers = {10, 20, 30, 40, 5};
        for(int x : numbers ) {
            if( x == 30 ) {
                continue;
            }
            System.out.print( x );
            System.out.print("\n");
        }
    }
}

```

## 21) Write a Java program to print the alphabets using for loop?

```

public class PrintAlphabets
{
    public static void main(String[] args)
    {
        char i;
        System.out.printf("The Alphabets from A to Z are: \n");
        for (i = 'A'; i <= 'Z'; i++)
        {
            System.out.printf("%c ", i);
        }
    }
}

```

## 22) Write a Java program to demonstrate for each loop?

```

public class FindHighestMarks {
    public static void main(String[] arg) {
        int[] marks = { 125, 132, 95, 116, 110 };
        int highest_marks = maximum(marks);
        System.out.println("The highest score is " + highest_marks);
    }
}

```



```

public static int maximum(int[] numbers) {
    int max = numbers[0];
    // for each loop
    for (int num : numbers) {
        if (num > max) {
            max = num;
        }
    }
    return max;
}

```

### 23) Write a Java program for printing the Multiplication table?

```

public class MultiplicationTable {
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter number");
        int number = sc.nextInt();
        System.out.println("Enter multiplication range");
        int range = sc.nextInt();
        for(int i=1;i<=range;i++){
            System.out.printf("%d * %d = %d \n", number, i, number * i);
        }
    }
}

```

### 24) Write a Java program for printing the prime numbers?

```

public class FindNumberIsPrime {
    public static void main(String[] args){
        boolean flag = false;
        System.out.println("Enter input number");
        Scanner sc = new Scanner(System.in);
        int num = sc.nextInt();
        for(int i=2;i<=num/2;i++){
            if(num%i==0){
                flag = true;
                break;
            }
        }
    }
}

```

```

}
if(!flag){
System.out.println(num + " is a prime number");
} else{
System.out.println(num + " is not a prime number");
}
}
}
}

```

## 25) Write a Java program to check whether a given number is Armstrong ?

```

public class ArmstrongNumber {
public static void main(String[] args){
int c=0,a,temp;
Scanner sc = new Scanner(System.in);
System.out.println("Enter a number");
int num = sc.nextInt();
temp = num;
while(num>0){
a=num%10;
num=num/10;
c=c+(a*a*a);
}
if(temp==c) {
System.out.println(temp + " is an Armstrong number");
} else
System.out.println(temp + " is not an armstrong number");
}
}

```

# Java for Testers – Interview Questions and Answers Part-2

## 1) Write a Java program to print Floyd's triangle?

```

public class FloydTriangle {

```

```

public static void main(String[] args){
Scanner sc = new Scanner(System.in);
System.out.println("Enter the number of rows");
int rows = sc.nextInt();
printFloydTriangle(rows);
}

public static void printFloydTriangle(int n){
int number = 1;
for(int i=0;i<n;i++){
for(int j=0;j<=i;j++){
System.out.print(number + " ");
number++;
}
System.out.println();
}
}
}

```

## 2) Write a Java program to find all the sub-string of given string?

```

public class FindSubString {
public static void main(String[] args) {
String name = "Selenium And Java Interview Questions";
System.out.println(name.contains("Java")); // true
System.out.println(name.contains("java")); // false
System.out.println(name.contains("Interview")); // true
System.out.println(name.contains("questions")); // false
}
}

```

## 3) Write a Java program to print the given string in reverse?

```

public class ReverseString {
public static void main(String[] args){
Scanner sc = new Scanner(System.in);
System.out.println("Enter input string");
String s1 = sc.next();
String s2 = reverseString(s1);
System.out.println("Reversed String is: "+s2);
}

public static String reverseString(String s){
String rev="";

```

```

char[] arr = s.toCharArray();
for(int i=arr.length-1;i>=0;i--)
rev = rev + arr[i];
return rev;
}
}

```

#### 4) Write a Java program to check whether the given number is palindrome?

```

public class PalindromeNumber {
public static void main(String[] args){
int r,sum=0,temp;
Scanner sc = new Scanner(System.in);
System.out.println("Enter a number");
int n = sc.nextInt();
temp=n;
while (n > 0) {
r = n%10;
sum = (sum*10) + r;
n=n/10;
}
if(temp==sum)
System.out.println("Number is palindrome");
else
System.out.println("Number is not palindrome");
}
}

```

#### 5) Write a Java program to add two matrix?

```

public class AddTwoMatrix {
public static void main(String args[]) {
//creating two matrices
int a[][] = {{1, 3, 4}, {2, 4, 3}, {3, 4, 5}};
int b[][] = {{1, 3, 4}, {2, 4, 3}, {1, 2, 4}};
//creating another matrix to store the sum of two matrices
int c[][] = new int[3][3];
//adding and printing addition of 2 matrices
for (int i = 0; i < 3; i++) {
for (int j = 0; j < 3; j++) {
c[i][j] = a[i][j] + b[i][j];
System.out.print(c[i][j] + " ");
}
}
}
}

```

```

}
System.out.println();
}
}
}

```

## 6) Write a Java program to multiply two matrix?

```

public class MultiplyTwoMatrix {
    public static void main(String args[]) {
        //creating two matrices
        int a[][] = {{1, 1, 1}, {2, 2, 2}, {3, 3, 3}};
        int b[][] = {{1, 1, 1}, {2, 2, 2}, {3, 3, 3}};
        //creating another matrix to store the multiplication of two matrices
        int c[][] = new int[3][3];
        //multiplying and printing multiplication of 2 matrices
        for(int i = 0; i < 3; i++) {
            for(int j = 0; j < 3; j++) {
                c[i][j] = 0;
                for(int k = 0; k < 3; k++) {
                    c[i][j] += a[i][k] * b[k][j];
                }
                System.out.print(c[i][j] + " ");
            }
            System.out.println();
        }
    }
}

```

## 7) Write a Java program to get the transpose of matrix?

```

public class TransposeMatrix{
    public static void main(String args[]){
        //creating a matrix
        int original[][]={{1,3,4},{2,4,3},{3,4,5}};
        //creating another matrix to store transpose of a matrix
        int transpose[][]=new int[3][3]; //3 rows and 3 columns
        //Code to transpose a matrix
        for(int i=0;i<3;i++){
            for(int j=0;j<3;j++){
                transpose[i][j]=original[j][i];
            }
        }
    }
}

```

```

System.out.println("Printing Matrix without transpose:");
for(int i=0;i<3;i++){
for(int j=0;j<3;j++){
System.out.print(original[i][j]+" ");
}
System.out.println();
}
System.out.println("Printing Matrix After Transpose:");
for(int i=0;i<3;i++){
for(int j=0;j<3;j++){
System.out.print(transpose[i][j]+" ");
}
System.out.println();
}
}
}
}

```

## 8) Write a Java program to compare two strings?

```

public class CompareTwoStrings {
public static void main(String[] args){
Scanner sc = new Scanner(System.in);
System.out.println("Enter first string");
String first = sc.next();
System.out.println("Enter second string");
String second = sc.next();
compare(first,second);
}

public static void compare(String s1, String s2){
if(s1.compareTo(s2)==0) {
System.out.println("Strings are equal");
} else {
System.out.println("Strings are not equal");
}
}
}
}

```

## 9) How to find whether a String ends with a specific character or text using Java program?

```

public class StringEndWith{
public static void main(String args[]) {

```

```
String s1 = "Java is a programming language";
//Check if string ends with particular character
boolean endsWithCharacter = s1.endsWith("e");
System.out.println("String ends with character 'e': " + endsWithCharacter);
//Check if string ends with particular text
boolean endsWithText = s1.endsWith("java");
System.out.println("String ends with String 'lang': " + endsWithText);
}
}
```

## 10) Write a Java program to demonstrate indexOf()?

```
public class IndexOfExample{
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the input string:");
        String inputString = sc.nextLine();
        System.out.println("Enter the sub string:");
        String subString = sc.nextLine();
        int index = inputString.indexOf(subString);
        System.out.println("Index of sub string is: " + index);
    }
}
```

## 11) Write a Java program to demonstrate how to replace a string with another string?

```
public class ReplaceString{
    public static void main(String args[]) {
        String originalString = "Java for dummies";
        String newString = originalString.replace("dummies","experts");
        System.out.println("Original string is: " + originalString);
        System.out.println("New String is: " + newString);
    }
}
```

## 12) Write a Java program to split the given string?

```
class SplitString{
    public static void main(String []args){
        String strMain = "Java,C,Python,Perl";
        String[] arrSplit = strMain.split(",");
        for (int i=0; i < arrSplit.length; i++)
```

```

{
System.out.println(arrSplit[i]);
}
}
}
}

```

### 13) Write a Java program to remove the spaces before and after the given string?

```

class RemoveSpacesInString{
public static void main(String []args){
String s1 = "Interview Questions for Java";
String newString = s1.replaceAll("\\s","");
System.out.println("Old String: " + s1);
System.out.println("New String: " + newString);
}
}

```

### 14) Write a Java program to convert all the characters in given string to lower case?

```

class ConvertToLowerCase{
public static void main(String []args){
String s1 = "Interview QUESTIONS";
String newString = s1.toLowerCase();
System.out.println("Old String: " + s1);
System.out.println("New String: " + newString);
}
}

```

### 15) Write a Java program to demonstrate creating a method?

```

public class CreateMethodExample {
public static void main(String[] args){
Scanner sc = new Scanner(System.in);
System.out.println("Enter the number");
int num = sc.nextInt();
reverseNumber(num);
}

public static void reverseNumber(int number){
int reverse = 0;
while(number!=0){
int digit = number % 10;

```



```

reverse = reverse * 10 + digit;
number = number/10;
}
System.out.println("Reversed number " + reverse);
}
}

```

### 16) Write a Java program to find the length of the given string?

```

class FindLength{
public static void main(String []args){
String s1 = "Interview Questions In Java";
int length = s1.length();
System.out.println("Length of string is: " + length);
}
}

```

### 17) Write a Java program to concatenate the given strings?

```

class StringConcatenation{
public static void main(String []args){
String s1 = "Interview Questions In Java";
String s2 = " And Selenium";
String s3 = s1.concat(s2);
System.out.println("After concatenation: "+ s3);
}
}

```

### 18) Write a Java program to replace a string?

```

class ReplaceString{
public static void main(String []args){
String s1 = "Interview Questions In Java";
String s2 = "Answers";
String s3 = s1.replace("Questions","Answers");
System.out.println("Original String: "+ s1);
System.out.println("New String: "+ s3);
}
}

```

### 19) Write a Java program to demonstrate a Static block?

```

class StaticTest {
static int i;
int j;

```

```
// start of static block
static {
i = 10;
System.out.println("static block called ");
}
// end of static block
}

class Main {
public static void main(String args[]) {
System.out.println(Test.i);
}
}
```

## 20) Explain the difference between static and instance methods in Java?

Instance method are methods which require an object of its class to be created before it can be called. Static methods are the methods in Java that can be called without creating an object of class.

## 21) Write a Java program to demonstrate creating multiple classes?

```
public class A {
public static void print() {
System.out.println("This is a method");
}
}

public class B {
public static void main(String args[]) {
print();
}
}
```

## 22) Write a Java program to demonstrate creating a constructor?

```
class ConstructorTest {
ConstructorTest(){
System.out.println("Constructor called");
}
}

class Main
{
public static void main (String[] args)
```

```

{
ConstructorTest test = new ConstructorTest();
}
}

```

### 23) Write a Java program to demonstrate constructor overloading?

```

class Box
{
double width, height, depth;
int boxNo;
Box(double w, double h, double d, int num)
{
width = w;
height = h;
depth = d;
boxNo = num;
}
Box()
{
width = height = depth = 0;
}
Box(int num)
{
this();
boxNo = num;
}
public static void main(String[] args)
{
Box box1 = new Box(1);
System.out.println(box1.width);
}
}

```

### 24) Write a Java program to demonstrate Exception Handling?

```

public class ExceptionExample{
public static void main(String args[]){
try{
//code that may raise exception
int data=100/0;
}
}

```

```
catch(ArithmeticException e)
{System.out.println(e);}
}
}
```

## 25) Write a Java program to demonstrate throwing an exception?

```
class ThrowExceptionExample
{
public static void main(String[] args)throws InterruptedException
{
Thread.sleep(10000);
System.out.println("Hello World");
}
}
```

# Java for Testers – Interview Questions and Answers Part-3

## 1) Write a Java program to demonstrate the creation of Interface?

// A simple interface

```
interface A
{
final int a = 10;
void display();
}
```

// A class that implements the interface.

```
class TestClass implements A
{
public void display()
{
```

```

System.out.println("this is an interface");

}

public static void main (String[] args)

{

TestClass t = new TestClass();

t.display();

System.out.println(a);

}

}

```

## 2) Write a Java program to print date and time?

```

public class CurrentDateTime {

public static void main(String[] args) {

DateTimeFormatter df = DateTimeFormatter.ofPattern("yyyy/MM/dd HH:mm:ss");

LocalDateTime now = LocalDateTime.now();

System.out.println(df.format(now));

}

}

```

## 3) Write a Java program to demonstrate SQL Date?

```

public class SQLDate {

public static void main(String[] args) {

long millis=System.currentTimeMillis();

java.sql.Date date=new java.sql.Date(millis);

System.out.println(date);

}

}

```

#### 4) Write a Java program to demonstrate Date format?

