CORE JAVA

INTERVIEW QUESTIONS

NARESHi™ technologies
SOFTWARE TRAINING & DEVELOPMENT
Opp. Satyam Theatre,
Ameerpet, Hyderabad - 500 016.
E-mail:info@nareshit.com www.nareshit.com
Ph:23746666, 23734842 Cell: 9000994007, 9000994008
An ISO 9001 : 2008 Certified Company

Java History Interview Questions

1. Why JAVA language is developed?
2. When should we use C language & when should we use Java language?
3. What are the important features of Java?
4. What is OAK?
5. What is the abbreviation of Java?
6. What is a platform, Platform Dependency and Platform Independency?
7. What is the meaning of
    a. source code, compiled code, executable code
    b. compilation, execution
    c. compiler & interpreter
8. Why C, C++ programming languages are platform dependency?
9. How Java achieved Platform independency?
10. Why JVM platform dependent?
11. What is the Java's Slogan?
12. Explain why Java is called language, platform, & technology?

Java Software Interview Questions

1. How many types of Java Softwares do we have?
2. What is the difference between JDK, JRE, JVM & JIT?
3. What is Java Plug-in?
4. What is Java SE & what is the difference between Java SE & JDK?
5. What is the meaning of Major & Minor versions of Java?
6. What is the difference between Latest version & Stable version?
7. Will JVM installed automatically with OS?
8. Do we have single Java software for all types of OS?
9. What are the important folders & files of Java software?
10. Why Java is called open source software?

Environment Variables Interview Questions

1. What is an environment variable?
2. What is the difference between binary files & library files?
3. What is the difference between path, classpath & JAVA_HOME?
4. Why "path" is used by OS & Why "classpath" is used by compiler & JVM?

5. Who will use JAVA_HOME?
6. Why we do not set path & classpath for Game software?
7. What are the different ways to set path & classpath variables?
8. Is it possible to set multiple softwares in path variables?
9. What is the role of ";" in path & classpath?
10. What is the difference between set & export commands?
11. What is the difference between %path% & $path?
12. What is the difference between ";" & ":" in path?
13. Why should we place ".;" in classpath variable?
14. How will you solve if you find the "Error: could not find or load main class"

Java programming elements Interview Questions

1. What are the 10 programming elements of Java language?
2. List out all types of programming elements with their sub types?
3. What is the creation syntax of 10 programming elements?
4. What is a package & sub package, what is the use of it?
5. What is an interface, what is the use of it?
6. What is an abstract class, what is the use of it?
7. What is a class, what is the use of it?
8. What is a final class, what is the use of it?
9. What is an enum, what is the use of it?
10. What is an annotation, what is the use of it?
11. What is a variable, what is the use of it?
12. What is a method, what is the use of it?
13. What is a constructor, what is the use of it?
14. What is a block, what is the use of it?
15. What is an inner class, what is the use of it?
16. Explain all above programming elements with real-world examples?
17. What is the difference between class & abstract class & interface?
18. What are the different mathematical values we can store in a variable?
19. What is the difference between initialization logic & business logic?
20. Can we write business logic in constructor?
21. What is the use of import statement?
22. What is the difference between import & static import?

23. What are the essential statements of Java program?
24. What is the procedure to develop, compile & execute java program?

Java source File Level interview Questions

1. Can we create empty Java file, can we compile and execute it?
2. Can we create empty class, can we compile and execute it?
3. Is main method mandatory for compilation or for execution?
4. Is it mandatory that java file name and class name must be same?
5. When Java file name & class name should be same?
6. Why Java file name & public class name should be same?
7. What is the initial point of java class execution?
8. In a single Java file how many classes can we define?
9. If a java file has multiple classes what is the java file name?
10. In a single java file, how many public classes can we define?
11. If a java file has multiple classes, how many .class files are generated by compiler?
12. Can we compile multiple classes at a time?
13. Can we execute multiple classes at a time?
14. What is a user defined method?
15. Will JVM executes user defined method automatically?
16. What is the rule in calling user defined method?
17. What are the two keywords we have in Java to provide memory to class members?
18. What is the difference between static & new keywords?
19. Why JVM does not execute user defined methods, why it executes only main method?
20. How many user defined methods can we create in a class?
21. What is the order of execution of user defined methods?
    Choose one option:
    a. In the order they are declared
    b. In the order they are called
22. If a class does not have main method how can we execute that class methods?
23. If a java file has multiple classes, can we define main method in all classes?

