JAVA

TOP Fresher

Coding Interview Q&A Part II

SWIPE NEXT

01 WRITE A PROGRAM TO DISPLAY PRIME NUMBERS FROM 1 TO N?

```java
class Prime
{
    public static void main (String [] args)
    {
        java.util.Scanner sc=new java.util.Scanner (System.in);
        System.out.println ("enter number");
        int n=sc.nextInt ();
        System.out.println ("Prime numbers between 1 and " + n);
//loop through the numbers one by one
        for (int i=1; i < n; i++)
        {
            boolean isPrime = true;
//check to see if the number is prime
            for (int j=2; j < i ; j++)
            {
                if (i % j == 0)
                {
                    isPrime = false;
                    break;
                }
            }
            if (isPrime)
                System.out.print (i + " ");
        }
    }
}
/*
OUTPUT:
enter number
25
Prime numbers between 1 and 25
        1 2 3 5 7 11 13 17 19 23   */
```

02 WRITE A PROGRAM TO FIND SUM OF PRIME NUMBERS?

Main.java | SumofPrime.java

```java
import java.util.Scanner;
public class SumofPrime
{
    public static void main(String[] args)
    {
        Scanner scn=new Scanner(System.in);
        System.out.println("Enter the range to print sum of prime Nos.....");
        int range=scn.nextInt();
        int sum=0;
        for(int i=1;i<=range ;i++)
        {
            if(isPrime(i))
                sum=sum+i;
        }
        System.out.println(sum);
    }
    1 usage
    public static boolean isPrime(int num)
    {
        if(num==1) return false;
        for(int i=2;i<num ;i++)
        {
            if(num%i==0)
            {
                return false;
            }
        }
        return true;
    }
}
/
OUTPUT: Enter the range to print sum of prime Nos.....10  17  */
```

03 WRITE A PROGRAM TO DISPLAY MULTIPLICATION TABLE?

```java
class Multiplication
{
    public static void main(String[] args)
    {
        java.util.Scanner sc=new java.util.Scanner(System.in);
        System.out.println("enter value of n");
        int n=sc.nextInt();
        for(int i=1;i<=10;i++)
        {
            System.out.println(n+"*"+i+"="+(n*i));
        }
    }
}
/*
Output:
enter value of n: 2
        2*1=2
        2*2=4
        2*3=6
        2*4=8
        2*5=10
        2*6=12
        2*7=14
        2*8=16
        2*9=18
        2*10=20 */
```

04 WRITE A PROGRAM TO SWAP TWO NUMBERS WITHOUT USING 3RD VARIABLE?

```java
class Swap
{
    public static void main(String[] args) {
        int i=10;
        int j=20;
        i=i + j;
        j=i-j;
        i=i-j;
        System.out.println("i="+i);
        System.out.println("j="+j);
    }
}
// OUTPUT: i=20 j=10
```

05 WRITE A PROGRAM TO CONVERT DECIMAL TO BINARY?

Main.java | Dectobin.java

```java
import java.util.*;
public class Dectobin
{
    public static void main(String[] args)
    {
        System.out.println("enter the decimal number");
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        String bin="";
        while(n>0)
        {
            int r=n%2;
            bin= r + bin;
            n=n/2;
        }
        System.out.println("Binary Equivalent:" + bin);
    }
}
/*
OUTPUT:
enter the decimal number
3855
Binary Equivalent:111100001111 */
```

06 WRITE A PROGRAM TO INSERT THE ELEMENTS IN AN ARRAY?

Main.java | InstSingArray.java

```java
import java.util.Scanner;
public class InstSingArray
{
    public static void main (String [ ] args)
    {
        Scanner sc= new Scanner (System.in);
        System.out.println ("enter the size");
        int length= sc.nextInt ();
        int arr [ ] =new int [length];
        System.out.println ("enter the "+length+" elements");
        for (int i = 0; i < arr.length; i++)
        {
            arr[i] =sc.nextInt ();
        }
        for (int i = 0; i < arr.length; i++)
        {
            System.out.println ("arr ["+i+"] ---->"+arr[i]);
        }
    }
} /*
Output: enter the size
5
Enter the 5 elements
2 3 5 8 64
arr [0] ----->2
arr [1] ----->3
arr [2] ----->5
arr [3] ----->8
arr [4] ---->64 */
```

CORE JAVA

INTERVIEW QUESTIONS

YOU'LL MOST LIKELY BE ASKED

JAVA COLLECTIONS

Cheat Sheet

List	Add	Remove	Get	Contains	Next	Data Structure
ArrayList	O(1)	O(n)	O(1)	O(n)	O(1)	Array
LinkedList	O(1)	O(1)	O(n)	O(n)	O(1)	Linked List
CopyOnWriteArrayList	O(n)	O(n)	O(1)	O(n)	O(1)	Array

Set	Add	Remove	Contains	Next	Size	Data Structure
HashSet	O(1)	O(1)	O(1)	O(h/n)	O(1)	Hash Table
LinkedHashSet	O(1)	O(1)	O(1)	O(1)	O(1)	Hash Table + Linked List
EnumSet	O(1)	O(1)	O(1)	O(1)	O(1)	Bit Vector
TreeSet	O(log n)	O(log n)	O(log n)	O(log n)	O(1)	Redblack tree
CopyOnWriteArraySet	O(n)	O(n)	O(n)	O(1)	O(1)	Array
ConcurrentSkipListSet	O(log n)	O(log n)	O(log n)	O(1)	O(n)	Skip List