```
public class GetCurrentDateTime {  
  
    private static final DateFormat sdf = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss");  
  
    private static final DateTimeFormatter dtf = DateTimeFormatter.ofPattern("yyyy/MM/dd HH:mm:ss");  
  
    public static void main(String[] args) {  
  
        Date date = new Date();  
  
        System.out.println(sdf.format(date));  
  
        Calendar cal = Calendar.getInstance();  
  
        System.out.println(sdf.format(cal.getTime()));  
  
        LocalDateTime now = LocalDateTime.now();  
  
        System.out.println(dtf.format(now));  
  
        LocalDate localDate = LocalDate.now();  
  
        System.out.println(DateTimeFormatter.ofPattern("yy/MM/dd").format(localDate));  
  
    }  
}
```

#### 5) Write a Java program to demonstrate generating a random number?

```
public class GenerateRandomNumber {  
  
    public static void main(String[] args){  
  
        Random rand = new Random();  
  
        int r1 = rand.nextInt(1000);  
  
        System.out.println("Random numbers: " + r1);  
  
    }  
}
```

#### 6) Write a Java program to demonstrate garbage collection?

```
public class GarbageCollector {  
  
    public static void main(String[] args) throws InterruptedException{
```

```

GarbageCollector t1 = new GarbageCollector();

GarbageCollector t2 = new GarbageCollector();

// Nullifying the reference variable

t1 = null;

// requesting JVM for running Garbage Collector

System.gc();

// Nullifying the reference variable

t2 = null;

// Requesting JVM for running Garbage Collector

Runtime.getRuntime().gc();

}

@Override

protected void finalize() throws Throwable {

System.out.println("Garbage collector called");

System.out.println("Object garbage collected: " + this);

}

}

```

## 7) Write a Java program to get the IP Address of own machine?

```

public class FindIPAddress {

public static void main(String[] args) throws Exception{

// Returns the instance of InetAddress containing local host name and address

InetAddress localhost = InetAddress.getLocalHost();

System.out.println("System IP Address : " + (localhost.getHostAddress()).trim());

// Find public IP address

String systemipaddress = "";

```

```

try {

URL url_name = new URL("http://bot.whatismyipaddress.com");

BufferedReader sc = new BufferedReader(new InputStreamReader(url_name.openStream()));

// reads system IP Address

systemipaddress = sc.readLine().trim();

} catch (Exception e) {

systemipaddress = "Cannot Execute Properly";

}

System.out.println("Public IP Address: " + systemipaddress + "\n");

}

}

```

## 8) Write a Java program to open a notepad?

```

public class OpenNotepad {

public static void main(String[] args){

Runtime rs = Runtime.getRuntime();

try{

rs.exec("notepad");

}

catch (IOException e){

System.out.println(e);

}

}

}

```

## 9) Write a Java program to demonstrate Linear Search?

```

public class LinearSearch {

```



```

static int search(int arr[], int n, int x){

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

        if(arr[i] == x)

            return i;

    }

    return -1;

}

public static void main(String[] args){

    int[] arr = {5,1,9,4,3,8};

    int n = arr.length;

    int x = 4;

    int index = search(arr,n,x);

    if(index == -1){

        System.out.println("Element is not present in the array");

    }

    else

        System.out.println("Element found at position " + index);

}

}

```

## 10) Write a Java program to demonstrate Binary Search?

```

class BinarySearch {

    int binarySearch(int arr[], int l, int r, int x)

    {

        if (r >= l) {

            int mid = l + (r - l) / 2;

```

```

// If the element is present at the middle itself
if (arr[mid] == x)

return mid;

// If element is smaller than mid, then it can only be present in left subarray
if (arr[mid] > x)

return binarySearch(arr, l, mid - 1, x);

// Else the element can only be present in right subarray
return binarySearch(arr, mid + 1, r, x);

}

// If element is not present in array
return -1;

}

public static void main(String args[])

{

BinarySearch bs = new BinarySearch();

int arr[] = { 2, 3, 4, 10, 40 };

int n = arr.length;

int x = 10;

int result = bs.binarySearch(arr, 0, n - 1, x);

if (result == -1)

System.out.println("Element not present");

else

System.out.println("Element found at index " + result);

}

}

```

## 11) Write a Java program to demonstrate Bubble sort?

```
class BubbleSort
```

```
{
```

```
void bubbleSort(int arr[]) {
```

```
int n = arr.length;
```

```
for (int i = 0; i < n-1; i++)
```

```
for (int j = 0; j < n-i-1; j++)
```

```
if (arr[j] > arr[j+1])
```

```
{
```

```
int temp = arr[j];
```

```
arr[j] = arr[j+1];
```

```
arr[j+1] = temp;
```

```
}
```

```
}
```

```
void printArray(int arr[]) {
```

```
int n = arr.length;
```

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

```
System.out.print(arr[i] + " ");
```

```
System.out.println();
```

```
}
```

```
public static void main(String args[]) {
```

```
BubbleSort bs = new BubbleSort();
```

```
int arr[] = {64, 34, 25, 12, 22, 11, 90};
```

```
bs.bubbleSort(arr);
```

```
System.out.println("Sorted array");
```

```
bs.printArray(arr);
```

```
}
```

```
}
```

## 12) Write a Java program to demonstrate connecting to a Database?

```
public class ConnectMSSQLServer
```

```
{
```

```
public void dbConnect(String db_connect_string,String db_userid,String db_password)
```

```
{
```

```
try {
```

```
Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
```

```
Connection conn = DriverManager.getConnection(db_connect_string,db_userid, db_password);
```

```
System.out.println("connected");
```

```
Statement statement = conn.createStatement();
```

```
String queryString = "select * from Employees where lastName='Smith'";
```

```
ResultSet rs = statement.executeQuery(queryString);
```

```
while (rs.next()) {
```

```
System.out.println(rs.getString(1));
```

```
}
```

```
} catch (Exception e) {
```

```
e.printStackTrace();
```

```
}
```

```
}
```

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

```
{
```

```
ConnectMSSQLServer connServer = new ConnectMSSQLServer();
```

```

connServer.dbConnect("jdbc:sqlserver://<hostname>", "<user>", "<password>");
}

}

```

### 13) Write a Java program to demonstrate inserting data into a table using JDBC?

```

public class InsertIntoDB {

    public static void main(String[] args) {

        Connection connection = null;

        Statement stmt = null;

        try

        {

            Class.forName("com.mysql.jdbc.Driver");

            connection = DriverManager

            .getConnection("jdbc:mysql://localhost:3306/JDBCdemo", "root", "password");

            stmt = connection.createStatement();

            stmt.execute("INSERT INTO EMPLOYEE (ID,FIRST_NAME, LAST_NAME, STAT_CD) "

            + "VALUES (1,'John','Smith',5)");

        }

        catch (Exception e) {

            e.printStackTrace();

        } finally {

            try {

                stmt.close();

                connection.close();

            } catch (Exception e) {

```

```

e.printStackTrace();

}

}

}

}

```

## 14) Write a Java program to demonstrate executing a Procedure in JDBC?

```

public class StoredProcedureCallExample1 {

    public static void main(String[] args) {

        String dbURL = "jdbc:mysql://localhost:3306/booksdb";

        String user = "root";

        String password = "P@ssw0rd";

        try (

            Connection conn = DriverManager.getConnection(dbURL, user, password);

            CallableStatement statement = conn.prepareCall("{call create_author(?, ?)}");

        ) {

            statement.setString(1, "Bill Gates");

            statement.setString(2, "bill@microsoft.com");

            statement.execute();

            statement.close();

            System.out.println("Stored procedure called successfully!");

        } catch (SQLException ex) {

            ex.printStackTrace();

        }

    }

}

```

## 15) Write a Java program to check Regular Expressions?

```
class CheckRegularExpression{

    public static void main(String []args){

        Pattern pattern = Pattern.compile("auto*");

        Matcher m = pattern.matcher("automation");

        while (m.find())

            System.out.println("Pattern found from " + m.start() + " to " + (m.end()-1));

    }

}
```

## 16) Write a Java program to create Multi-threading?

```
class Multithreading extends Thread

{

    public void run()

    {

        try

        {

            // Displaying the thread that is running

            System.out.println ("Thread " + Thread.currentThread().getId() + " is running");

        }

        catch (Exception e)

        {

            System.out.println ("Exception is caught");

        }

    }

}
```

```

public class Multithread
{
    public static void main(String[] args)
    {
        int n = 8; // Number of threads

        for (int i=0; i<8; i++)
        {
            Multithreading object = new Multithreading();

            object.start();
        }
    }
}

```

## 17) Write a Java program to demonstrate joining thread?

```

public class JoinThreads {
    public static void main(String args[]) throws InterruptedException{
        System.out.println(Thread.currentThread().getName() + " is Started");

        Thread exampleThread = new Thread(){
            public void run(){
                try {
                    System.out.println(Thread.currentThread().getName() + " is Started");

                    Thread.sleep(2000);

                    System.out.println(Thread.currentThread().getName() + " is Completed");
                } catch (InterruptedException ex) {
                    Logger.getLogger(Join.class.getName()).log(Level.SEVERE, null, ex);
                }
            }
        };
    }
}

```



```

}

}

exampleThread.start();

exampleThread.join();

System.out.println(Thread.currentThread().getName() + " is Completed");

}

}

```

### 18) Write a Java program to write data into the text files?

```

public static void WriteTextFile() throws IOException
{

String fileContent = "This is Java interview questions";

BufferedWriter writer = new BufferedWriter(new FileWriter("c:/temp/samplefile.txt"));

writer.write(fileContent);

writer.close();

}

```

### 19) Write a Java program to read data from the text files?

```

public class ReadTextAsString
{

public static String readFileAsString(String fileName) throws Exception
{

String data = "";

data = new String(Files.readAllBytes(Paths.get(fileName)));

return data;

}

public static void main(String[] args) throws Exception

```

```
{  
  
String data = readFileAsString("C:\\temp\\test.java");  
  
System.out.println(data);  
  
}  
  
}
```

## 20) Write a Java program to convert string to integer?

```
public class ConvertStringToInteger{  
  
    public static void main(String[] args){  
  
        String number = "10";  
  
        int result = Integer.parseInt(number);  
  
        System.out.println(result);  
  
    }  
  
}
```

## 21) Write a Java program to convert integer to string?

```
public class ConvertIntegerToString{  
  
    public static void main(String[] args){  
  
        int a = 100  
  
        String str = Integer.toString(a);  
  
        System.out.println(str);  
  
    }  
  
}
```

## 22) Write a Java program to convert string to long?

```
public class ConvertStringToLong{  
  
    public static void main(String[] args){  
  
        String s = "10";  
  
        long result = Long.parseLong(s);
```

```
System.out.println(result);  
  
}  
  
}
```

### 23) Write a Java program to convert string to float?

```
public class ConvertStringToLong{  
  
    public static void main(String[] args){  
  
        String s = "13.6";  
  
        float result = Float.parseFloat(s);  
  
        System.out.println(result);  
  
    }  
  
}
```

### 24) Write a Java program to convert string to double?

```
public class ConvertStringToDouble{  
  
    public static void main(String[] args){  
  
        String s = "6.54";  
  
        double result = Double.parseDouble(s);  
  
        System.out.println(result);  
  
    }  
  
}
```

### 25) Write a Java program to convert string to date?

```
public class StringToDate {  
  
    public static void main(String[] args) throws Exception {  
  
        String sDate="01/01/2020";  
  
        Date date=new SimpleDateFormat("dd/MM/yyyy").parse(sDate);  
  
        System.out.println(sDate+"\t"+date);  
  
    }  
  
}
```

```
}
```

# Java for Testers – Interview Questions and Answers Part-4

## 1) What is the use of static keyword in Main()?

Since main() method is the entry point of any Java application, hence making the main() method as static is mandatory for launching the Java application. Since the main method is static, Java virtual Machine can call it without creating any instance of a class which contains the main method.

## 2) Can a class without main() gets compilation successful?

The main method is only used when the Java Virtual Machine is executing your code. Code cannot be executed without a main method but it can still be compiled.

## 3) Can we create private access specifier inside interface?

We cannot declare class/interface with private or protected access specifiers.

## 4) Is there any way to deallocate memory in JAVA?

We should dereference the object, then memory will be deallocated automatically during garbage collection

## 5) Write a program for removing white spaces in a String?

```
class RemoveWhiteSpaces{  
  
    public static void main(String[] args){  
  
        String str = " Java Interview Questions ";  
  
        str = str.replaceAll("\\s", "");  
  
        System.out.println(str);  
  
    }  
}
```

```
}
```

## 6) What is Object class?

The Object class is the parent class of all the classes in java by default. Every class in Java is directly or indirectly derived from the Object class.

## 7) Write a Java program for pascle triangle?