24. Can we call main method explicitly, what is the syntax to call main method explicitly?
25. What will happen if we call main method in its own body?
26. Can we overload main method?
27. If we overload main method which main method is executed?
28. If we create every class in a separate java file, how can we call one class method from another class; in this case should we compile each class separately?
29. What is the meaning of auto compilation?
30. What is the difference
    a. System.out.print()
    b. System.out.println()
    c. System.out.printf()
31. What is the Java source file structure?
32. Can we place package statement anywhere in the java file?
33. Can we place import statement anywhere in the java file?
34. How many package statements can we place in the java file?
35. How many import statements can we place in the java file?

Main method interview questions

1. What is the purpose of main method?
2. What is the valid main method syntax?
3. If we create a class with name String, then will main method executed?
4. Why main method should be public?
5. Why main method should be static?
6. What is the right order to place public & static?
7. Why main method has void, can we change void to some other type?
8. Why main method name is main?
9. What is the use of main method parameter?
10. Why main method parameter type should be String[], can we change it to other type?
11. Which part of the main method can be changeable?
12. Can we add final keyword to main method?

13. Identify which syntax main method is executed by JVM?
    a. public static void main(String[] args)
    b. public static void main(String... args)
    c. public static void main(int[] args)
    d. public static void mian(String[] args)
    e. public static final void main(String[] args)
    f. public static void main(String[5] args)
    g. public static void main(String[] HariNareshIT)
14. Can we also call main method explicitly?
15. What is the syntax to call main method?
16. If we call main method in its own body, what will happen?
17. Can we overload main method?
18. If we overload main method which main method is executed?
19. If a java file has multiple classes, can we define main method in all classes?
20. Why main method is called mediator method between programmer & JVM?
21. Why main method is called initial point of class logic execution?
22. Can we execute a class without main method?

Coding standards and Naming conventions interview Questions

1. What is coding standards and Naming conventions?
2. If we do not follow above CS and NS will it leads to CE or RE?
3. What is the naming convention we must follow in creating
    1. Class name
    2. Variable name
    3. Method name
    4. Package name
    5. Final variable
4. By following Naming convention can you identify the type of programming element just by seeing its name?

Java Tokens interview Questions

1. What is a token?
2. List out all 7 types of Java tokens?

Java Comments interview Questions

1. What is a comment?
2. What is the need of comment?
3. How many types of comments are there?
4. Can we comment program statements like variables or methods using a comment?
5. How can we provide documentation for programming elements?
6. Will compiler read statements those are placed in comment?
7. If we comment a variable, will it be available in .class file?
8. Can we call a variable or method which is placed in comment?
9. Can we comment a comment?
10. Can we comment multiline comment using multiline comment?
11. If we create a java file only with comments, will .class file generated?
12. Identify invalid comments from below list?
    a. ////////////////
    b. /*
    c. /* /* */
    d. /*/* */ */
    e. /* // */
    f. /* /* * / */

Identifiers Interview Questions

1. What is an identifier?
2. Can we create a class or method without name?
3. When will we get CE: <identifier> expected?
4. What are the Rules in defining Identifier?
5. Can we start identifier with **number**?
6. Can we start identifier with **$** or **_**?

