240 CORE JAVA INTERVIEW QUESTIONS AND ANSWERS

Table of Contents

1) what are static blocks and static initalizers in Java ? 9

2) How to call one constructor from the other constructor ? 9

3) What is method overriding in java ? 9

4) What is super keyword in java ? 9

5) Difference between method overloading and method overriding in java ? 9

6) Difference between abstract class and interface ? 10

7) Why java is platform independent? 10

8) What is method overloading in java ? 10

9) What is difference between c++ and Java ? 10

10) What is JIT compiler ? 10

11) What is bytecode in java ? 10

12) Difference between this() and super() in java ? 11

13) What is a class ? 11

14) What is an object ? 11

15) What is method in java ? 11

16) What is encapsulation ? 11

17) Why main() method is public, static and void in java ? 12

18) Explain about main() method in java ? 12

19)What is constructor in java ? 12

20) What is difference between length and length() method in java ? 12

21) What is ASCII Code? 12

22) What is Unicode ? 13

23) Difference between Character Constant and String Constant in java ? 13

24) What are constants and how to create constants in java? 13

25) Difference between '>>' and '>>>' operators in java? 13

Core java Interview questions on Coding Standards 13

26) Explain Java Coding Standards for classes or Java coding conventions for classes? 13

27) Explain Java Coding standards for interfaces? 13

179) Define interface in java? 34
180) What is the purpose of interface? 34
181) Explain features of interfaces in java? 34
182) Explain enumeration in java? 34
183) Explain restrictions on using enum? 34
184) Explain about field hiding in java? 34
185) Explain about Varargs in java? 34
186) Explain where variables are created in memory? 35
187) Can we use Switch statement with Strings? 35
188) In java how do we copy objects? 35
Oops concepts interview questions 35
189) Explain about procedural programming language or structured programming language and its features? 35
190) Explain about object oriented programming and its features? 35
191) List out benefits of object oriented programming language? 35
192) Differences between traditional programming language and object oriented programming language? 35
193) Explain oops concepts in detail? 35
194) Explain what is encapsulation? 36
195) What is inheritance ? 36
196) Explain importance of inheritance in java? 36
197) What is polymorphism in java? 36
Collection Framework interview questions 36
198) What is collections framework ? 36
199) What is collection ? 37
200) Difference between collection, Collection and Collections in java? 37
201) Explain about Collection interface in java ? 37
202) List the interfaces which extends collection interface ? 37
203) Explain List interface ? 37
204) Explain methods specific to List interface ? 38
205) List implementations of List Interface ? 38
206) Explain about ArrayList ? 38

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.

```
}

}
```

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

178) Can we define static methods inside interface?
We can't declare static methods inside interface. Only instance methods are permitted in interfaces.only public and abstract modifiers are permitted for interface methods. If we try to declare static methods inside interface we get compilation error saying
"Illegal modifier for the interface method Classname.methodName(); only public & abstract are permitted".

179) Define interface in java?
Interface is collection of abstract methods and constants. An interface is also defined as pure or 100 percent abstract class.Interfaces are implicitly abstract whether we define abstract access modifier or not. A class implementing interface overrides all the abstract methods defined in interface. Implements keyword is used to implement interface.

180) What is the purpose of interface?
Interface is a contract . Interface acts like a communication between two objects. When we are defining interface we are defining a contract what our class should do but not how it does. An interface does'nt define what a method does. The power of interface lies when different classes that are unrelated can implement interface. Interfaces are designed to support dynamic method resolution at run time.

181) Explain features of interfaces in java?
1) All the methods defined in interfaces are implicitly abstract even though abstract modifier is not declared.
2) All the methods in interface are public whether they are declared as public or not.
3) variables declared inside interface are by default public, static and final.
4) Interfaces cannot be instantiated.
5) we cannot declare static methods inside interface.
6) ` implements' keyword is used to implement interface.
7) Unlike class, interface can extend any number of interfaces.
8) We can define a class inside interface and the class acts like inner class to interface.
9) An interface can extend a class and implement an interface.
10) Multiple inheritance in java is achieved through interfaces.