```
class PascalTriangle {  
  
    static void printPascal(int n)  
  
    {  
  
        for (int line = 0; line < n; line++)  
  
        {  
  
            for (int i = 0; i <= line; i++)  
  
                System.out.print(pattern(line, i)+" ");  
  
                System.out.println();  
  
        }  
  
    }  
  
    static int pattern(int n, int k)  
  
    {  
  
        int res = 1;  
  
        if (k > n - k)  
  
            k = n - k;  
  
        for (int i = 0; i < k; ++i)  
  
        {  
  
            res *= (n - i);  
  

```

```

    res /= (i + 1);
}

return res;
}

public static void main(String args[])
{
    int n = 7;

    printPascal(n);
}
}

```

### 8) What are the differences between interface and inheritance?

- Inheritance is an OOP concept to derive new classes from the existing classes. Interface is a mechanism in OOP to implement abstraction and multiple inheritance.
- Inheritance provides code reusability. Interfaces provide abstraction and multiple inheritance.

### 9) If we close the driver in try block then, FINALLY will execute or not?

Java finally block is always executed whether exception is handled or not.

### 10) What is difference between method overloading and constructor overloading with example?

If we want to have different ways of initializing an object using different number of parameters, then we must do constructor overloading and we do method overloading when we want different definitions of a method based on different parameters.

### 11) What is the difference between normal class and final class?

Final class cannot be inherited by any other class but normal class can be inherited by other classes

## 12) Adapter design in java?

Adapter design pattern is one of the structural design pattern and its used so that two unrelated interfaces can work together. The object that joins these unrelated interface is called an Adapter.

## 13) Write a Java program for sorting of numbers?

```
public class SortNumbers
{
    public static void main(String[] args)
    {
        int[] arr = {13, 7, 6, 45, 21, 9, 101, 102};
        Arrays.sort(arr);
        System.out.printf("Modified arr[] : %s", Arrays.toString(arr));
    }
}
```

## 14) Write a Java program for searching a letter in string?

```
public class SearchLetter {
    public static void main(String args[]) {
        String str = "Automation Questions"
        int index = str.indexOf('Q');
        System.out.println("Index of the letter Q :: "+index);
    }
}
```

### 15) Write a Java program for sorting an array?

```
public class SortArray
{
    public static void main(String[] args)
    {
        int[] arr = {13, 7, 6, 45, 21, 9, 101, 102};
        Arrays.sort(arr);
        System.out.printf("Modified arr[] : %s", Arrays.toString(arr));
    }
}
```

### 16) What is a good approach to throw an exception?

By using throws keyword.

```
public class SortArray
{
    public static void main(String[] args)
    {
        int[] arr = {13, 7, 6, 45, 21, 9, 101, 102};
        Arrays.sort(arr);
        System.out.printf("Modified arr[] : %s", Arrays.toString(arr));
    }
}
```



### 17) Find how many duplicate values in Array List?

```
public static Set<Integer> findDuplicates(int[] input) {  
  
    Set<Integer> duplicates = new HashSet<Integer>();  
  
    for (int i = 0; i < input.length; i++) {  
  
        if(duplicates.add(input[i])!=false){  
  
            System.out.println("Found duplicate element in array")  
  
        }  
  
    }  
  
}
```

### 18) String [] str={"abc","efg","fgh"}; convert array to string?

```
Arrays.toString (str);
```

### 19) Explain the features of JAVA?

Most important features are:

- Simple
- Object-Oriented
- Portable
- Platform independent
- Secured
- Robust
- Architecture neutral
- Interpreted
- High Performance
- Multithreaded
- Distributed
- Dynamic

### 20) How do u say JAVA is platform independent?

Platform independent language means once compiled you can execute the program on any platform (OS). Java is platform independent. Because the Java compiler converts the source code to bytecode, which is Intermediate Language. Bytecode can be executed on any platform (OS) using JVM (Java Virtual Machine).

### **21) Is JVM platform independent?**

Yes. Java is platform independent.

### **22) What is the difference between a Class and Object?**

- Object is an instance of a class. Class is a blueprint or template from which objects are created.
- Object is a real world identity. Class is a group of similar objects
- Object allocates memory when it is created. Class doesn't allocated memory when it is created.

### **23) Write a Java program to check whether an year is leap year or not?**

```
public class LeapYear {  
  
    public static void main(String[] args){  
  
        Scanner sc = new Scanner(System.in);  
  
        System.out.println("Enter a year");  
  
        int year = sc.nextInt();  
  
        System.out.println(isLeapYear(year)? "Leap Year": "Not a Leap Year");  
  
    }  
  
    private static boolean isLeapYear(int num){  
  
        if(num%400 == 0)  
  
            return true;  
  
        if(num%100 == 0)  
  
            return true;
```

```
if(num%4 == 0)

return true;

return false;

}

}
```

#### **24) Where to use Hashtable in real time?**

Hashtable should be used in multithreading applications as it is thread safe and synchronized.

#### **25) What is the difference between Volatile & Transient in Java?**

- transient keyword is used along with instance variables to exclude them from serialization process
- if a field is transient its value will not be persisted. On the other hand volatile keyword can also be used in variables to indicate compiler and JVM that always read its value from main memory
- transient keyword can not be used along with static keyword but volatile can be used along with static.
- transient variables are initialized with default value during de-serialization and there assignment or restoration of value has to be handled by application code.

## **Java for Testers – Interview Questions and Answers Part-5**

### **1) What is the difference between List and Set?**

- List in Java allows duplicates while Set doesn't allow duplicates.
- List is an Ordered Collection while Set is an unordered Collection. List maintains insertion order of elements, means any element which is inserted before will go on lower index than any element which is inserted after. Set in Java doesn't maintain any order.
- Set uses equals() method to check uniqueness of elements stored in Set, while SortedSet uses compareTo() method to implement natural sorting order of elements.
- Popular implementation of List interface in Java includes ArrayList, Vector and LinkedList. While popular implementation of Set interface includes HashSet, TreeSet and LinkedHashSet.

### **2) What is the purpose of using Constructors in Java?**

Constructors are used to assign values to the class variables at the time of object creation, either explicitly done by the programmer or by Java itself (default constructor).

### **3) How Constructors are different from Methods in Java?**

- Name of the constructor must be same with the name of the Class but there is no such requirement for a method in Java. Methods can have any arbitrary name in Java.
- Constructor doesn't have any return type but the method has the return type and returns something unless its void.
- Constructors are chained and they are called in a particular order, there is no such facility for methods.
- Constructors are not inherited by child classes but methods are inherited by child classes until they are made private. in which case they are only visible in class on which they are declared.

### **4) What is the purpose of using 'this' keyword in Java?**

Below are the different usage of this keyword:

- this can be used to refer current class instance variable.
- this can be used to invoke current class method (implicitly)

- this() can be used to invoke current class constructor.
- this can be passed as an argument in the method call.
- this can be passed as argument in the constructor call.
- this can be used to return the current class instance from the method.

## 5) What is Overloading in Java?

Overloading allows different methods to have the same name, but different signatures where the signature can differ by the number of input parameters or type of input parameters or both. Overloading is related to compile-time (or static) polymorphism.

## 6) What is the purpose of using Packages in Java?

Package are used in Java, in-order to avoid name conflicts and to control access of class, interface and enumeration etc.

## 7) What is the keyword used by a Java class to inherit the properties say variables and methods of another Class?

The keyword used for inheritance is extends.

```
class derived-class extends base-class
```

```
{
```

```
//methods and fields
```

```
}
```

## 8) How to access the variables and methods of another Class in Java?

By creating an instance of another class.

```
class A
```

```
{
```

```
int i = 0,j = 1;
```

```

}

class B

{

//Accessing variables of class A by creating an instance of the class A

A a1 = new A():

int a = a1.i;

int b = a1.j;

}

```

## 9) What is Overriding in Java?

Overriding is a feature that allows a subclass or child class to provide a specific implementation of a method that is already provided by one of its super-classes or parent classes. When a method in a subclass has the same name, same parameters or signature and same return type(or sub-type) as a method in its super-class, then the method in the subclass is said to override the method in the super-class.

## 10) Is Overriding applicable for Constructors in Java?

Constructors cannot be overridden in java.

## 11) What are the different modifiers in Java?

- Default: The data members, class or methods which are not declared using any access modifiers i.e. having default access modifier are accessible only within the same package.
- Private: The methods or data members declared as private are accessible only within the class in which they are declared.
- Protected: The methods or data members declared as protected are accessible within same package or sub classes in different package.
- Public: Classes, methods or data members which are declared as public

are accessible from every where in the program. There is no restriction on the scope of a public data members.

### **12) What is the difference between default and protected access modifiers in Java?**

A Default member may be accessed only if the Class accessing the member belongs to the same Package only where as a Protected member can be accessed (through Inheritance) by a SubClass even if the SubClass is in a different Package.

### **13) What is the difference between static and instance variable in Java?**

In case of static variable, each and every instance of the class will share the same variable, so that if you change it in one instance, the change will reflect in all instances, created either before or after the change.

Instance variables belong to the instance of a class, thus an object. And every instance of that class (object) has it's own copy of that variable. Changes made to the variable don't reflect in other instances of that class.

### **14) What is the difference between static and non-static methods in Java?**

- A static method can access only static members and can not access non-static members but a non-static method can access both static as well as non-static members.
- Static method uses compile time binding or early binding but Non-static method uses run time binding or dynamic binding.
- A static method cannot be overridden being compile time binding but a non-static method can be overridden being dynamic binding.
- Static method occupies less space and memory allocation happens once but a non-static method may occupy more space.
- A static method is declared using static keyword but a normal method is not required to have any special keyword.

### **15) What happens when we specify the final non-access modifier with variables and methods in Java?**

A final variable can be explicitly initialized only once. A reference variable declared final can never be reassigned to refer to a different object.

A final method cannot be overridden by any subclasses. The final modifier prevents a method from being modified in a subclass.

## **16) What is the difference between abstract classes and interfaces in Java?**

- Type of methods: Interface can have only abstract methods. Abstract class can have abstract and non-abstract methods.
- Final Variables: Variables declared in a Java interface are by default final. An abstract class may contain non-final variables.
- Type of variables: Abstract class can have final, non-final, static and non-static variables. Interface has only static and final variables.
- Implementation: Abstract class can provide the implementation of interface. Interface can't provide the implementation of abstract class.
- Inheritance vs Abstraction: A Java interface can be implemented using keyword "implements" and abstract class can be extended using keyword "extends".
- Multiple implementation: An interface can extend another Java interface only, an abstract class can extend another Java class and implement multiple Java interfaces.
- Accessibility of Data Members: Members of a Java interface are public by default. A Java abstract class can have class members like private, protected, etc.

## **17) What is the keyword used for inheriting the interfaces in Java?**

Extends keyword is used for inheriting the interfaces.

```
class A
```

```
{
```

```
int i = 0, j = 1;
```

```
}
```

```
class B
```

```
{
```



```
//Accessing variables of class A by creating an instance of the class A
```

```
A a1 = new A():
```

```
int a = a1.i;
```

```
int b = a1.j;
```

```
}
```

## 18) How to handle exceptions in Java?

There are 5 keywords which are used in handling exceptions in Java – try, catch, finally, throw and throws.

```
class A
```

```
{
```

```
int i = 0,j = 1;
```

```
}
```

```
class B
```

```
{
```

```
//Accessing variables of class A by creating an instance of the class A
```

```
A a1 = new A():
```

```
int a = a1.i;
```

```
int b = a1.j;
```

```
}
```

## 19) What is the difference between checked and unchecked exceptions in Java?

The classes which directly inherit Throwable class except RuntimeException and Error are known as checked exceptions e.g. IOException, SQLException etc. Checked exceptions are checked at compile-time.

The classes which inherit RuntimeException are known as unchecked exceptions e.g. ArithmeticException, NullPointerException, ArrayIndexOutOfBoundsException etc. Unchecked exceptions are not checked at compile-time, but they are checked at runtime.

## **20) What is the disadvantage of Arrays and how to overcome it in Java?**

Arrays are of fixed size and cannot be increased or decreased once declared. This can be overcome by using ArrayList.

## **21) What is the difference between ArrayList and HashSet in Java ?**

- Implementation : ArrayList implements List interface while HashSet implements Set interface in Java.
- Internal implementation: ArrayList is backed by an Array while HashSet is backed by an HashMap.
- Duplicates : ArrayList allows duplicate values while HashSet doesn't allow duplicates values.
- Ordering : ArrayList maintains the order of the object in which they are inserted while HashSet is an unordered collection and doesn't maintain any order.
- Indexing : ArrayList is index based we can retrieve object by calling get(index) method or remove objects by calling remove(index) method while HashSet is completely object based. HashSet also does not provide get() method.
- Null Object: ArrayList does not apply any restriction, we can add any number of null value while HashSet allow one null value.

## **22) What are the predefined methods of HashMap, which are used for adding the value and retrieving the value in Java?**

- get – It is used to retrieve or fetch the value mapped by a particular key.
- put – It is used to insert a particular mapping of key-value pair into a map.

**23) What will this Java code print: String x = "Latest version"; String y = "of Selenium"; int z = 3; System.out.println("We are learning Selenium"+" and the "+x+" "+y+" is "+z); ?**

Output – We are learning Selenium and the Latest version of Selenium is 3

**24) Is Java case sensitive?**

Yes, it is case-sensitive.

**25) Which keyword is used for defining / creating a Class in Java?**

Class keyword is used for creating a class in Java.

## Java for Testers – Interview Questions and Answers Part-6

**1) How many ways we can define a String in Java?**

– Using String literal

```
String str = "Automation";
```

– Using new keyword

```
String str1 = new String("Automation");
```

– Using Character Array

```
char ch[] = {'A','U','T','O','M','A','T','I','O','N'}
```

```
String str2 = new String(ch);
```

**2) What is the difference between super and this keyword in Java?**

The main difference between this and super in Java is that this is used in context of the class you are working on, while super is used to refer current instance of parent class.

### 3) What is the difference between final and finally keywords?

The final keyword makes a Java variable, method or class as final and it cannot be changed once it is initialized.

Java finally block is part of try-catch-finally blocks for exception handling. A finally block is guaranteed to be executed despite whether an exception is thrown or not from the try block.

### 4) Can you access the private method from outside the class?

No. Private method cannot be accessed outside the class.

### 5) How to convert Array to ArrayList and ArrayList to Array?

#### Array to ArrayList:

```
String[] array = {"Java","Python"};
```

```
List list = Arrays.asList(array);
```

#### Arraylist to Array:

```
ArrayList<String> list = new ArrayList<>();
```

```
list.add("A");
```

```
list.add("B");
```

```
list.add("C");
```

```
list.add("D");
```

```
//Convert to object array
```

```
Object[] array = list.toArray();
```

## **6) Why static needs to be specified before variables and classes? What actually is its real time purpose/advantage?**

Static methods can be utilized without having to instantiate the class they belong to. We declare methods static so they can be accessed without holding an instance of an Object based on that class.

## **7) Why we have to use the synchronized block in Java?**

We synchronize a block of code to avoid any modification in state of the Object and to ensure that other threads can execute rest of the lines within the same method without any interruption.

## **8) What is the difference between HashSet and TreeSet?**

- HashSet gives better performance (faster) than TreeSet for the operations like add, remove, contains, size etc. HashSet offers constant time cost while TreeSet offers  $\log(n)$  time cost for such operations.
- HashSet does not maintain any order of elements while TreeSet elements are sorted in ascending order by default.

## **9) What is the difference between public and private access modifier?**

- Public methods are accessible from anywhere but private methods are only accessible inside the same class where it is declared.
- Public keyword provides lowest level of Encapsulation and Private modifier provides highest level of Encapsulation in Java.

## **10) What is the difference between String array and Char array?**

- String refers to a sequence of characters represented as a single data type. Character Array is a sequential collection of data type char.
- Strings are immutable. Character Arrays are mutable.
- Strings can be stored in any any manner in the memory. Elements in Character Array are stored contiguously in increasing memory locations.
- All Strings are stored in the String Constant Pool. All Character Arrays are stored in the Heap.

## **11) Is null a keyword in Java?**

No. null is not a keyword in Java.

### **12) What are class level variables in Java?**

- These variables are created when an object of the class is created and destroyed when the object is destroyed.
- Initialisation of Instance Variable is not Mandatory. Its default value is 0
- Instance Variable can be accessed only by creating objects.

### **13) What is the default value of a boolean variable in Java?**

Default value of boolean variable is false.

### **14) What is the default value of an integer variable in Java?**

Default value of integer variable is 0.

### **15) What is the default value of an object in Java?**

Default value of an object is null.

### **16) What is final?**

The final keyword makes a Java variable, method or class as final and it cannot be changed once it is initialized.

### **17) How to read data from the XML files?**

```
File xmlFile = new File("/Users/mkyong/staff.xml");
```

```
DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
```

```
DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
```

```
Document doc = dBuilder.parse(xmlFile);
```

```
doc.getDocumentElement().normalize();
```

```
System.out.println("Root element : " + doc.getDocumentElement().getNodeName());
```

### **18) What is JDBC and how it can be used to connect to Database?**

JDBC stands for Java Database Connectivity. JDBC is a Java API to connect and execute the query with the database.

## Connect to Oracle DB:

```
Class.forName("oracle.jdbc.driver.OracleDriver");

DriverManager.registerDriver(new oracle.jdbc.driver.OracleDriver());

Connection con = DriverManager.getConnection(url,user,password);

Statement st = con.createStatement();

int m = st.executeUpdate(sql);
```

## 19) Class.forName(X). What will you write in place of X?

X represents the Driver class file for the Database.

## 20) If we write FileInputStream statements in try block, what are the statements we will write in finally block?

```
if (fileInputStream != null) {

fileInputStream.close();

}
```

## 21) Explain about typecasting?

The process of converting the value of one data type (int, float, double, etc.) to another data type is known as typecasting.

Two types of typecasting:

- Widening or Automatic

Widening conversion takes place when two data types are automatically converted. This happens when:

The two data types are compatible.

When we assign value of a smaller data type to a bigger data type.

Byte -> Short -> Int -> Float -> Double

- Narrowing or Explicit

If we want to assign a value of larger data type to a smaller data type we perform explicit type casting or narrowing.

This is useful for incompatible data types where automatic conversion cannot be done.

Here, target-type specifies the desired type to convert the specified value to.

Double -> Float -> Long -> Int -> Short -> Byte

## **22) Explain stack and heap in Java for memory management?**

Stack Memory in Java is used for static memory allocation and the execution of a thread. It contains primitive values that are specific to a method and references to objects that are in a heap, referred from the method.

Heap space in Java is used for dynamic memory allocation for Java objects and JRE classes at the runtime. New objects are always created in heap space and the references to this objects are stored in stack memory.

## **23) Explain SingletonDesign pattern in Java?**

Singleton pattern restricts the instantiation of a class and ensures that only one instance of the class exists in the java virtual machine.

## **24) How to remove Junk or special characters in a String?**

We can use regular expression and `replaceAll()` method of `java.lang.String` class to remove all special characters from String.

## **25) What is Reflection and Singleton in Java?**

Singleton pattern restricts the instantiation of a class and ensures that only one instance of the class exists in the java virtual machine. Reflection can be caused to destroy singleton property of singleton class.

# **Java for Testers – Interview Questions and Answers Part-7**

## **1. What is threading?**

- A thread is a lightweight subprocess, the smallest unit of processing.
- It is a separate path of execution.
- Threads are independent.



- If there occurs exception in one thread, it doesn't affect other threads.
- It uses a shared memory area.

## **2. How does multi-threading is achieved?**

- Multithreading can be achieved by executing two or more threads simultaneously to maximum utilization of CPU.
- Multithreaded applications execute two or more threads run concurrently.
- Each thread runs parallel to each other.

## **3. How to initiate a thread in Java?**

Thread can be initiated by the following ways:

- Extending the Thread class
- Implementing the Runnable Interface

## **4. What do you mean by thread safe?**

Thread-safe code is code that will work even if many Threads are executing it simultaneously.

A piece of code is thread-safe if it only manipulates shared data structures in a manner that guarantees safe execution by multiple threads at the same time.

## **5. What is the difference between collection and collections?**

The Collection is an interface whereas Collections is a class. The Collection interface provides the standard functionality of data structure to List, Set, and Queue. However, Collections class is to sort and synchronize the collection elements.

## **6. What is a Collection and what are the types of collections?**

A Collection is a group of individual objects represented as a single unit. Java provides Collection Framework which defines several classes and interfaces to represent a group of objects as a single unit.

Different types of Collection are:

- Set
- List

- Map
- Queue

### **7. What is the difference between Array and ArrayList?**

- Array is static in size but ArrayList is dynamic in size.
- It is mandatory to provide the size of array during initialization while arraylist can be created without declaring its size.
- Array is faster than Arraylist.
- Array can be multidimensional but Arraylist is always single-dimensional.

### **8. What is the difference between Set and HashSet?**

- Set is an interface, HashSet – implementation of interface.
- The Set interface represents a set of some objects, non-ordered, without random-element access. HashSet – implementation of the Set interface, based on the .hashCode() function.

### **9. What is the difference between HashMap and HashTable?**

- HashMap is non synchronized and not thread safe but HashTable is synchronized and thread safe
- HashMap allows one null key and multiple null values but HashTable doesn't allow any null key or value
- HashMap is faster than HashTable

### **10. What is the difference between ArrayList and LinkedList?**

- Insertions are easy and fast in LinkedList as compared to ArrayList.
- An ArrayList class can act as a list only because it implements List only. LinkedList class can act as a list and queue both because it implements List and Deque interfaces.
- ArrayList internally uses a dynamic array to store the elements. LinkedList internally uses a doubly linked list to store the elements.

### **11. Write a Java program to calculate the power of a number using a while loop?**

```
public class Power {
```

```

public static void main(String[] args) {

    int base = 3, exponent = 4;

    long result = 1;

    while (exponent != 0)

    {

        result *= base;

        --exponent;

    }

    System.out.println("Answer = " + result);

}

}

```

## 12. Can we have duplicate key value in HashMap?

HashMap doesn't allow duplicate keys but allows duplicate values.

## 13. How to fetch values from a HashMap?

The java.util.HashMap.get() method of HashMap class is used to retrieve or fetch the value mapped by a particular key mentioned in the parameter.

Hash\_Map.get(Object key\_element)

## 14. Write a Java program to verify whether a number is perfect number or not?

```

class PerfectNumber

{

    static boolean isPerfect(int n)

    {

        int sum = 1;

        for (int i = 2; i * i <= n; i++)

```

```

{
    if (n % i == 0)
    {
        if (i * i != n)
            sum = sum + i + n / i;
        else
            sum = sum + i;
    }
}

if (sum == n && n != 1)
    return true;
    return false;
}

public static void main (String[] args)
{
    System.out.println("Below are all perfect" +
        "numbers till 10000");
    for (int n = 2; n < 10000; n++)
        if (isPerfect(n))
            System.out.println(n +
                " is a perfect number");
    }
}

```

## 15. Write a Java program for printing the Fibonacci series from 1 to 10?

```
class FibonacciNumber
```

```
{
```

```
// Method to print first n Fibonacci Numbers
```

```
static void printFibonacciNumbers(int n)
```

```
{
```

```
int f1 = 0, f2 = 1, i;
```

```
if (n < 1)
```

```
return;
```

```
for (i = 1; i <= n; i++)
```

```
{
```

```
System.out.print(f2+" ");
```

```
int next = f1 + f2;
```

```
f1 = f2;
```

```
f2 = next;
```

```
}
```

```
}
```

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

```
{
```

```
printFibonacciNumbers(10);
```

```
}
```

```
}
```

## 16. Write a Java program to find the greatest of three numbers?

```
public class GreatestNumber {
```

```

public static void main(String args[]) {

    int num1 = 10;

    int num2 = 21;

    int num3 = 7;

    if (num1 >= num2 && num1 >= num3)

        System.out.println( num1 + " is the maximum number.");

    else if (num2 >= num1 && num2 >= num3)

        System.out.println( num2 + " is the maximum number.");

    else

        System.out.println( num3 + " is the maximum number.");

}

}

```

## 17. Find and print the largest two numbers from an array of given numbers?

```

public class LargestTwoNumbers

{

    public static void main (String[] args)

    {

        Scanner scn = new Scanner (System.in);

        System.out.print("Enter no. of elements you want in array.");

        int n = scn.nextInt();

        int array[] = new int[n];

        System.out.println("Enter all the elements:");

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

```

```
{  
  
array[i] = scn.nextInt();  
  
}  
  
int largest1, largest2, temp;  
  
largest1 = array[0];  
largest2 = array[1];  
  
if (largest1 < largest2)  
{  
  
temp = largest1;  
  
largest1 = largest2;  
  
largest2 = temp;  
  
}  
  
for (int i = 2; i < array.length; i++)  
{  
  
if (array[i] > largest1)  
{  
  
largest2 = largest1;  
  
largest1 = array[i];  
  
}  
  
else if (array[i] > largest2 && array[i] != largest1)  
{  
  
largest2 = array[i];  
  
}  
  
}  
  
System.out.println ("The First largest is " + largest1);
```

```
System.out.println ("The Second largest is " + largest2);  
  
}  
  
}
```

## 18. In the given String, remove the white spaces, reverse it and print only the odd position characters?

```
public Class PrintOddChars{  
  
    public static void main(String[] args) {  
  
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));  
  
        System.out.println("Enter the string: ");  
  
        String name = br.readLine();  
  
        String odd = "";  
  
        String reversedString = reverseString(name.replaceAll("\\s", ""));  
  
        for(int i=0;i<reversedString.length();i++){  
  
            if(i%2!=0){  
  
                odd = odd + reversedString.charAt(i);  
  
            }  
  
        }  
  
        System.out.println(odd);  
  
    }  
  
    public static String reverseString(String s){  
  
        String rev="";  
  
        char[] arr = s.toCharArray();  
  
        for(int i=arr.length-1;i>=0;i--)  
  
            rev = rev + arr[i];  
  
    }  
  
}
```



```
return rev;
```

```
}
```

```
}
```

## 19. Check whether a string is anagram of another string?

```
public class AnagramString {
```

```
    static void isAnagram(String str1, String str2) {
```

```
        String s1 = str1.replaceAll("\\s", "");
```

```
        String s2 = str2.replaceAll("\\s", "");
```

```
        boolean status = true;
```

```
        if (s1.length() != s2.length()) {
```

```
            status = false;
```

```
        } else {
```

```
            char[] ArrayS1 = s1.toLowerCase().toCharArray();
```

```
            char[] ArrayS2 = s2.toLowerCase().toCharArray();
```

```
            Arrays.sort(ArrayS1);
```

```
            Arrays.sort(ArrayS2);
```

```
            status = Arrays.equals(ArrayS1, ArrayS2);
```

```
        }
```

```
        if (status) {
```

```
            System.out.println(s1 + " and " + s2 + " are anagrams");
```

```
        } else {
```

```
            System.out.println(s1 + " and " + s2 + " are not anagrams");
```

```
        }
```

```
}  
  
}
```

## 20. In a given string, change few characters to upper case as asked?

```
public class ChangeCharCase {  
  
    public static void main(String[] args) throws IOException {  
  
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));  
  
        System.out.println("Enter the string:");  
  
        String name = br.readLine();  
  
        String s1 = name.substring(0, 1).toUpperCase();  
  
        String nameCapitalized = s1 + name.substring(1);  
  
        System.out.println(nameCapitalized);  
  
    }  
  
}
```

## 21. In a given string, print the occurrence of each character?