Map	Put	Remove	Get	ContainsKey	Next	Data Structure
HashMap	O(1)	O(1)	O(1)	O(1)	O(h / n)	Hash Table
LinkedHashMap	O(1)	O(1)	O(1)	O(1)	O(1)	Hash Table + Linked List
IdentityHashMap	O(1)	O(1)	O(1)	O(1)	O(h / n)	Array
WeakHashMap	O(1)	O(1)	O(1)	O(1)	O(h / n)	Hash Table
EnumMap	O(1)	O(1)	O(1)	O(1)	O(1)	Array
TreeMap	O(log n)	O(log n)	O(log n)	O(log n)	O(log n)	Redblack tree
ConcurrentHashMap	O(1)	O(1)	O(1)	O(1)	O(h / n)	Hash Tables
ConcurrentSkipListMap	O(log n)	O(log n)	O(log n)	O(log n)	O(1)	Skip List

Queue	Offer	Peak	Poll	Remove	Size	Data Structure
PriorityQueue	O(log n)	O(1)	O(log n)	O(n)	O(1)	Priority Heap
LinkedList	O(1)	O(1)	O(1)	O(1)	O(1)	Array
ArrayDequeue	O(1)	O(1)	O(1)	O(n)	O(1)	Linked List
ConcurrentLinkedQueue	O(1)	O(1)	O(1)	O(n)	O(1)	Linked List
ArrayBlockingQueue	O(1)	O(1)	O(1)	O(n)	O(1)	Array
PrioriritiyBlockingQueue	O(log n)	O(1)	O(log n)	O(n)	O(1)	Priority Heap
SynchronousQueue	O(1)	O(1)	O(1)	O(n)	O(1)	None
DelayQueue	O(log n)	O(1)	O(log n)	O(n)	O(1)	Priority Heap
LinkedBlockingQueue	O(1)	O(1)	O(1)	O(n)	O(1)	Linked List

6) Difference between abstract class and interface ?

Interface	Abstract Class
1) Interface contains only abstract methods	1) Abstract class can contain abstract methods, concrete methods or both
2) Access Specifiers for methods in interface must be public	2) Except private we can have any access specifier for methods in abstract class.
3) Variables defined must be public , static , final	3) Except private variables can have any access specifiers
4) Multiple Inheritance in java is implemented using interface	4)We cannot achieve multiple inheritance using abstract class.
5) To implement an interface we use implements keyword	5)To implement an interface we use implements keyword

7) Why java is platform independent?

The most unique feature of java is platform independent. In any programming language soruce code is compiled in to executable code . This cannot be run across all platforms. When javac compiles a java program it generates an executable file called .class file.
class file contains byte codes. Byte codes are interpreted only by JVM's . Since these JVM's are made available across all platforms by Sun Microsystems, we can execute this byte code in any platform. Byte code generated in windows environment can also be executed in linux environment. This makes java platform independent.

8) What is method overloading in java ?

A class having two or more methods with same name but with different arguments then we say that those methods are overloaded. Static polymorphism is achieved in java using method overloading.
Method overloading is used when we want the methods to perform similar tasks but with different inputs or values. When an overloaded method is invoked java first checks the method name, and the number of arguments ,type of arguments; based on this compiler executes this method.
Compiler decides which method to call at compile time. By using overloading static polymorphism or static binding can be achieved in java.
Note : Return type is not part of method signature. we may have methods with different return types but return type alone is not sufficient to call a method in java.

9) What is difference between c++ and Java ?

Java	C++
1) Java is platform independent	C++ is platform dependent.
2) There are no pointers in java	There are pointers in C++.
3) There is no operator overloading in java	C ++ has operator overloading.
4) There is garbage collection in java	There is no garbage collection
5) Supports multithreading	Does'nt support multithreading
6) There are no templates in java	There are templates in java
7) There are no global variables in java	There are global variables in c++

10) What is JIT compiler ?

JIT compiler stands for Just in time compiler. JIT compiler compiles byte code in to executable code . JIT a part of JVM .JIT cannot convert complete java program in to executable code it converts as and when it is needed during execution.

11) What is bytecode in java ?

When a javac compiler compiler compiles a class it generates .class file. This .class file contains set of instructions called byte code. Byte code is a machine independent language and contains set of instructions which are to be executed only by JVM. JVM can understand this byte codes.

1) what are static blocks and static initializers in Java ?

Static blocks or static initializers are used to initialize static fields in java. we declare static blocks when we want to intialize static fields in our class. Static blocks gets executed exactly once when the class is loaded . Static blocks are executed even before the constructors are executed.

2) How to call one constructor from the other constructor ?

With in the same class if we want to call one constructor from other we use this() method. Based on the number of parameters we pass appropriate this() method is called.
Restrictions for using this method :
1) this must be the first statement in the constructor
2)we cannot use two this() methods in the constructor

3) What is method overriding in java ?