7. Can we use number in identifier, if yes in which position?
8. Can we use spaces in identifier?
9. Can we create two variables with the same name ***balance***, ***Balance***?
10. Can we create class name as ***Bank Account***?
11. Can we use keyword as an identifier?
12. Can we use predefined class name as identifier?
13. If we create a class with name String, how can we differentiate predefined class String?
14. In which programming elements name **$** and **_** are generally used?
15. Why underscore is called connector symbol?
16. Is there any limitation in identifier length?
17. Identify invalid identifier from the below list?
    a. hihellohru
    b. program1
    c. 1stStudent
    d. No#1Student
    e. MIN_BALANCE
    6. String
    7. Public
    8. $$_$$_$$
    9. main
    10. Bank Account

Keywords Interview Questions

1. What is a keyword?
2. For what purpose we use keyword?
3. Using keyword what are the different operations we will do in a Java file?
4. How many keywords Java supports **50** or **53**?
5. Can you tell all 50 keywords?
6. Why **goto, const** keywords are given without implementation?
7. Is null a keyword?
8. Is true a keyword?
9. Is false a keyword?
10. Is annotation a keyword?
11. Can we use null as user defined identifier, means can we create a class with name **null**?

12. Identify valid key words in the below list?
    a. static
    b. default
    c. Public
    d. imported
    e. synchronize
    f. Class
    7. main
    8. marker
    9. sizeof
    10. instanceof
    11. null
    12. finalize

Separator interview questions

1. What is a separator?
2. How many separators java supports?
3. What is the difference between separator & operator?
4. Is + a separator or operator?
5. Is {} a separator or operator?
6. In which Java version the separator <> is added?
7. What are the 3 places we can use *:* as separator?
8. Why *;* is called terminator?
9. Where *;* is used as separator & terminator?

Datatype interview Questions

1. What is a data type & what is the use of data type?
2. How many types of data types Java supports?
3. What is the difference between primitive type & referenced type?
4. Why do we have 8 primitive data types & 4 referenced data types?
5. What is size, range & default value of primitive types?
6. What is the difference between "int" data type in Java & C language?
7. Why char data type size is 2 bytes in Java, whereas its size is 1 byte in C language?
8. Will referenced data type have size & range?
9. Why referenced data types are called derived data types?
10. How much memory is occupied by a String?
11. What is a variable, how can we create a variable?
12. What is the variable creation syntax?
13. What is the difference between primitive variable & referenced variable?

14. What is the limitation of variable?
15. What is the syntax for creating an array?
16. What is the difference in creating an array in C language & in Java?
17. What is the limitation of Array?
18. Do we have any inbuilt data type to store different type of values?
19. What is a user defined data type, what is the use of it?
20. How can we create user defined data type & why class is called user defined data type?
21. Create a class to store differe nt type of values in JVM?
22. What is the keyword we must use to create an object from a class?
23. What is a class & object as per data type?
24. Draw class object memory diagram that is created internally in JVM?
25. What are the two memories created using a class as data type?
26. What is the default size of a referenced variable & an empty class object?
27. What are the different ways to store data in Java application?
28. When should we use a variable, array object & class object?
29. Will JVM store given data in decimal format?

Literal Interview Questions

1. What is a literal?
2. How many types of literals Java supports?
3. What is the default type of integer literal?
4. How can we store integer literal as long type value?
5. What is the default type of floating-point literal?
6. How can we store floating-point literal as float type value?
7. Do we have byte & short type literals?
8. How can we represent int literals as byte & short type literals?
9. What is a cast operator, what is righ t place to use it in an expression?
10. Identify the data types of the literals: 45, 45L, 45F, 45D, 45.0, (byte)45, (short)45?
11. What is the default type "binary & octal & hexadecimal" literal?
12. How can you identify *integer literal* is decimal or binary or octal or hexadecimal type?
13. What is the new literal added in Java 7 version?
14. What are the rules in creating binary, octal, hexadecimal literals?

15. How can we convert binary, octal, hexadecimal literal to decimal value?
16. How can we identify a char literal?
17. Can we store more than one character in single quote?
18. Can we use empty single quote?
19. How can we represent int type literal as char type?
20. How can we convert char to any number type literal?
21. Can we represent int type literal as Boolean?
22. How can we identify a String literal?
23. Can we store more than one character in double quote?
24. Can we use empty double quote?
25. If we place an alphabet say p directly in a class, will it be consider as char literal?
26. If we place a word say hi directly in a class, will it be consider as String literal?
27. Can you explain when will you get below compile time errors?
    1. CE: not a statement
    2. CE: unexpected type
    3. CE: cannot find symbol
    4. CE: unclosed character literal
    5. CE: empty character literal
    6. CE: integer number too large

Type conversion & casting Interview Questions

1. What is the meaning of type conversion & casting?
2. What are the rules we must follow in type conversion?
3. What is the meaning of compatible types?
4. Which data type is incompatible type among all primitive types?
5. Draw primitive type auto conversion chart?
6. Is "type casting" meant for converting incompatible type or height range data type?
7. What will happen if result value *is greater than* casting data type range?
8. What is the short-cut formula for type casting in reducing highest range value?