```
public class FindFrequencyOfCharactersInString {  
  
    public static void main(String[] args){  
  
        Map<Character,Integer> map = new HashMap<>();  
  
        map = findFrequency("Java J2EE Java JSP J2EE");  
  
        for(Map.Entry<Character,Integer> entry: map.entrySet()){  
  
            System.out.println("Frequency of character " + entry.getKey() + " is " + entry.getValue());  
  
        }  
  
    }  
  
}
```

```

public static Map<Character,Integer> findFrequency(String input){

char[] arr = input.toLowerCase().replaceAll("\\s+","").toCharArray();

Map<Character,Integer> frequency = new HashMap<>();

for(char c: arr){

if(frequency.containsKey(c)){

frequency.put(c,frequency.get(c) + 1);

}

else{

frequency.put(c,1);

}

}

return frequency;

}

}

```

## 22. Find the duplicate strings in a given statement and remove them?

```

public class RemoveDuplicateWords

{

public static void main(String[] args)

{

String input="Welcome to QAScript Java Interview Question in QAScript";

String[] words=input.split(" ");

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

{

```

```

if(words[i]!=null)
{
    for(int j=i+1;j<words.length;j++)
    {
        if(words[i].equals(words[j]))
        {
            words[j]=null;
        }
    }
}

for(int k=0;k<words.length;k++)
{
    if(words[k]!=null)
    {
        System.out.println(words[k]);
    }
}

```

### 23. Use split method to print each word of a statement?

```

public class SplitStatement{

    public static void main(String args[]){

```

```

String strMain ="Welcome to QAScript";

String[] arrSplit = strMain.split("\\s");

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

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

}

}

}

```

## 24. Find and remove the duplicate characters from a given string and print ?

```

public static String removeDuplicates(String word){

Set<Character> chars = new HashSet<>();

StringBuilder output = new StringBuilder(word.length());

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

char ch = word.charAt(i);

if (chars.add(ch)) {

output.append(ch);

}

}

return output.toString();

}

```

## 25. Write a Java program to print the triangle of numbers?

```

public class FloydTriangle {

public static void main(String[] args){

```

```

Scanner sc = new Scanner(System.in);

System.out.println("Enter the number of rows");

int rows = sc.nextInt();

printFloydTriangle(rows);

}

public static void printFloydTriangle(int n){

int number = 1;

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

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

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

number++;

}

System.out.println();

}

}

}

```

# Java for Testers – Interview Questions and Answers Part-8

## 1) What are the additional features in Java 8?

- Lambda Expressions: Lambda expressions gives the ability to pass a functionality as a method argument. Lambda expression help us reduce the code clutter in using a single method class.
- Pipelines and Streams: Pipeline and streams will make our task easier in

accessing the elements from collections and applying operations on it.

- Date and Time API: This helps in handling date and time operations in an easier way.
- Default Methods: Default methods gives the ability to add default implementation for methods in an interface.
- Type Annotations: Annotations can be applied wherever a type is used like in new instance creates, exception throws clause.

## **2) What is the difference between for and for-each loops in Java?**

- for loop is a control structure for specifying iteration that allows code to be repeatedly executed but foreach loop is a control structure for traversing items in an array or a collection.
- for loop can be used to retrieve a particular set of elements but foreach loop cannot be used to retrieve a particular set of elements.
- for loop is harder to read and write than the foreach loop but foreach loop is easier to read and write than the for loop.
- for loop is used as a general purpose loop but foreach loop is used for arrays and collections.

## **3) Can we have multiple public classes inside a single class in Java?**

Yes. There are two ways of implementing multiple classes:

- Nested Classes
- Multiple non-nested classes

## **4) What are the different types of inheritance in Java?**

- Single Inheritance : In single inheritance, subclasses inherit the features of one superclass.
- Multilevel Inheritance : In Multilevel Inheritance, a derived class will be inheriting a base class and as well as the derived class also act as the base class to other class.
- Hierarchical Inheritance : In Hierarchical Inheritance, one class serves as a superclass (base class) for more than one sub class.

## **5) What is Polymorphism, give example and how can we achieve it?**

Polymorphism allows us to perform a single action in different ways. In other words, polymorphism allows you to define one interface and have multiple implementations.

In Java polymorphism is mainly divided into two types:

- Compile Time Polymorphism
- Run Time Polymorphism

### **6) Can we achieve method overloading when the two methods have only difference in return type?**

No. The compiler will give error as the return value alone is not sufficient for the compiler to figure out which function it has to call. Only if both methods have different parameter types (so, they have different signature), then Method overloading is possible.

### **7) What is Garbage Collection in Java and how exactly it is done?**

Garbage collection (GC), as its name implies, is a means of freeing space occupied by waste materials, or garbage, and avoid memory leaks. Through performing the GC mechanism, available memory can be effectively used. Moreover, through this process, objects that are dead or unused for a long time in the memory heap will be deleted and the memory space used by these objects will be reclaimed.

Garbage collection in Java is an automatic process and the programmer does not have to explicitly mark objects to be deleted. The implementation mainly lives in the JVM. Each JVM can implement garbage collection. The only requirement is that it should meet the JVM specification.

### **8) What is encapsulation in Java?**

Encapsulation is defined as the wrapping up of data under a single unit. It is the mechanism that binds together code and the data it manipulates.

In encapsulation, the variables or data of a class is hidden from any other class and can be accessed only through any member function of own class in which they are declared.

Encapsulation can be achieved by declaring all the variables in the class as private.



### 9) What is IS-A and Has-A relationship in Java with examples?

In Object oriented programming, IS-A relationship denotes "one object is type of another". IS-A relation denotes Inheritance methodology.

In Object orientation design, We can say "class one is in Has-A relationship with class B if class A holds reference of Class B".

### 10) What is super and final keywords in Java and the difference between them?

Super is a keyword of Java which refers to the immediate parent of a class and is used inside the subclass method definition for calling a method defined in the superclass. A superclass having methods as private cannot be called. Only the methods which are public and protected can be called by the keyword super. It is also used by class constructors to invoke constructors of its parent class.

Final is a keyword in Java that is used to restrict the user and can be used in many respects. Final can be used with:

- Class
- Methods
- Variables

### 11) Explain run time polymorphism and compile time polymorphism with examples?

- Compile time Polymorphism: It is also known as static polymorphism. This type of polymorphism is achieved by function overloading or operator overloading.

```
// Java program for Method overloading
```

```
class OverloadedMethods {
```

```
// Method with 2 parameter
```

```
static int Multiply(int a, int b)
```

```
{
```

```
return a * b;
```

```

}

// Method with the same name but 3 parameter

static int Multiply(int a, int b, int c)

{

return a * b * c;

}

}

class Main {

public static void main(String[] args)

{

System.out.println(MultiplyFun.Multiply(5,2));

System.out.println(MultiplyFun.Multiply(1,4,6));

}

}

```

Output:

10  
24

– Runtime Polymorphism: It is also known as Dynamic Method Dispatch. It is a process in which a function call to the overridden method is resolved at Runtime. This type of polymorphism is achieved by Method Overriding.

```

// Java program for Method overriding

class Parent {

void Print()

{

System.out.println("parent class");

```

```

}

}

class subclass1 extends Parent {

void Print()

{

System.out.println("subclass1");

}

}

class TestPolymorphism {

public static void main(String[] args)

{

Parent a;

a = new subclass1();

a.Print();

}

}

```

Output: subclass1

## 12) Can final methods be overloaded?

Yes. It is possible to overload the final methods.

Ex-

```

public final void addNumbers(int a, int b){}

public final void addNumbers(float a, float b) {}

```

### 13) Can static methods be overloaded?

Yes. We can have two or more static methods with same name, but differences in input parameters.

Ex-

```
public class Test {  
  
    public static void print() {  
  
        System.out.println("Test.print() called ");  
  
    }  
  
    public static void print(int a) {  
  
        System.out.println("Test.print(int) called ");  
  
    }  
  
    public static void main(String args[])  
  
    {  
  
        Test.print();  
  
        Test.print(10);  
  
    }  
  
}
```

Output:

Test.print() called

Test.print(int) called

### 14) Can final methods be overridden?

No. Any method that is declared as final in the superclass cannot be overridden by a subclass. If we try to override the final method of super class we will get an error in Java.

### 15) Can static methods be overridden?

No. If a derived class defines a static method with same signature as a static

method in base class, the method in the derived class hides the method in the base class.

#### **16) Can we overload a main method in Java?**

Yes, We can overload the main method in java but JVM only calls the original main method, it will never call our overloaded main method.

#### **17) Can we execute a class without a main method?**

No. We cannot execute a class without a main method.

#### **18) What is a Package in Java?**

Package in Java is a mechanism to encapsulate a group of classes, sub packages and interfaces. Packages are used for:

- Preventing naming conflicts.
- Making searching/locating and usage of classes, interfaces, enumerations and annotations easier
- Providing controlled access: protected and default have package level access control.
- Data encapsulation (or data-hiding).

#### **19) What is an Abstract Class in Java and explain with an example?**

A class which contains the abstract keyword in its declaration is known as abstract class.

- Abstract classes may or may not contain abstract methods, i.e., methods without body.
- But, if a class has at least one abstract method, then the class must be declared abstract.
- If a class is declared abstract, it cannot be instantiated.
- To use an abstract class, you have to inherit it from another class, provide implementations to the abstract methods in it.
- If you inherit an abstract class, you have to provide implementations to all the abstract methods in it.

Ex –

```
// An example abstract class in Java
```

```
abstract class Shape {
```

```
int color;
```

```
// An abstract function
```

```
abstract void draw();  
  
}
```

## **20) What is an Interface and how it is different from Abstract Class?**

Like a class, an interface can have methods and variables, but the methods declared in an interface are by default abstract (only method signature, no body).

- Interfaces specify what a class must do and not how. It is the blueprint of the class.
- If a class implements an interface and does not provide method bodies for all functions specified in the interface, then the class must be declared abstract.

Interface can have only abstract methods. Abstract class can have abstract and non-abstract methods.

## **21) Can we use private and protect access modifiers inside an Interface?**

Java interfaces are meant to specify fields and methods that are publicly available in classes that implement the interfaces. Therefore you cannot use the private and protected access modifiers in interfaces.

## **22) Can multiple inheritances supported in Interface?**

Yes. Multiple inheritance is supported in Interfaces.

## **23) What is the difference between throw and throws?**

- throw keyword is used to throw an exception explicitly but Throws keyword is used to declare one or more exceptions, separated by commas.
- Only single exception is thrown by using throw but multiple exceptions can be thrown by using throws.
- throw keyword is used within the method but throws keyword is used with the method signature.
- Unchecked exception can be propagated using throw but checked

exception must use throws keyword followed by specific exception class name.

#### **24) What is Exception and what is its base class?**

An exception is an unwanted or unexpected event, which occurs during the execution of a program i.e at run time, that disrupts the normal flow of the program's instructions.

All exception and errors types are sub classes of class Throwable, which is base class of hierarchy. This class is used for exceptional conditions that user programs should catch. NullPointerException is an example of such an exception.

#### **25) What is Final, Finally and Finalize?**

- Final is used to apply restrictions on class, method and variable. Final class can't be inherited, final method can't be overridden and final variable value can't be changed.
- The finally keyword is used in association with a try/catch block and guarantees that a section of code will be executed, even if an exception is thrown.
- Finalize is used to perform clean up processing just before object is garbage collected.

## **Java for Testers – Interview Questions and Answers Part-9**

#### **1) JVM is dependent or independent platform?**

The JVM is not platform independent. Java Virtual Machine (JVM) provides the environment to execute the java file(. Class file).

So at the end it's depends on your kernel , and kernel is differ from OS (Operating System) to OS.

The JVM is used to both translate the bytecode into the machine language for a particular computer, and actually execute the corresponding machine-

language instructions as well. Without the JVM, you can't run a Java application.

## **2) What is toString() method ?What happens when I use in the program?**

Object class contains toString() method. We can use toString() method to get string representation of an object. Whenever we try to print the Object reference then internally toString() method is invoked.

## **3) What is the capacity of String Buffer?**

Every string buffer has a capacity. As long as the length of the character sequence contained in the string buffer does not exceed the capacity, it is not necessary to allocate a new internal buffer array. If the internal buffer overflows, it is automatically made larger.

## **4) What is dom concept?**

The Document Object Model is a cross-platform and language-independent interface that treats an XML or HTML document as a tree structure wherein each node is an object representing a part of the document. The DOM represents a document with a logical tree.

## **5) What is generics?**

Generics enable types (classes and interfaces) to be parameters when defining classes, interfaces and methods. Much like the more familiar formal parameters used in method declarations, type parameters provide a way for you to re-use the same code with different inputs.

## **6) What is synchronization?**

Synchronization in java is the capability to control the access of multiple threads to any shared resource. Java Synchronization is better option where we want to allow only one thread to access the shared resource.

## **7) What is the difference between hashmap and hash set?**

- HashMap stores a key-value pair but HashSet stores objects
- HashMap doesn't allow duplicate keys and allows duplicate values but



HashSet doesn't allow duplicate values

- HashMap is faster than HashSet because values are associated with a unique key but HashSet is slower than HashMap because the member object is used for calculating hashCode value, which can be same for two objects
- HashMap contains a null key and multiple null values but HashSet contains a single null value
- In HashMap 2 objects are created during put operation, one for key and one for value but in HashSet only 1 object is created for add operation

### **8) What is the difference between set and linkedlist?**

- List is an ordered collection it maintains the insertion order, which means upon displaying the list content it will display the elements in the same order in which they got inserted into the list.

Set is an unordered collection, it doesn't maintain any order. There are few implementations of Set which maintains the order such as LinkedHashSet (It maintains the elements in insertion order).

- List allows duplicates while Set doesn't allow duplicate elements. All the elements of a Set should be unique if you try to insert the duplicate element in Set it would replace the existing value.
- List allows any number of null values. Set can have only a single null value at most.

### **9) What is the difference between arraylist and vector list?**

- ArrayList is not synchronized but Vector is synhronized
- ArrayList is fast because it is non-synchroized but Vector is slow because it is synchroized
- ArrayList is not a legacy class but Vector is a legacy class.

### **10) What is the difference between linkedhash set and hashset?**

- HashSet does not maintain insertion order but LinkedHashSet maintains insertion order of objects
- HashSet performance is better than LinkedHashSet because LinkedHashSet maintains LinkedList internally to maintain the insertion order of elements

### **11) What are the types of assertion and what are assertion in java?**

An assertion allows testing the correctness of any assumptions that have been made in the program. Assertion is achieved using the assert statement in Java. While executing assertion, it is believed to be true. If it fails, JVM throws an error named AssertionError. It is mainly used for testing purposes during development.

## **12) What is the default package in java ?**

The Java standard libraries include java.lang package by default, which contains a number of components that are used very commonly in Java programs.

## **13) What are inner classes ..name them ?**

Inner class means one class which is a member of another class. There are basically four types of inner classes in java.

- Nested Inner class
- Method Local inner classes
- Anonymous inner classes
- Static nested classes

## **14) In public static void main(String arr[])... what if i replace public with private ..... remove static .....replace void with string?**

If we replace public with private then JVM will be unable to access/locate the main method.

If we remove static then JVM cannot invoke it without instantiating class.

If we replace void with String, it will give compilation error as an unexpected return value.

## **15) In hash map we have (key and value ) pair , can we store inside a value =(key, value ) again ?**

Yes. We can store key/value inside a value.

## **16) What are variable scope in java (in class , in method , in static block)?**

- Each variable declared inside a class with private access modifier but outside of any method, has class scope. As a result, these variables can be

used everywhere in the class, but not outside of it.

- When a variable is declared inside a method, it has method scope and it will only be valid inside the same method.
- Variables declared inside a block are accessible only inside of that block.

## 17) Write a Java program so that when ever you create a object, you get to know how many object u have created?

// Java program Find Out the Number of Objects Created of a Class