If we have methods with same signature (same name, same signature, same return type) in super class and subclass then we say
subclass method is overridden by superclass.
When to use overriding in java
If we want same method with different behaviour in superclass and subclass then we go for overriding.
When we call overridden method with subclass reference subclass method is called hiding the superclass method.

4) What is super keyword in java ?

Variables and methods of super class can be overridden in subclass . In case of overriding , a subclass object call its own variables and methods. Subclass cannot access the variables and methods of superclass because the overridden variables or methods hides the methods and variables of super class. But still java provides a way to access super class members even if its members are overridden. Super is used to access superclass variables, methods, constructors.
Super can be used in two forms :
1) First form is for calling super class constructor.
2) Second one is to call super class variables,methods.
Super if present must be the first statement.

5) Difference between method overloading and method overriding in java ?

Method Overloading	Method Overriding
1) Method Overloading occurs with in the same class	Method Overriding occurs between two classes superclass and subclass
2) Since it involves with only one class inheritance is not involved.	Since method overriding occurs between superclass and subclass inheritance is involved.
3)In overloading return type need not be the same	3) In overriding return type must be same.
4) Parameters must be different when we do overloading	4) Parameters must be same.
5) Static polymorphism can be acheived using method overloading	5) Dynamic polymorphism can be acheived using method overriding.
6) In overloading one method can't hide the another	6) In overriding subclass method hides that of the superclass method.

Overloading

```
class Dog{
    public void bark(){
        System.out.println("woof ");
    }
    //overloading method
    public void bark(int num){
        for(int i=0; i<num; i++)
            System.out.println("woof ");
    }
}
```

Same Method Name, Different Parameter

Overriding

```
class Dog{
    public void bark(){
        System.out.println("woof ");
    }
}
class Hound extends Dog{
    public void sniff(){
        System.out.println("sniff ");
    }

    public void bark(){
        System.out.println("bowl");
    }
}
```

Same Method Name, Same parameter

12) Difference between this() and super() in java ?

this() is used to access one constructor from another with in the same class while super() is used to access superclass constructor. Either this() or super() exists it must be the first statement in the constructor.

13) What is a class ?

Classes are fundamental or basic unit in Object Oriented Programming .A class is kind of blueprint or template for objects. Class defines variables, methods. A class tells what type of objects we are creating. For example take Department class tells us we can create department type objects. We can create any number of department objects.
All programming constructs in java reside in class. When JVM starts running it first looks for the class when we compile. Every Java application must have atleast one class and one main method.
Class starts with class keyword. A class definition must be saved in class file that has same as class name. File name must end with .java extension.

```
public class FirstClass
{public static void main(String[] args)
{System.out.println("My First class");
}
}
```

If we see the above class when we compile JVM loads the FirstClass and generates a .class file(FirstClass.class). When we run the program we are running the class and then executes the main method.

14) What is an object ?

An Object is instance of class. A class defines type of object. Each object belongs to some class.Every object contains state and behavior. State is determined by value of attributes and behavior is called method. Objects are alos called as an instance.
To instantiate the class we declare with the class type.

```
public classFirstClass {public static voidmain(String[] args)
{
FirstClass f=new FirstClass();
System.out.println("My First class");
}

}
```

To instantiate the FirstClass we use this statement
FirstClass f=new FirstClass();
f is used to refer FirstClass object.

15)What is method in java ?

It contains the executable body that can be applied to the specific object of the class.
Method includes method name, parameters or arguments and return type and a body of executable code.

```
Syntax : type methodName(Argument List){
}
```

ex : public float add(int a, int b, int c)
methods can have multiple arguments. Separate with commas when we have multiple arguments.

16) What is encapsulation ?

The process of wrapping or putting up of data in to a single unit class and keeps data safe from misuse is called encapsulation .

```
}

}
```

Since a is integer object it returns true.
There will be a compile time check whether reference expression is subtype of destination type. If it is not a subtype then compile time error will be shown as Incompatible types

36) What does null mean in java?

When a reference variable doesn't point to any value it is assigned null.
Example : Employee employee;
In the above example employee object is not instantiate so it is pointed no where

37) Can we have multiple classes in single file ?

Yes we can have multiple classes in single file but it people rarely do that and not recommended. We can have multiple classes in File but only one class can be made public. If we try to make two classes in File public we get following compilation error.

```
"The public type must be defined in its own file".
```

38) What all access modifiers are allowed for top class ?

For top level class only two access modifiers are allowed. public and default. If a class is declared as public it is visible everywhere.
If a class is declared default it is visible only in same package.
If we try to give private and protected as access modifier to class we get the below compilation error.

```
Illegal Modifier for the class only public,abstract and final are permitted.
```

39) What are packages in java?

Package is a mechanism to group related classes ,interfaces and enums in to a single module.
Package can be declared using the following statement :
Syntax : package <package-name>
Coding Convention : package name should be declared in small letters.
package statement defines the namespace.
The main use of package is

1) To resolve naming conflicts
2) For visibility control : We can define classes and interfaces that are not accessible outside the class.

40) Can we have more than one package statement in source file ?

We can't have more than one package statement in source file. In any java program there can be atmost only 1 package statement. We will get compilation error if we have more than one package statement in source file.

41) Can we define package statement after import statement in java?