9. Can you explain when will you get below compile time errors?
    1. CE: incompatible types
    2. CE: possible loss of precision
    3. CE: inconvertible types
10. Can we assign (long to float) & (float to long)?
11. Can we assign (char literal to int variable) & (int literal to char variable)?
12. When "?" is stored in char variable?
13. Can we assign 10.5 to float variable?
14. Can we assign 1234567891234567 number to long type variable?
15. What types of literals are only allowed to assign to lesser range data type variable?
16. Identify will below statements compiled?
    a. byte b1 = 50; 3) char ch1 = 97; 5) boolean bo = 1;
    b. byte b2 = 'a'; 4) float f1 = 10.5; 6) byte b3 = 128;
17. What is the diff between primitive type conversion & referenced type conversion?
18. When will two classes are said to be compatible?
19. What is the meaning of upcasting & downcasting?
20. When will we get ClassCastException?
21. What is the use of *instanceof* operator?
22. What is the difference between cast operator & *instanceof* operator?

Type Promotion Interview Questions

1. What is the meaning of type promotion?
2. What is an operator, operand, & expression?
3. What is the rule in preparing an expression?
4. What is the result type will be return from an expression?
5. If an expression contains only literals how compiler calculates?
6. If an expression contains at least one variable how compiler calculates?
7. Why byte, short, byte data types are promoted to int type?

8. What is the algorithm compiler & JVM follows in evaluating an expression?
9. Will Java follows BODMAS rule in evaluating an expression?
10. What are the operators executed from RIGHT to LEFT?

Reading runtime values interview questions

1. What is are the different ways we can read input values from keyboard?
2. Explain advantage & disadvantage of each approach?
3. What is the default type applied to values sending from keyboard?
4. How can you convert a number from string type to primitive type?
5. How can you create CUI & GUI for reading runtime values?
6. What is the advantage of Scanner class over BufferedReader class?
7. What is special feature given in Console class?
8. How can you read password from CUI & GUI?

Command line arguments interview questions

1. What is the meaning of command line arguments?
2. What is the object JVM uses to pass command line arguments to a Java program?
3. Why main method has String[] parameter?
4. How can we read command line arguments in a java program?
5. In the below command, how many arguments are passing to Java application

   >java Hi Hello Hru?
6. Assume we are reading 3 values in java program using args[0], args[1], args[3] then what will happen in the below two cases
   a. only two values are passing from command line
   b. Three values are passed

Operators interview Questions

1. What is an operator & operand?
2. What is the meaning of operator precedence & operator associativity?

3. What are the operators executed from RIGHT to LEFT in java?
4. How many types of operators Java supports?
5. Totally how many operators Java has?
6. What is the overloaded operator in Java?
7. When + operator act an Addition operator & Concatenation operator?
8. What is the value of ½ in java?
9. When will we get ArithmeticException, Infinity & NaN from div operator?
10. What is the difference between ++a & a++?
11. How can we compare two primitive values & two objects?
12. What is the difference between
    a. ==, != operator
    b. equals() method
    c. equalsIgnoreCase() method
13. Can you identify when will you get below CEs?
    a. CE: bad operand types for the binary operator
    b. CE: incomparable types
14. What is the difference between &&, ||?
15. What is the difference between &, |?
16. What is the difference between &&, &?
17. What is the difference between ||, |?
18. What the functionality ternary operator?