```
class FindCreatedObjects {

    static int noOfObjects = 0;

    {

        noOfObjects += 1;

    }

    public Test()

    {

    }

    public Test(int n)

    {

    }

    public Test(String s)

    {

    }

    public static void main(String args[])

    {

        Test t1 = new Test();

        Test t2 = new Test(5);

        Test t3 = new Test("Automation");
```

```
System.out.println(Test.noOfObjects);  
  
}  
  
}
```

### **18) What is difference between .equals() , (==) and compare-to(); ?**

- Double equals operator is used to compare two or more than two objects, If they are referring to the same object then return true, otherwise return false.
- String equals() method compares the two given strings based on the data / content of the string. If all the contents of both the strings are same then it returns true. If all characters are not matched then it returns false.
- The java compare two string is based on the Unicode value of each character in the strings. If two strings are different, then they have different characters at some index that is a valid index for both strings, or their lengths are different, or both.

### **19) What is the difference between hash code and equals?**

- equals: a method provided by java.lang.Object that indicates whether some other object passed as an argument is “equal to” the current instance. Two objects are equal if and only if they are stored in the same memory address.
- hashCode(): a method provided by java.lang.Object that returns an integer representation of the object memory address. By default, this method returns a random integer that is unique for each instance. This integer might change between several executions of the application and won’t stay the same.

### **20) Do you have any idea on Enumeration?**

Enumeration means a list of named constant. In Java, enumeration defines a class type. An Enumeration can have constructors, methods and instance variables. It is created using enum keyword. Each enumeration constant is public, static and final by default. Even though enumeration defines a class type and have constructors, you do not instantiate an enum using new. Enumeration variables are used and declared in much a same way as you do a primitive variable.

## **21) What is priority queue in collection, what is its use and how you have use in your project?**

A PriorityQueue is used when the objects are supposed to be processed based on the priority. The PriorityQueue is based on the priority heap. The elements of the priority queue are ordered according to the natural ordering, or by a Comparator provided at queue construction time, depending on which constructor is used.

## **22) What is the difference between MAP & Set ?**

- While a Map holds two objects per Entry e.g. a key and a value and It may contain duplicate values but keys are always unique. Set doesn't allow duplicates.
- Set just allow one null element as there is no duplicate permitted while in Map you can have null values and at most one null key.

## **23) How to call the super class method in subclass?**

If you override a method in a subclass, but still need to call the method defined in the superclass, you can do so using the super reference, like this:  
public class Car extends Vehicle {

```
public void setLicensePlate(String license) {  
  
    super.setLicensePlate(license);  
  
}  
  
}
```

## **24) What is the base class for all java classes? And mention its methods?**

The Object class is the parent class of all the classes in java by default.

Following are different methods of Object class:

- getClass(): returns the Class class object of this object. The Class class can further be used to get the metadata of this class.
- hashCode(): returns the Class class object of this object. The Class class can further be used to get the metadata of this class.

- equals(Object obj): compares the given object to this object.
- toString(): returns the string representation of this object.

## 25) What is hashCode method? Explain pragmatically by implementing hashCode method?

The default implementation of hashCode() in Object class returns distinct integers for different objects.

// Java puzzle to illustrate use of hashCode() and equals() method

```
public class HashCodeExample {  
  
    private final String first, last;  
  
    public Name(String first, String last)  
  
    {  
  
        this.first = first;  
  
        this.last = last;  
  
    }  
  
    public boolean equals(Object o)  
  
    {  
  
        if (!(o instanceof Name))  
  
            return false;  
  
        Name n = (Name)o;  
  
        return n.first.equals(first) && n.last.equals(last);  
  
    }  
  
    public static void main(String[] args)  
  
    {  
  
        Set<Name> s = new HashSet<Name>();  
  
        s.add(new Name("Bijan", "Patel"));
```

```
System.out.println(  
  
s.contains(new Name("Bijan", "Patel")));  
  
}  
  
}
```

# Java for Testers – Interview Questions and Answers Part-10

---

## 1) What is a Class in Java?

A class is the blueprint from which individual objects are created. It is the basic building block of an object-oriented language such as Java.

Ex-

```
class ClassName {  
    // variables  
    // methods  
}
```

---

## 2) What is the difference between heap and stack?

- The main difference between heap and stack is that stack memory is used to store local variables and function call while heap memory is used to store objects in Java.
- If there is no memory left in the stack for storing function call or local variable, JVM will throw `java.lang.StackOverflowError`, while if there is no more heap space for creating an object, JVM will throw `java.lang.OutOfMemoryError: Java Heap Space`.
- The size of stack memory is a lot lesser than the size of heap memory in

Java.

- Variables stored in stacks are only visible to the owner Thread while objects created in the heap are visible to all thread. In other words, stack memory is kind of private memory of Java Threads while heap memory is shared among all threads.

---

### **3) What is a Constructor and the different types of Constructors?**

A constructor in Java is a special method that is used to initialize objects. The constructor is called when an object of a class is created. It can be used to set initial values for object attributes. There are two types of constructor in Java:

- No-argument constructor: A constructor that has no parameter is known as default constructor. If we don't define a constructor in a class, then compiler creates default constructor (with no arguments) for the class.
  - Parameterized Constructor: A constructor that has parameters is known as parameterized constructor. If you want to initialize fields of the class with your own values, then use a parameterized constructor.
- 

### **4) What is the difference between break and continue statements?**

- Break leaves the loop completely and executes the statements after the loop. Continue leaves the current iteration and executes with the next value in the loop.
  - Break completely exits the loop. Continue skips the statements after the continue statement and keeps looping.
- 

### **5) What is the command used in Java for exiting the system from current execution?**

The `java.lang.System.exit()` method exits current program by terminating running Java virtual machine. This method takes a status code.



---

### **6) What is the method from which Java Programs starts execution?**

A Java program starts by executing the main method of some class.

---

### **7) Give some examples for compile time errors in Java Programs?**

- File Not Found Exception
  - No Such Field Exception
  - Interrupted Exception
  - No Such Method Exception
  - Class Not Found Exception
- 

### **8) What is the difference between print and println statements in Java?**

The only difference between `println()` and `print()` method is that `println()` throws the cursor to the next line after printing the desired result whereas `print()` method keeps the cursor on the same line.

---

### **9) What are the different types of comments in Java?**

In Java there are three types of comments:

- Singleline comments: The single line comment is used to comment only one line.
  - Multiline comments: The multi line comment is used to comment multiple lines of code.
  - Documentation comments: The documentation comment is used to create documentation API. To create documentation API, you need to use javadoc tool.
-

### **10) What are the different things required for storing data in Java?**

- Stack Memory
  - Heap Memory
  - External data source
- 

### **11) What are the different data types in Java and what is their purpose?**

Data types are divided into two groups:

Primitive data types – includes byte, short, int, long, float, double, boolean and char

Non-primitive data types – such as String, Arrays and Classes

---

### **12) Is String a primitive data type?**

String is not a primitive data type. Java.lang package provides the String class therefore, it is an object type.

---

### **13) What are the different types of Operators in Java?**

- Arithmetic Operators: They are used to perform simple arithmetic operations on primitive data types.
- Unary Operators: Unary operators need only one operand. They are used to increment, decrement or negate a value.
- Assignment Operator: Assignment operator is used to assign a value to any variable.
- Relational Operators: These operators are used to check for relations like equality, greater than, less than. They return boolean result after the comparison
- Logical Operators: These operators are used to perform “logical AND” and “logical OR” operation, i.e. the function similar to AND gate and OR gate in digital electronics.
- Ternary Operator: Ternary operator is a shorthand version of if-else

statement. It has three operands and hence the name ternary.

- Bitwise Operators: These operators are used to perform manipulation of individual bits of a number. They can be used with any of the integer types.
  - Shift Operators: These operators are used to shift the bits of a number left or right thereby multiplying or dividing the number by two respectively.
- 

#### **14) What are the different flow control structures in Java?**

There are 3 types of flow control structures:

- Decision-making statements : if-then, if-then-else, switch.
  - Looping statements : for, while, do-while.
  - Branching statements : break, continue, return.
- 

#### **15) What is the difference between while and do..while loop in Java?**

If the condition in a while loop is false, not a single statement inside the loop is executed. In contrast, if the condition in 'do-while' loop is false, then also the body of the loop is executed at least once then the condition is tested.

---

#### **16) Can we call the same method multiple times in Java?**

We can use multiple threads to call same method multiple times.

---

#### **17) What is the keyword to be used in Java while declaring methods, when the method don't have anything to return?**

Void() can be used for methods which don't return anything.

---

#### **18) What are the different types of Arrays in Java?**

There are two types of array.

- Single Dimensional Array: In it each element is represented by a single subscript. The elements are stored in consecutive memory locations.
  - Multidimensional Array: In it each element is represented by two subscripts.
- 

### **19) What is Abstraction in Java?**

Data abstraction is the process of hiding certain details and showing only essential information to the user. Abstraction can be achieved with either abstract classes or interfaces.

---

### **20) What is the difference between Java and JavaScript?**

- Java is strongly typed language and variable must be declare first to use in program. JavaScript is weakly typed language and have more relaxed syntax and rules.
  - Java is an object oriented programming language. JavaScript is an object based scripting language.
  - Java applications can run in any virtual machine(JVM) or browser. JavaScript code run on browser only as JavaScript is developed for browser only.
  - Java program uses more memory. JavaScript requires less memory therefore it is used in web pages.
  - Java has a thread based approach to concurrency. Javascript has event based approach to concurrency.
- 

### **21) How to find duplicate elements in a Java Array?**

```
public class DuplicateElement {  
  
    public static void main(String[] args) {
```

```
//Initialize array

int [] arr = new int [] {1, 2, 3, 4, 2, 7, 8, 8, 3};

System.out.println("Duplicate elements in given array: ");

//Searches for duplicate element

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

    for(int j = i + 1; j < arr.length; j++) {

        if(arr[i] == arr[j])

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

    }

}

}
```

---

## 22) How to find the smallest and largest numbers in a Java Array?

```
public class FindLargestSmallestNumber {

    public static void main(String[] args) {

        //numbers array

        int numbers[] = new int [] {55,32,45,98,82,11,9,39,50};

        //assign first element of an array to largest and smallest

        int smallest = numbers[0];

        int largest= numbers[0];

        for (int i = 1; i < numbers.length; i++) {

            if (numbers[i] > largest)

                largest= numbers[i];

            else if (numbers[i] < smallest)
```

```

smallest = numbers[i];

}

System.out.println("Largest Number is : " + largest);

System.out.println("Smallest Number is : " + smallest);

}

}

```

---

### 23) Explain String manipulation in Java?

Java String class provides a lot of methods to perform operations on strings such as `compare()`, `concat()`, `equals()`, `split()`, `length()`, `replace()`, `compareTo()`, `intern()`, `substring()` etc.

---

### 24) Write a Java program to swap two strings without using temp or third variable?