We can't define package statement after import statement in java. package statement must be the first statement in source file. We can have comments before the package statement.

42) What are identifiers in java?

Identifiers are names in java program. Identifiers can be class name, method name or variable name.
Rules for defining identifiers in java:

1) Identifiers must start with letter,Underscore or dollar($) sign.
2) Identifiers can't start with numbers .
3) There is no limit on number of characters in identifier but not recommended to have more than 15 characters
4) Java identifiers are case sensitive.
5) First letter can be alphabet ,or underscore and dollar sign. From second letter we can have numbers .
6) We should'nt use reserve words for identifiers in java.

43) What are access modifiers in java?

The important feature of encapsulation is access control. By preventing access control we can misuse of class, methods and members.

A class, method or variable can be accessed is determined by the access modifier. There are three types of access modifiers in java. public,private,protected. If no access modifier is specified then it has a default access.

44) What is the difference between access specifiers and access modifiers in java?

In C++ we have access specifiers as public,private,protected and default and access modifiers as static, final. But there is no such divison of access specifiers and access modifiers in java. In Java we have access modifiers and non access modifiers.
Access Modifiers : public, private, protected, default
Non Access Modifiers : abstract, final, stricfp.

45) What access modifiers can be used for class ?

We can use only two access modifiers for class public and default.
public: A class with public modifier can be visible
1) In the same class
2) In the same package subclass
3) In the same package nonsubclass
4) In the different package subclass
5) In the different package non subclass.

default : A class with default modifier can be accesed
1) In the same class
2) In the same package subclass
3) In the same package nonsubclass
4) In the different package subclass
5) In the different package non subclass.

46) Explain what access modifiers can be used for methods?

We can use all access modifiers public, private,protected and default for methods.
public : When a method is declared as public it can be accessed
6) In the same class
7) In the same package subclass
8) In the same package nonsubclass
9) In the different package subclass
10) In the different package non subclass.

default : When a method is declared as default, we can access that method in
1) In the same class
2) In the same package subclass
3) In the same package non subclass
We cannot access default access method in
1) Different package subclass
2) Different package non subclass.

protected : When a method is declared as protected it can be accessed
1) With in the same class
2) With in the same package subclass
3) With in the same package non subclass
4) With in different package subclass
It cannot be accessed non subclass in different package.

private : When a method is declared as private it can be accessed only in that class.
It cannot be accessed in
1) Same package subclass
2) Same package non subclass
3) Different package subclass
4) Different package non subclass.

47) Explain what access modifiers can be used for variables?

We can use all access modifiers public, private,protected and default for variables.
public : When a variables is declared as public it can be accessed
1) In the same class

2) In the same package subclass
3) In the same package nonsubclass
4) In the different package subclass
5) In the different package non subclass.

default : When a variables is declared as default, we can access that method in
1) In the same class
2) In the same package subclass
3) In the same package non subclass
We cannot access default access variables in
4) Different package subclass
5) Different package non subclass.

protected : When a variables is declared as protected it can be accessed
1) With in the same class
2) With in the same package subclass
3) With in the same package non subclass
4) With in different package subclass
It cannot be accessed non subclass in different package.

private : When a variables is declared as private it can be accessed only in that class.
It cannot be accessed in
1) Same package subclass
2) Same package non subclass
3) Different package subclass
4) Different package non subclass.

48) What is final access modifier in java?

final access modifier can be used for class, method and variables. The main advantage of final access modifier is security no one can modify our classes, variables and methods. The main disadvantage of final access modifier is we cannot implement oops concepts in java. Ex : Inheritance, polymorphism.
final class : A final class cannot be extended or subclassed. We ar e preventing inheritance by marking a class as final. But we can still access the methods of this class by composition. Ex: String class
final methods: Method overriding is one of the important features in java. But there are situations where we may not want to use this feature. Then we declared method as final which will print overriding. To allow a method from being overridden we use final access modifier for methods.
final variables : If a variable is declared as final ,it behaves like a constant . We cannot modify the value of final variable. Any attempt to modify the final variable results in compilation error. The error is as follows
"final variable cannot be assigned."

49) Explain about abstract classes in java?

Sometimes we may come across a situation where we cannot provide implementation to all the methods in a class. We want to leave the implementation to a class that extends it. In such case we declare a class as abstract.To make a class abstract we use key word abstract. Any class that contains one or more abstract methods is declared as abstract. If we don't declare class as abstract which contains abstract methods we get compile time error. We get the following error.
"The type <class-name> must be an abstract class to define abstract methods."
Signature ; abstract class <class-name>
{
}
For example if we take a vehicle class we cannot provide implementation to it because there may be two wheelers , four wheelers etc. At that moment we make vehicle class abstract. All the common features of vehicles are declared as abstract methods in vehicle class. Any class which extends vehicle will provide its method implementation. It's the responsibility of subclass to provide implementation.
The important features of abstract classes are :
1) Abstract classes cannot be instantiated.
2) An abstract classes contains abstract methods, concrete methods or both.
3) Any class which extends abstract class must override all methods of abstract class.
4) An abstract class can contain either 0 or more abstract methods.

is the reason, if we see C language we can write c language only in English we can't write in other languages because it uses ASCII code.

22) What is Unicode ?