182) Explain enumeration in java?
Enumeration is a new feature from Java 5.0. Enumeration is set of named constants . We use enum keyword to declare enumeration. The values defined in enumeration are enum constants.Each enum constant declared inside a enum class is by default public , static and final.
Example :
package javaexamples;
public enum Days {
SUN,MON,TUE,WED,THU,FRI,SAT;
}
SUN,MON,TUE,WED,THU,FRI,SAT are enum constants.

183) Explain restrictions on using enum?
1) Enums cannot extend any other class or enum.
2) We cannot instantiate an enum.
3) We can declare fields and methods in enum class. But these fields and methods should follow the enum constants otherwise we get compilation error.

184) Explain about field hiding in java?
If superclass and subclass have same fields subclass cannot override superclass fields. In this case subclass fields hides the super class fields. If we want to use super class variables in subclass we use super keyword to access super class variables.

185) Explain about Varargs in java?
Beginning with Java 5 has a new feature Varargs which allows methods to have variable number of arguments. It simplifies creation of methods when there are more number of arguments. Earlier to java 5 Varargs are handled by creating method with array of arguments.
Ex : public static void main(String[] args)
A variable length argument is specified using ellispses with type in signature. main method with var args is written as follows:
public static void main(String … args)
If no arguments are passes we get array with size 0.There is no need for null check if no arguments are passed.

The advantage of using copyOnWriteArrayList is no need to synchronize list explicitly. So when we use copyOnWriteArrayList when a thread modifies the list while the other thread was iterating it does not modify original list but creates a copy of list with modified contents so that the iterator won't know the modifications made to original list.

229) Explain about fail fast iterators in java?

When iterator iterates over collection, collection should not be modified except by that iterator. Modification means collection cannot be modified by thread when other thread is iterating, if such modification happens a concurrent modification exception will be thrown.Such kind of iterators are fail fast iterators.
Ex : ArrayList,HashSet,HashMap. Almost all the iterators implemented in collections framework are fail fast.

230) Explain about fail safe iterators in java?

Fail safe iterators are iterators which does not throw concurrent modification exception, when one thread modifies collection and other thread in the process of iterating the collection.
It does not throw concurrent modification exception because when other thread was iterating it does not modify original list but creates a copy of list with modified contents so that the iterator won't know the modifications made to original list.
Ex : copyOnWriteArrayList

Core java Serialization interview questions

231) What is serialization in java?

Serialization is the process of converting an object in to bytes, so that it can be transmitted over the network,or stored in a flat file and can be recreated later. Serialized object is an object represented as sequence of bytes that includes objects data, object type, and the types of data stored in the object.

232) What is the main purpose of serialization in java?

The main uses of serialization are :
1) Persistence:
We can write data to a file or database and can be used later by deserializing it.
2) Communication :
To pass an object over network by making remote procedure call.
3) Copying :
We can create duplicates of original object by using byte array.
4) To distribute objects across different JVMs.

233) What are alternatives to java serialization?

XML based data transfer
JSON based data transfer.
XML based data transfer : We can use JIBX or JAXB where we can marshall our object's data to xml and transfer data and then unmarshall and convert to object.
JSON based transfer : We can use json to transfer data.

234) Explain about serializable interface in java?

To implement serialization in java there is an interface defined in java.io package called serializable interface. Java.io.Serializable interface is an marker interface which doesnot contain any any methods. A class implements Serializable lets the JVM know that the instances of the class can be serialized.
Syntax:

```
public interface Serializable {
}
```

235) How to make object serializable in java?

1) Our class must implement serializable interface.If our object contains other objects those class must also implement serializable interface.
2) We use ObjectOutputStream which extends OutputStream used to write objects to a stream.
3) We use ObjectInputStream which extends InputStream used to read objects from stream

236) What is serial version UID and its importance in java?

Serial version unique identifier is a 64 bit long value .This 64 bit long value is a hash code of the class name,super interfaces and member. Suid is a unique id no two classes will have same suid. Whenever an object is serialized suid value will also serialize with it.
When an object is read using ObjectInputStream, the suid is also read. If the loaded class suid does not match with suid read from object stream, readObject throws an InvalidClassException.