```

class SwapNumber {

public static void main(String a[])

{

int x = 10;

int y = 5;

x = x + y;

y = x - y;

x = x - y;

System.out.println("After swaping:"

+ " x = " + x + ", y = " + y);

}

```

}

---

### **25) How to achieve multiple inheritance in Java?**

Multiple inheritance in Java can be achieved by using interfaces.

## **Java for Testers – Interview Questions and Answers Part-11**

### **1) What is verbose?**

Verbose is an option that can be used on the Java command to enable debug output for class loading.

### **2) What is thread count?**

The Thread Count parameter specifies the maximum number of simultaneous requests the server can handle.

When the server has reached the limit or request threads, it defers processing new requests until the number of active requests drops below the maximum amount.

### **3) When a class implements two interfaces having a common method, which interface method does the class implement?**

If a type implements two interfaces, and each interface define a method that has identical signature, then in effect there is only one method, and they are not distinguishable. There will be one implementation of both.

### **4) What is the flow of try, catch and finally blocks?**

If a statement in try block raised an exception, then the rest of the try block doesn't execute and control passes to the corresponding catch block. After executing the catch block, the control will be transferred to finally block(if present) and then the rest program will be executed.

### **5) What is finally in Java?**

A finally block contains all the crucial statements that must be executed whether exception occurs or not.

The statements present in this block will always execute regardless of whether exception occurs in try block or not such as closing a connection, stream etc.

### **6) How many values will a method return in Java?**

We can return only one value from a method in Java.

### **7) What is abstract in Java?**

An abstract class, in the context of Java, is a superclass that cannot be instantiated and is used to state or define general characteristics. An object cannot be formed from a Java abstract class.

Trying to instantiate an abstract class only produces a compiler error.

### **8) Which design patterns have you used to develop frameworks?**

We have used Singleton Design pattern to develop frameworks.

### **9) try block has exit method, catch block has exception and finally block is returning an int value. Can you explain the flow of execution?**

It will exit the try block and skip the catch block. Then it will execute finally block to return int value.

### **10) Have you ever come across conflict in code. How do you resolve?**

To resolve a merge conflict caused by competing line changes, you must choose which changes to incorporate from the different branches in a new commit.

### **11) Can a constructor be private?**

Yes, we can declare a constructor as private. If we declare a constructor as private then we cannot create an object of that class.

### **12) Assert is an abstract or static method?**

Assert is a static method type.



### **13) What is instance block at class level and method level?**

Instance Initialization Blocks are used to initialize instance variables.

### **14) What is the difference between a Class and Interface?**

- A class describes the attributes and behaviors of an object. An interface contains behaviors that a class implements.
- A class may contain abstract methods, concrete methods. An interface contains only abstract methods.

### **15) Can we have multiple catch blocks in try catch statements in Java?**

Yes we can have multiple catch blocks in try catch statements in Java.

### **16) What is reflection in Java?**

Reflection is an API which is used to examine or modify the behavior of methods, classes, interfaces at runtime.

### **17) What is nested Class?**

A nested class is a member of its enclosing class. A nested class has access to the members, including private members, of the class in which it is nested.

### **18) What is mutable and immutable?**

Mutable objects have fields that can be changed, immutable objects have no fields that can be changed after the object is created.

A class is immutable, as in, it is not mutable.

### **19) What are the types of Synchronization in Java?**

Multi-threaded programs may often come to a situation where multiple threads try to access the same resources and finally produce erroneous and unforeseen results. So it needs to be made sure by some synchronization method that only one thread can access the resource at a given point of time.

### **20) What are the types of Exceptions in Java?**

Java's exceptions can be categorized into two types:

- Checked exceptions
- Unchecked exceptions

### **21) What is a class and object?**

A class is a user defined blueprint or prototype from which objects are created.

It represents the set of properties or methods that are common to all objects of one type.

An object is an instance of a class.

### **22) How to call the function of a Class without creating an object?**

Declare the function as static so that it can be called without creating an object of the class.

### **23) What is HashMap?**

HashMap is a Map based collection class that is used for storing Key & value pairs.

### **24) How to synchronize Collection classes?**

The `synchronizedCollection()` method of `java.util.Collections` class is used to return a synchronized (thread-safe) collection backed by the specified collection.

### **25) Why `main()` is declared as static?**

The `main()` method is static so that JVM can invoke it without instantiating the class.

# **Java for Testers – Interview Questions and Answers Part-12**

## **1) Explain about Diamond problem in Java?**

The “diamond problem” is an ambiguity that can arise as a consequence of allowing multiple inheritance. The problem occurs when there exist methods with same signature in both the super classes and subclass. On calling the method, the compiler cannot determine which class method to be called and even on calling which class method gets the priority.

## **2) What are primitive data types in Java?**

There are 8 primitive data types:

- boolean: boolean data type represents only one bit of information either true or false
- byte: The byte data type is an 8-bit signed two’s complement integer.
- short: The short data type is a 16-bit signed two’s complement integer
- int: It is a 32-bit signed two’s complement integer.
- long: The long data type is a 64-bit two’s complement integer
- float: The float data type is a single-precision 32-bit IEEE 754 floating point
- double: The double data type is a double-precision 64-bit IEEE 754 floating point
- char: The char data type is a single 16-bit Unicode character

## **3) Why String is non primitive?**

String is not a primitive data type. Java.lang package provides the String class therefore, it is an object type.

## **4) Can we create an object for an interface?**

We can’t instantiate an interface in java. That means we cannot create the object of an interface.

## **5) Difference between Instantiate and Initialize in Java?**

Initialization-Assigning a value to a variable i.e a=0,setting the initial values.  
Instantiation- Creating the object i.e when we reference a variable to an object with new operator.

## **6) Difference between == and =?**

== is used to compare values and = is used to assign a value.

**7) Are all methods in an abstract class, abstract?**

No. Abstract class can have both abstract and non-abstract methods.

**8) What's Singleton class?**

Singleton is a design pattern rather than a feature specific to Java. It ensures that only one instance of a class is created.

**9) Can we have Finally block without Try & Catch blocks?**

We can use finally block without a catch block but we need to have try block.

**10) How to execute Java program from the command prompt?**

We use Java compiler javac to compile and the Java interpreter java to run the Java program.

**11) Any example or practical usage of Run time polymorphism?**

Method overriding is an example of runtime polymorphism. In method overriding, a subclass overrides a method with the same signature as that of in its superclass

**12) Why we use public static void for main?**

It is made public so that JVM can invoke it from outside the class as it is not present in the current class. The main() method is static so that JVM can invoke it without instantiating the class. As main() method doesn't return anything, its return type is void. It is the identifier that the JVM looks for as the starting point of the java program.

**13) What is the difference between error and exception?**

Exceptions are the conditions that occur at runtime and may cause the termination of program.

An Error "indicates serious problems that a reasonable application should not try to catch."

**14) Is Java a value based or reference based?**

Java uses only call by value while passing reference variables as well. It creates a copy of references and passes them as valuable to the methods.

### **15) Explain collections?**

The Java collections framework provides a set of interfaces and classes to implement various data structures and algorithms.

### **16) How to synchronize collection class?**

The `synchronizedCollection()` method of Java Collections class is used to get a synchronized (thread-safe) collection backed by the specified collection.

### **17) How do you know when to use abstract class and Interface?**

If the functionality we are creating will be useful across a wide range of disparate objects, use an interface. Abstract classes should be used primarily for objects that are closely related, whereas interfaces are best suited for providing a common functionality to unrelated classes.

### **18) What are the different access specifiers in Java?**

There are four types of access modifiers available in java:

Default – When no access modifier is specified for a class , method or data member. It is said to be having the default access modifier by default.

Private – The methods or data members declared as private are accessible only within the class in which they are declared.

Protected – The methods or data members declared as protected are accessible within same package or sub classes in different package.

Public – The public access modifier is specified using the keyword public.

### **19) Write a Java program to demonstrate the advantage of finally in Exception Handling?**

```
public class Example {  
  
    public static void main(String[] args) {  
  
        BufferedReader br = null;  
  
        try {
```

```

br = new BufferedReader(new FileReader("C:\sample.txt"));

System.out.println(br.readLine());

} catch (IOException e) {

e.printStackTrace();

} finally {

try {

if (br != null)

br.close();

} catch (IOException ex) {

ex.printStackTrace();

}

}

}

}

```

## 20) Is it mandatory to have the same name for Class name and Java file ?

While writing a java program first it is saved as a ".java" file, when it is compiled it forms byte code which is a ".class" file

## 21) How do you write custom class which is immutable?

```

public final class Student

{

final String name;

final int regNo;

public Student(String name, int regNo)

```

```

{

this.name = name;

this.regNo = regNo;

}

public String getName()

{

return name;

}

public int getRegNo()

{

return regNo;

}

}

```

## 22) How to execute multiple Java programs at a time?

We can execute multiple java programs using multithreading.

## 23) How do you call function in java?

To call a function in Java, write the function's name followed by two parentheses () and a semicolon.

## 24) Can we execute java code without main()?

Yes, we can execute a java program without a main method by using a static block.

Static block in Java is a group of statements that gets executed only once when the class is loaded into the memory by Java ClassLoader, It is also known as a static initialization block.

## 25) What are abstract classes and methods in Java?

Abstract classes may or may not contain abstract methods, i.e., methods without body.

# Java for Testers – Interview Questions and Answers Part-13

1) Java Program: Change a string such that first character is upper case, second is lower case and so on?

```
public static String changeToUpperCase(String str) {  
  
    StringBuffer s = new StringBuffer();  
  
    char ch = '';  
  
    for (int i = 0; i < str.length(); i++) {  
  
        if (ch == '' && str.charAt(i) != '')  
  
            s.append(Character.toUpperCase(str.charAt(i)));  
  
        else  
  
            s.append(str.charAt(i));  
  
        ch = str.charAt(i);  
  
    }  
  
    return s.toString().trim();  
  
}
```

2) Java Program: To find duplicates in a list?

```
public static Set<String> findDuplicates(List<String> listDuplicates) {
```



```

final Set<String> set1 = new HashSet<String>();

final Set<String> set2 = new HashSet<String>();

for (String str : listDuplicates) {

    if (!set1.add(str)) {

        set2.add(str);

    }

}

return set2;

}

```

3) Write a program to compare two Hashmap for equality?

```

public static Boolean compareMaps(){

    Map<String, String> country1 = new HashMap<String, String>();

    country1.put("Japan", "Tokyo");

    country1.put("China", "Beijing");

    Map<String, String> country2 = new HashMap<String, String>();

    country2.put("Japan", "Tokyo");

    country2.put("China", "Beijing");

    if(country1==country2){

        return true;

    } else

        return false;

}

```

4) Write a Java program to demonstrate creating an Array?

```
public static void main (String[] args)
{
    int[] arr;
    arr = new int[5];
    arr[0] = 10;
    arr[1] = 20;
    arr[2] = 30;
    arr[3] = 40;
    arr[4] = 50;
    for (int i = 0; i < arr.length; i++)
        System.out.println("Element at index " + i + " : " + arr[i]);
}
```

5) Java Program: String s = "sub53od73th"; Eliminate the numbers alone. Print the Alphabets.

```
public static String removeNumFromString(String str) {
    if (str == null) {
        return null;
    }
    char[] ch = str.toCharArray();
    int length = ch.length;
```

```

StringBuilder sb = new StringBuilder();

int i = 0;

while (i < length) {

    if (Character.isDigit(ch[i])) {

        i++;

    } else {

        sb.append(ch[i]);

        i++;

    }

}

return sb.toString();

}

```

6) Java program: Reverse the words in the sentence?

```

public static String reverseTheSentence(String inputString)

{

    String[] words = inputString.split("\\s");

    String outputString = "";

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

    {

        outputString = outputString + words[i] + " ";

    }

    return outputString;

}

```

7) Java program: Find the count of each element in a two dimensional matrix?

```
static void printCount(int a[][], int n, int m, int z[], int l)
{
    for (int i = 0; i < n; i++)
    {
        Map<Integer,Integer> mp = new HashMap<>();
        for (int j = 0; j < m; j++)
            mp.put(a[i][j], 1);
        int count = 0;
        for (int j = 0; j < l; j++)
        {
            if (mp.containsKey(z[j]))
                count += 1;
        }
        System.out.println("row" + (i + 1) + " = " + count);
    }
}
```

8) Java Program: Find duplicate elements in an array of numbers?

```
public static void main(String[] args) {
    int [] arr = new int [] {1, 2, 3, 4, 2, 7, 8, 8, 3};
    System.out.println("Duplicate elements in given array: ");
}
```

```

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

for(int j = i + 1; j < arr.length; j++) {

if(arr[i] == arr[j])

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

}

}

}

```

9) Write a program to read/write data from a Property file?

```

public static void main(String[] args){

Properties prop = new Properties();

prop.setProperty("db.url", "localhost");

prop.setProperty("db.user", "mkyong");

prop.setProperty("db.password", "password");

prop.store(outputStream, "");

prop.load(inputStream)

prop.getProperty("db.url");

prop.getProperty("db.user");

prop.getProperty("db.password");

prop.keySet();

prop.forEach((k, v) -> System.out.println("Key : " + k + ", Value : " + v));

}

```

10) Write a Java program to demonstrate creating factorial of a number using recursion?