Unicode is a character set developed by Unicode Consortium. To support all languages in the world Java supports Unicode values. Unicode characters were represented by 16 bits and its character range is 0-65,535.
Java uses ASCII code for all input elements except for Strings,identifiers, and comments. If we want to use telugu we can use telugu characters for identifiers.We can enter comments in telugu.

23) Difference between Character Constant and String Constant in java ?

Character constant is enclosed in single quotes. String constants are enclosed in double quotes. Character constants are single digit or character. String Constants are collection of characters.
Ex :'2', 'A'
Ex : "Hello World"

24) What are constants and how to create constants in java?

Constants are fixed values whose values cannot be changed during the execution of program. We create constants in java using final keyword.
Ex : final int number =10;
final String str="java-interview –questions"

25) Difference between '>>' and '>>>' operators in java?

>> is a right shift operator shifts all of the bits in a value to the right to a specified number of times.
int a =15;
a= a >> 3;
The above line of code moves 15 three characters right.
>>> is an unsigned shift operator used to shift right. The places which were vacated by shift are filled with zeroes.

Core java Interview questions on Coding Standards

26) Explain Java Coding Standards for classes or Java coding conventions for classes?

Sun has created Java Coding standards or Java Coding Conventions . It is recommended highly to follow java coding standards.
Classnames should start with uppercase letter. Classnames names should be nouns. If Class name is of multiple words then the first letter of inner word must be capital letter.
Ex : Employee, EmployeeDetails, ArrayList, TreeSet, HashSet

27) Explain Java Coding standards for interfaces?

1) Interface should start with uppercase letters
2) Interfaces names should be adjectives

Example : Runnable, Serializable, Marker, Cloneable

28) Explain Java Coding standards for Methods?

1) Method names should start with small letters.
2) Method names are usually verbs
3) If method contains multiple words, every inner word should start with uppercase letter.

Ex : toString()

4) Method name must be combination of verb and noun

Ex : getCarName(),getCarNumber()

29) Explain Java Coding Standards for variables ?

1) Variable names should start with small letters.
2) Variable names should be nouns
3) Short meaningful names are recommended.
4) If there are multiple words every innerword should start with Uppecase character.

Ex : string,value,empName,empSalary

30) Explain Java Coding Standards for Constants?

Constants in java are created using static and final keywords.
1) Constants contains only uppercase letters.
2) If constant name is combination of two words it should be separated by underscore.

Through encapsulation we can hide and protect the data stored in java objects.Java supports encapsulation through access control. There are four access control modifiers in java public , private ,protected and default level.
For example take a car class , In car we have many parts which is not required for driver to know what all it consists inside. He is required to know only about how to start and stop the car. So we can expose what all are required and hide the rest by using encapsulation.

17) Why main() method is public, static and void in java ?

public : "public" is an access specifier which can be used outside the class. When main method is declared public it means it can be used outside class.
static : To call a method we require object. Sometimes it may be required to call a method without the help of object. Then we declare that method as static. JVM calls the main() method without creating object by declaring keyword static.
void : void return type is used when a method does'nt return any value . main() method does'nt return any value, so main() is declared as void.

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

18) Explain about main() method in java ?

Main() method is starting point of execution for all java applications.

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

String args[] are array of string objects we need to pass from command line arguments.
Every Java application must have atleast one main method.

19)What is constructor in java ?

A constructor is a special method used to initialize objects in java.

we use constructors to initialize all variables in the class when an object is created. As and when an object is created it is initialized automatically with the help of constructor in java.
We have two types of constructors
Default Constructor
Parameterized Constructor
Signature : public classname()
{
}

Signature : public classname(parameters list)
{
}

20) What is difference between length and length() method in java ?

length() : In String class we have length() method which is used to return the number of characters in string.
Ex : String str = "Hello World";
System.out.println(str.length());
Str.length() will return 11 characters including space.

length : we have length instance variable in arrays which will return the number of values or objects in array.
For example :
String days[]={" Sun","Mon","wed","thu","fri","sat"};
Will return 6 since the number of values in days array is 6.

21) What is ASCII Code?

ASCII stands for American Standard code for Information Interchange. ASCII character range is 0 to 255. We can't add more characters to the ASCII Character set. ASCII character set supports only English. That

3) Constant names are usually nouns.
Ex:MAX_VALUE, MIN_VALUE, MAX_PRIORITY, MIN_PRIORITY

31) Difference between overriding and overloading in java?

Overriding	Overloading
In overriding method names must be same	In overloading method names must be same
Argument List must be same	Argument list must be different atleast order of arguments.
Return type can be same or we can return covariant type. From 1.5 covariant types are allowed	Return type can be different in overloading.
We cant increase the level of checked exceptions. No restrictions for unchecked exceptions	In overloading different exceptions can be thrown.
A method can only be overridden in subclass	A method can be overloaded in same class or subclass
Private,static and final variables cannot be overridden.	Private , static and final variables can be overloaded.
In overriding which method is called is decided at runtime based on the type of object referenced at run time	In overloading which method to call is decided at compile time based on reference type.
Overriding is also known as Runtime polymorphism, dynamic polymorphism or late binding	Overloading is also known as Compile time polymorphism, static polymorphism or early binding.

32) What is 'IS-A ' relationship in java?

'is a' relationship is also known as inheritance. We can implement 'is a' relationship or inheritance in java using extends keyword. The advantage or inheritance or is a relationship is reusability of code instead of duplicating the code.
Ex : Motor cycle is a vehicle
Car is a vehicle Both car and motorcycle extends vehicle.

33) What is 'HAS A'' relationship in java?

'Has a ' relationship is also known as "composition or Aggregation". As in inheritance we have 'extends' keyword we don't have any keyword to implement 'Has a' relationship in java. The main advantage of 'Has-A' relationship in java code reusability.

34) Difference between 'IS-A' and 'HAS-A' relationship in java?

IS-A relationship	HAS- A RELATIONSHIP
Is a relationship also known as inheritance	Has a relationship also known as composition or aggregation.
For IS-A relationship we uses extends keyword	For Has a relationship we use new keyword
Ex : Car is a vehicle.	Ex : Car has an engine. We cannot say Car is an engine
The main advantage of inheritance is reusability of code	The main advantage of has a relationship is reusability of code.

35) Explain about instanceof operator in java?

Instanceof operator is used to test the object is of which type.
Syntax : <reference expression> instanceof <destination type>
Instanceof returns true if reference expression is subtype of destination type.
Instanceof returns false if reference expression is null.

```
Example : public classInstanceOfExample {public static voidmain(String[] args) {Integer a =
newInteger(5);if (a instanceof java.lang.Integer) {
System.out.println(true);
} else {
System.out.println(false);
}
```

3. Fibonacci Series –

In Fibonacci series, next number is the sum of previous two numbers
Input = First 10 Numbers
Output = 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55 etc.
The first two numbers of Fibonacci series are 0 and 1.

```
public static void main(String[] args) {
int num1 = 0, num2 = 1, num=10;
for (int i = 0; i <= num; i++) {
System.out.print(num1 + " ");
int num3 = num2 + num1;// Swap
num1 = num2;
num2 = num3;
}
}
```

---

4. Reverse a numbers and Number is Palindrome or Not.

Input = 12321
Output =12321

```
public static void main(String[] args) {
int num = 12321;
// 1. Reverse a Number Using the While Loop reversed number
int rev = 0;
int temp = num;
int rem; // remainder
while (num > 0) {
rem = num % 10;
rev = (rev * 10) + rem;
num = num / 10;
}
System.out.println("Reversed Number is " + rev);
// Verify number is palindrome or not
if (rev == temp) {
System.out.println("palindrome number ");
} else {
System.out.println("not palindrome");
}
}
```

---

5. Factorial Number

Factorial Program in Java: Factorial of n is the product of all positive descending integers.

Input = 5!
Output = 5! = 5*4*3*2*1 = 120

Frequently Asked Java Programs for QA

Top 10 numbers Questions

1. Swap two numbers

Input: a = 100, b= 200;
Output: a = 200, b= 100;

```
public static void main(String[] args) {
int a = 100, b = 200;
System.out.println("After swapping, a = " + a + " and b = " + b);
// 1. Swapping using three Variables
int temp = a;
a = b;
b = temp;
System.out.println("After swapping, a = " + a + " and b = " + b);
// 2. Using Two Variables
a = a + b;
b = a - b;
a = a - b;
System.out.println("After swapping, a = " + a + " and b = " + b);
// 3. Swapping a and b using XOR
a = a ^ b;
b = a ^ b;
a = a ^ b;
System.out.println("After swapping, a = " + a + " and b = " + b);
}
```

---

2. Armstrong number -

Armstrong number is a number that is equal to the sum of cubes of its digits.

Input: 153 , **Output:** Yes
153 is an Armstrong number. ==> (1*1*1) + (5*5*5) + (3*3*3) = 153

```
public static void main(String[] args) {
int sum = 0, res, temp;
int num = 153;// It is the number to check Armstrong
temp = num;
while (num > 0) {
res = num % 10;
num = num / 10;
sum = sum + (res * res * res);
}
if (temp == sum)
System.out.println(temp + " is armstrong number");
else
System.out.println(temp + " is Not armstrong number");
}
```

```
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
System.out.println("Enter number which you want for Factorial: ");
int num = sc.nextInt();
int fact = 1;
for (int i = 1; i <= num; i++) {
fact = fact * i;
}
System.out.println("Factorial of" + num + " is " + fact);
}
```

---

6. OddEvenNumbers

Input = 11
Output = Given number is odd number

```
public static void main(String[] args) {
// 1. Using Brute Forcew Approach
Scanner sc = new Scanner(System.in);
System.out.println("Enter Number:-");
int num = sc.nextInt();
if (num % 2 == 0)// Brute Forcew Approach
{
System.out.println("Given is even number");
} else {
System.out.println("Given number is odd number");
}
```

---

7. Prime Number

Prime number is a number that is greater than 1 and divided by 1 or itself only.