```
static int factorial(int n){  
  
    if (n == 0)  
  
        return 1;  
  
    else  
  
        return (n * factorial(n-1));  
  
}  
  
public static void main(String args[]){  
  
    int i,fact=1;  
  
    int number=4;  
  
    fact = factorial(number);  
  
    System.out.println("Factorial of "+number+" is: "+fact);  
  
}  
  
}
```

11) Java Program: To print the frequency of words in a paragraph?

```
public static void frequencyWords(String strInput){  
  
    Map<String, Integer> occurrences = new HashMap<String, Integer>();  
  
    String[] words = strInput.split("\\s+");  
  
    for(String word: words){  
  
        if(occurrences.containsKey(word)){  
  
            int count = occurrences.get(word);
```

```

map.put(word,count+1);

} else {

map.put(word,1);

}

}for(Map.Entry<String,Integer> entry: map.entrySet()){

System.out.println(entry.getKey() + ":" + entry.getValue());

}

}

```

12) Write a Java program to see the output as 0,1,1,2,3?

```

public static void main(String args[])

{

int n1=0,n2=1,n3,count=5;

System.out.print(n1+" "+n2);

for(int i=2;i<count;i++)

{

n3=n1+n2;

System.out.print(" "+n3);

n1=n2;

n2=n3;

} }

```

13) Write a java program to find the factorial of a given number ?

```

public static void findFactorial(int number){

    int i,fact=1;

    for(i=1;i<=number;i++){

        fact=fact*i;

    }

    System.out.println("Factorial of " + number + " is: " + fact);

}

```

14) Write a Java program to demonstrate ArrayList?

```

class ArrayListExample {

    public static void main(String[] args)

    {

        int n = 5;

        ArrayList<Integer> arrList = new ArrayList<Integer>(n);

        for (int i = 1; i <= n; i++)

            arrList.add(i);

        System.out.println(arrList);

        arrList.remove(3);

        System.out.println(arrList);

        for (int i = 0; i < arrList.size(); i++)

            System.out.print(arrList.get(i) + " ");

    }

}

```



15) Write a Java program to demonstrate LinkedList?

```
public class LinkedListExample {  
  
    public static void main(String args[])  
  
    {  
  
        LinkedList<String> list = new LinkedList<String>();  
  
        list.add("A");  
  
        list.add("B");  
  
        list.addLast("C");  
  
        list.addFirst("D");  
  
        list.add(2, "E");  
  
        System.out.println(list);  
  
        list.remove("B");  
  
        list.remove(3);  
  
        list.removeFirst();  
  
        list.removeLast();  
  
        System.out.println(list);  
  
    }  
  
}
```

16) Write a Java program to demonstrate ArrayList using List interface?

```
class ArrayListWithListInterface {
```

```

public static void main(String[] args)
{
    List<String> arrayList = new ArrayList<String>();
    arrayList.add("Java");
    arrayList.add("C#");
    arrayList.add("Python");
    System.out.println(arrayList);
}
}

```

17) Write a Java program to demonstrate Hashset?

```

class HashSetExample{
    public static void main(String[] args){
        Set<String> hs = new HashSet<String>();
        hs.add("India");
        hs.add("Australia");
        hs.add("South Africa");
        hs.add("India");
        System.out.println(hs);
    }
}

```

18) Write a Java program to demonstrate LinkedHashSet?

```

public class LinkedHashSetExample
{
    public static void main(String[] args)
    {
        LinkedHashSet<String> linkedset = new LinkedHashSet<String>();

        linkedset.add("A");
        linkedset.add("B");
        linkedset.add("C");
        linkedset.add("D");

        System.out.println(linkedset);
    }
}

```

19) Write a Java program to demonstrate TreeSet?

```

class TreeSetExample {
    public static void main(String[] args)
    {
        TreeSet<String> ts1 = new TreeSet<String>();

        ts1.add("A");
        ts1.add("B");
        ts1.add("C");
        ts1.add("C");

        System.out.println(ts1);
    }
}

```

```
}
```

20) Write a Java program to demonstrate PriorityQueue?

```
class PriorityQueueExample {  
  
    public static void main(String args[])  
  
    {  
  
        PriorityQueue<Integer> pQueue = new PriorityQueue<Integer>();  
  
        pQueue.add(10);  
  
        pQueue.add(20);  
  
        pQueue.add(15);  
  
        System.out.println(pQueue.peek());  
  
        System.out.println(pQueue.poll());  
  
    }  
  
}
```

21) Write a Java program to demonstrate creating HashMap using Map interface?

```
class HashMapExample{  
  
    public static void main(String[] args){  
  
        Map<Integer,String> h1 = new HashMap<>();  
  
        h1.put(1,"Java");  
  
        h1.put(2,"C#");  
  
        h1.put(3,"Python");  
  
    }  
  
}
```

```
System.out.println(h1);  
  
}  
  
}
```

22) Write a Java program to demonstrate LinkedHashMap ?

```
public class LinkedHashMapExample  
{  
  
public static void main(String a[])  
{  
  
    LinkedHashMap<String, String> map = new LinkedHashMap<String, String>();  
  
    map.put("one", "www.gmail.com");  
  
    map.put("two", "www.ymail.com");  
  
    map.put("four", "www.outlook.com");  
  
    System.out.println(lhm);  
  
}  
  
}
```

23) Write a Java program to demonstrate TreeMap?

```
class TreeMapExample{  
  
public static void main(String args[]){  
  
    TreeMap<Integer,String> map=new TreeMap<Integer,String>();  
  
    map.put(100,"Amit");  
  
}
```

```

map.put(102,"Ravi");

map.put(101,"Vijay");

map.put(103,"Rahul");

for(Map.Entry m:map.entrySet()){

System.out.println(m.getKey()+" "+m.getValue());

}

}

}

```

24) Write a Java program to demonstrate Hashtable?

```

class HashtableExample{

public static void main(String args[]){

Hashtable<Integer,String> hm=new Hashtable<Integer,String>();

hm.put(100,"George");

hm.put(102,"Smith");

hm.put(101,"Sarah");

hm.put(103,"John");

for(Map.Entry m:hm.entrySet()){

System.out.println(m.getKey()+" "+m.getValue());

}

}

}

```

25) Write a Java program to demonstrate creating Multidimensional array?

```
class MultiExample {  
  
    public static void main(String[] args)  
  
    {  
  
        int[][] arr = { { 1, 2 }, { 3, 4 } };  
  
        for (int i = 0; i < 2; i++)  
  
        for (int j = 0; j < 2; j++)  
  
            System.out.println("arr[" + i + "][" + j + "] = "  
  
            + arr[i][j]);  
  
    }  
  
}
```

## Java for Testers – Interview Questions and Answers Part-14

**1) Write a Java Program to print the below output: \* 1 \* 12 \* 123 \* 1234 \* 12345 \* 123456 \* 1234567**

```
public class Patterns {  
  
    public static void main(String[] args){  
  
        int c = 2;  
  
        int num = 1;  
  
        System.out.print("*" + " " + 1);  
  
        for(int i=1;i<7;i++){  
  
            int sum = (num*10) + c;  
  
            System.out.print(" *" + " " + sum);  
  
            num = sum;
```

```
c++;
```

```
}
```

```
}
```

```
}
```

**2) Write a Java program to create an integer array `int[] a = {9,3,6,8,4,7}` and print the elements of the array in reverse?**

```
public class ReverseArray {  
  
    public static void main(String[] args){  
  
        int[] a = {9,3,6,8,4,7};  
  
        for(int i=a.length-1;i>=0;i--){  
  
            System.out.println(a[i]);  
  
        }  
  
    }  
  
}
```

**3) Write a Java program to print alternative elements in a String array `String[] a = {"One","Two","Three","Four"}` ?**

```
public class PrintAlternateElements {public static void main(String[] args){  
  
    String[] a = {"One","Two","Three","Four"};  
  
    for(int i=0;i<a.length;i=i+2){  
  
        System.out.println(a[i]);  
  
    }  
  
}
```



**4) Write a Java program to find the greatest number in an integer array  
int[] a = {9,3,6,4,8,5} ?**

```
public class MaxNumber {  
  
    public static void main(String[] args){  
  
        Integer[] a = {9,3,6,4,8,5};  
  
        int max = Collections.max(Arrays.asList(a));  
  
        System.out.println(max);  
  
    }  
  
}
```

**5) Write a Java program to find the least number in an integer array int[] a  
= {9,3,6,4,8,5} ?**

```
public class MinNumber {  
  
    public static void main(String[] args){  
  
        Integer[] a = {9,3,6,4,8,5};  
  
        int min = Collections.min(Arrays.asList(a));  
  
        System.out.println(min);  
  
    }  
  
}
```

**6) What is the predefined variable of Arrays, which can be used to find the  
size of the arrays?**

length variable can be used to find the size of the arrays.

### 7) Provide an example for using for loop with single dimensional arrays?

```
public class ForLoopInArray {  
  
    public static void main(String[] args){  
  
        Integer[] a = {9,3,6,4,8,5};  
  
        for(int i=0;i<a.length-1;i++)  
  
            System.out.println(a[i]);  
  
    }  
  
}
```

### 8) Provide an example for using for-each loop with single dimensional arrays?

```
public class ForEachLoopInArray {  
  
    public static void main(String[] args){  
  
        Integer[] a = {9,3,6,4,8,5};  
  
        for(int arr: a)  
  
            System.out.println(arr);  
  
    }  
  
}
```

### 9) What are the different access modifiers in Java and explain each of them?

Following are different access modifiers in Java:

- private: Methods and variables declared as private can only be accessed within the same class in which they are declared
- protected: Methods and variables declared as protected can only be accessed within the same package or sub classes in different package
- public: Methods and variables declared as public can be accessed from

anywhere

– default: Methods and variables which are not declared using any access modifier are accessible within the same package

### 10) How to find the length of the String without using length function?

```
public class FindLengthOfString {  
  
    public static void main(String[] args){  
  
        String s = "qascript";  
  
        int i = 0;  
  
        for(char c: s.toCharArray())  
  
        {  
  
            i++;  
  
        }  
  
        System.out.println(i);  
  
    }  
  
}
```

### 11) How to find out the part of the string from a string?

We can use the substring function to find a part of the string from a string.

Ex –

```
String str = "QASCRIPt";  
System.out.println(str.substring(2));
```

### 12) What is data binding (Early and late binding)?

Early or static binding happens at compile time while late or dynamic binding happens at run time. The method definition and method call are linked during the compile time for early binding while the method definition and method call are linked during the run time for late binding.

**13) Write a Java program to find the biggest number among 1,2,3,4,5,65,76,5,4,33,4,34,232,3,2323?**

```
public class BiggestNumber {  
  
    public static void main(String[] args){  
  
        List<Integer> numbers = new ArrayList<>();  
  
        numbers.add(1);  
  
        numbers.add(2);  
  
        numbers.add(3);  
  
        numbers.add(4);  
  
        numbers.add(5);  
  
        numbers.add(65);  
  
        numbers.add(76);  
  
        numbers.add(5);  
  
        numbers.add(4);  
  
        numbers.add(33);  
  
        numbers.add(4);  
  
        numbers.add(34);  
  
        numbers.add(232);  
  
        numbers.add(3);  
  
        numbers.add(2323);  
  
        int maxNumber = Collections.max(numbers);  
  
        System.out.println(maxNumber);  
  
    }  
  
}
```

#### 14) What is String class and its methods?

String class has many methods to perform different operations on a string. Some of the methods are:

- length()
- charAt()
- substring()
- concat()
- indexOf()
- trim()
- replace()

#### 15) Is multilevel inheritance is possible in java? Give reason.

Multilevel inheritance is not possible in java because multiple inheritance leads to ambiguity. Also dynamic loading of classes makes multiple inheritance implementation difficult.

#### 16) What Java API is required for generating PDF reports?

iText and PdfBox APIs can be used for generating PDF reports

#### 17) What's the difference between a Maven project and a Java project?

In a Java project we need to manually download and configure the jar files. But in Maven project all project dependencies can be downloaded by including them in the POM.xml file.

#### 18) Write a Java program to read and write a file?

```
public class ReadFile {  
  
    public static void main(String[] args)  
  
    {  
  
        FileReader fr;  
  
        int c=0;  
  
        try {  
  
            fr = new FileReader("src/main/resources/File.txt");
```

```
while((c=fr.read())!= -1){  
  
    System.out.print((char) c);  
  
}  
  
} catch (FileNotFoundException e) {  
  
    e.printStackTrace();  
  
} catch (IOException e) {  
  
    e.printStackTrace();  
  
}  
  
}  
  
}
```

```
public class WriteFile {  
  
    public static void main(String[] args){  
  
        try {  
  
            FileWriter fileWriter = new FileWriter("src/main/resources/File.txt");  
  
            fileWriter.write("Add new line to file");  
  
            fileWriter.close();  
  
        } catch (IOException e) {  
  
            e.printStackTrace();  
  
        }  
  
        }  
  
    }
```

**19) Why do we provide “//” in java while fetching a path of excel?**

Double backslash is provided in java to avoid compiler interpret words for system directories

**20) What is a list?**

List is an ordered collection. It is an interface that extends Collection interface.

**21) Why vector class is defined Synchronized ?**

Vector is a legacy class and was made synchronized in JDK 1.0.

**22) String x=“ABC”; String x=“ab”; does it create two objects?**

Yes. It creates two objects because string is immutable.

**23) What is String tokenizer?**

String tokenizer allows an application to break a string into tokens.

**24) What is the method to convert string to integer?**

Integer.parseInt(String) can be used to convert string to integer.

**25) What is the method to convert integer to string?**

Integer.toString() can be used to convert integer to string.