Input = 31, Output = The number is prime.

```
public static void main(String[] args) {
int num = 31;
int count = 0;
if (num <= 1) {
System.out.println("The number is not prime");
return;
}
for (int i = 2; i <= num / 2; i++) {
if (num % i == 0)
count++;
}
if (count > 1) {
System.out.println("The number is not prime");
} else {
System.out.println("The number is prime"); }
```

```
num = num / 10;
count++;
}
System.out.println("Number of digits : " + count);
// 2. Converting given number to string solution to count digits in an integer
String result = Integer.toString(num2); // calculate the size of string
System.out.println(+result.length());
}
```

Top 15 String Questions

1. Reverse a string

Input = mama
Output = mama

```
public static void main(String[] args) {
String str = "mama";
String s2 = "";
// 1. by using the charAt() method
    for (int i = str.length() - 1; i >= 0; i--) {
    s2 = s2 + str.charAt(i);// extracts each character and store in string
    }
System.out.println("Reversed word: " + s2);
// below is code to check weather given string is Palindrome or not
if (str.equalsIgnoreCase(s2)) {
System.out.println("String is Palindrome");
} else {
System.out.println("String is not Palindrome");
}
}
// 2. Using built in reverse() method of the StringBuilder class:
String input = "Welcome To Jave Learning";
StringBuilder input1 = new StringBuilder();
input1.append(input); // append a string into StringBuilder input1
input1.reverse();
System.out.println(input1);

// 3. Using StringBuffer:
String strText = "Java Learning";
// conversion from String object to StringBuffer
StringBuffer sbr = new StringBuffer(strText);
sbr.reverse();
System.out.println(sbr);
```

---

8. Largest number from 3 number/ given list

```
public static void main(String[] args) {
// TODO Auto-generated method stub
// 1. By using if else condition
int num1 = 7, num2 = 9, num3 = 10;
if( num1 >= num2 && num1 >= num3)
System.out.println(num1 + " is the largest number.");
else if (num2 >= num1 && num2 >= num3)
System.out.println(num2 + " is the largest number.");
else
System.out.println(num3 + " is the largest number.");

// 2. Using Collections.max() method and ArrayList
ArrayList<Integer> x = new ArrayList<>();
x.add(12);
x.add(22);
x.add(54);
System.out.println(Collections.max(x)+ " is the largest number.");
}
```

---

9. Sum of Digits

Sum of all given numbers.

Input = 987

Output = 24

```
public static void main(String[] args) {
int n = 987;
int sum = 0;
while (n != 0) {
sum = sum + n % 10;
n = n / 10;
}
System.out.println("Using While:- " + sum);
}
```

---

10. Count digits in an integer number

Input = 29845315, **Output = 8**

```
public static void main(String[] args) {
// TODO Auto-generated method stub
long num = 29845315;
int count = 0, num2 = 298453;
// 1. by using while loop
while (num != 0) {
```

2. Remove space form given string

Input String = “hello java   Learning ”
Output String = “hellojavaLearning”

```
public static void main(String[] args) {
System.out.println("Enter String ");
Scanner sc = new Scanner(System.in);
String input = sc.nextLine();
System.out.println("Original String- " + input);
input = input.replaceAll("\\s", "");
System.out.println("Final String- " + input);
}
```

---

3. Finding Common Elements in Arrays

Input =
array1 = { 4, 2, 3, 1, 6 }; array2 = { 6, 7, 8, 4 };
Output = 6,4

```
// By using the for loop

Integer[] array1 = { 4, 2, 3, 1, 6 };
Integer[] array2 = { 6, 7, 8, 4 };
List<Integer> commonElements = new ArrayList<>();
for (int i = 0; i < array1.length; i++) {
for (int j = 0; j < array2.length; j++) {
if (array1[i] == array2[j]) {
commonElements.add(array1[i]);
} }}
System.out.println("Common Elements for given two array List is:" +
commonElements);

// by using ArrayList with retainAll method
ArrayList<Integer> list1 = new ArrayList<>(Arrays.asList(array1));
ArrayList<Integer> list2 = new ArrayList<>(Arrays.asList(array2));
list1.retainAll(list2);
System.out.println("Common Elements:" + list1);

// By using Java Stream
String[] array3 = { "Java", "JavaScript", "C", "C++" };
String[] array4 = { "Python", "C#", "Java", "C++" };
ArrayList<String> list3 = new ArrayList<>(Arrays.asList(array3));
ArrayList<String> list4 = new ArrayList<>(Arrays.asList(array4));
List<String> commonElements1 =
list3.stream().filter(list4::contains).collect(Collectors.toList());
System.out.println(commonElements1);
} }
```

---

4. Find first and last element of ArrayList in java

Input = array1 = { 4, 2, 3, 1, 6 };
Output = First is:4, Last is: 6

```
ArrayList<Integer> list = new ArrayList<Integer>(5);
// find first element
 int first = list.get(0);//First Element
 // find last element
 int last = list.get(list.size() - 1);//last Element
```

---

5. Second Largest and Second Smallest Numbers:

```
// Code to find second largest and second smallest numbers in an array
int[] arrayList = { 4, 2, 3, 1,0, 6,12,15,20 };
int num=arrayList.length;
Arrays.sort(arrayList);
System.out.println("Second Largest element is "+arrayList[num-2]); //Display Second Smallest
System.out.println("Second Smallest element is "+arrayList[1]);
```

---

6. How to sort an Array without using inbuilt method?

Input = array[] = { 10, 5, 20, 63, 12, 57, 88, 60 };

Output = 5 10 12 20 57 60 63 88

```
int temp, size;
int array[] = { 10, 5, 20, 63, 12, 57, 88, 60 };
size = array.length;
for (int i = 0; i < size; i++) {
for (int j = i + 1; j < size; j++) {
if (array[i] > array[j]) {
temp = array[i];
array[i] = array[j];
array[j] = temp;
}}}
for (int i = 0; i < array.length; i++) {
System.out.println("Array sorted: " + array[i]);
}
// Print 3rd Largest number from an Array
System.out.println("Third largest number is:: " + array[size - 3]);
System.out.println("**********");

// sort array using the Arrays.sort method
Arrays.sort(array);
System.out.println("sorted array- " + Arrays.toString(array));
int thirdMaxNum=array[size-3];
System.out.println("Third highest array- " +thirdMaxNum );
```

---

10. count the occurrences of each character?

Input = "This is an example";

Output = p = 1, a = 2, s = 2, T = 1, e = 2, h = 1, x = 1, i = 2, l = 1, m = 1, n = 1

```
public static void main(String[] args) {
String str = "This is an example";
HashMap<Character, Integer> count = new HashMap<Character, Integer>();
// convert string to character array
char[] arr = str.toCharArray();
// traverse every character and count the Occurrences
for (char c : arr) {
// filter out white spaces
if (c != ' ') {
if (count.containsKey(c)) {
// if character already traversed, increment it
count.put(c, count.get(c) + 1);
} else {
// if character not traversed, add it to hashmap
count.put(c, 1);
}
}
}
// traverse the map and print the number of occurences of a character
for (Map.Entry entry : count.entrySet()) {
System.out.print( entry.getKey() + " = " + entry.getValue()+", ");
}
}
```

---

11. Removing Duplicates from an Array

```
// using for loop
String[] strArray = {"abc", "def", "abc", "mno", "xyz", "pqr", "xyz", "pqr"};
//1. Using Brute Force Method
for (int i = 0; i < strArray.length-1; i++)
{
for (int j = i+1; j < strArray.length; j++)
{
if( (strArray[i]==(strArray[j])) )
{
System.out.println("Brute Force Method : Duplicate Element is : "+strArray[j]);
}}}
// using Hashset
HashSet<String> hs = new HashSet<String>();
for (String arrayElement : strArray)
{
if(!hs.add(arrayElement))
{System.out.println("HashSet :Duplicate Element is : "+arrayElement);
}}
```

---

7. Counting number of occurrences of given word in a string using Java?

String = "Java is a programming language. Java is widely used in software Testing";

Input = ”Java”, **Output** = 2

```
public static void main(String[] args) {
String string = "Java is a programming language. Java is widely used in software
Testing";
String[] words = string.toLowerCase().split(" ");
String word = "java";
int occurrences = 0;
for (int i = 0; i < words.length; i++)
if (words[i].equals(word))
occurrences++;
System.out.println(occurrences);
}
```

---

8. Find each word occurrence from given string in string java

Input = "Alice is girl and Bob is boy";
Output = {Bob=1, Alice=1, and=1, is=2, girl=1, boy=1}

```
public static void main(String[] args) {
String str = "Alice is girl and Bob is boy";
Map<String, Integer> hashMap = new HashMap<>();
String[] words = str.split(" ");
for (String word : words) {
if (hashMap.containsKey(word))
hashMap.put(word, hashMap.get(word) + 1);
else
hashMap.put(word, 1);
}
System.out.println(hashMap);
```

---

9. Reverse the entire sentence

Input = "India is country My"

Output = "My country is India"

```
public static void main(String[] args) {
String str[] = "India is country My".split(" ");
String ans = "";
for (int i = str.length - 1; i >= 0; i--) {
ans = ans + str[i] + " ";
}
System.out.println(ans.substring(0, ans.length() - 1));
}
```

12. Reverse each word in a sentence

Input = "reverse each word in a string";

Output = "esrever hcae drow ni a gnirts"

```
public static void main(String[] args) {
String str = "reverse each word in a string";
String words[] = str.split("\\s");
String reverseWord = "";
for (String w : words) {
StringBuilder sb = new StringBuilder(w);
sb.reverse();
reverseWord = reverseWord + sb.toString() + " ";
}
System.out.println(reverseWord.trim());
}
```

---

13. String Anagrams: Determine if two strings are anagrams of each other

Input =
String str1 = "Army";
String str2 = "Mary";
Output = **army** and **mary** are anagram.

```
public static void main(String[] args) {
String str1 = "Army";
String str2 = "Mary";
str1 = str1.toLowerCase();
str2 = str2.toLowerCase();
// check if length is same
if (str1.length() == str2.length()) {
// convert strings to char array
char[] charArray1 = str1.toCharArray();
char[] charArray2 = str2.toCharArray();
// sort the char array
Arrays.sort(charArray1);
Arrays.sort(charArray2);
// if sorted char arrays are same, then the string is anagram
boolean result = Arrays.equals(charArray1, charArray2);
if (result) {
System.out.println(str1 + " and " + str2 + " are anagram.");
} else {
System.out.println(str1 + " and " + str2 + " are not anagram.");
}
} else {
System.out.println(str1 + " and " + str2 + " are not anagram.");
}
```