Control statements Interview Questions

1. What is the meaning of control statement?
2. What is the difference between conditional, loop & branching statements?
3. What is condition value type allowed in if?
4. Is "if(1){}" a valid statement in Java?
5. What is the difference between if & else?
6. What is the difference between if...else & switch?
7. What is the difference between case & default in switch block?
8. What is the difference between while & do..while?
9. When should we use while, do...while, for loops?
10. What is enhanced for loop, what is the use of it?

11. "enhanced for" loop & "for-each" loop are both same?
12. What is the difference between break, continue, return?
13. What s the difference between "retun;" & "retun value;"?
14. Can we place break & continue & return statements anywhere in the method?
15. When will you get CE: unreachable statement?
16. How can we create infinite loop using while & for?
17. Can we place while(false){} in a java program?
18. What is the difference between infinite loop & recursive method call?
19. What is an inner loop?
20. What is the use of label concept?
21. **What is a label, is it a keyword?**
22. What is the syntax to create a label?
23. What are the rules we must follow in creating a label?
24. How can we break outer loop from inner loop?

Modifiers interview Questions

1. What is a modifier?
2. List out all 13 modifiers?
3. Is the "interface" a modifier keyword?
4. What is right place to use modifier in a programming element's declaration?
5. What is the difference between accessibility modifiers & other modifiers?
6. Can we apply more than one accessibility modifier?
7. Can we apply more than one execution level modifier?
8. Explain the functionality of all 13 modifiers?
9. What are the modifiers allowed for variable, method, constructor, block, & class?
10. Why only *final* modifier allowed to parameter & local variable?
11. Can you tell when will we get below compile time errors?
    a. CE: modifier not allowed here
    b. CE: illegal combination of modifiers
    c. CE: repeated modifier

Accessibility Modifiers interview Questions

1. What is an Accessibility modifier?
2. What is the difference between Accessibility modifiers & accessibility specifies?
3. How many accessibility modifiers java supports?
4. What are the different levels of accessibility permission levels?
5. What the accessibility modifier keywords?
6. Why accessibility modifiers are not allowed for local variables inside a method?
7. Why private & protected are not allowed to outer class?
8. Can we declare inner class as private or protected or static?
9. What is the default accessibility modifier of class and its members?
10. What is the default accessibility modifier of interface and its members?
11. How can create a class member to be accessed only in subclass from other package?

Method interview questions

1. What is a method,, What is the use of method?
2. What is the meaning of business logic?
3. Are both method & function same?
4. What is the method creation syntax?
5. Explain all 7 parts of the method creation syntax?
6. What is the difference between method prototype & signature?
7. What is the difference between method parameter & argument?
8. What are the different modifiers allowed to a method?
9. What is the difference between public & static in accessing a method?
10. Can we create a method inside another method?
11. What are the modifiers allowed to a method?
12. What are the different types of methods do we have in java?
    1. What are the differences between abstract method & concrete method?
    2. What is a private and non-private method?
    3. What is a Static and non-static method?

4. What is a Void and non-void method?
5. What is a Parameterized and Non-Parameterized method?
6. What is a Final and non-final method?
7. What is a Native and non-native method?
8. What is a Synchronized and non-synchronized method?
9. What is a Strictfp and non-strictfp method?

13. Is a native method an abstract method?
14. What are the points we must consider in calling a method?
15. What is the difference between calling a method & executing a method?
16. Can you identify when will you get below compile time errors?
    1. CE: illegal start of type
    2. CE: non-static method cannot be referenced from static method
    3. CE: missing return type
    4. CE: missing return value
    5. CE: cannot return a value from a method whose result type is void
    6. CE: missing return type
    7. CE: void type not allowed
    8. CE: method cannot apply to given arguments

Variable Interview Questions

1. What is a variable?
2. When should we create a variable?
3. What is the difference between primitive & referenced variables?
4. What are the different places we can create a variable in a java file?
5. What is the difference between variable declaration, initialization & assignment?
6. What is the difference between variable initialization & variable assignment?
7. What is the difference in modifying primitive & referenced variables?
8. What is the life time & scope of all 4 types of variables?

9. How many types of variables java supports?
    a. What is the difference between private and non-private variables?
    b. What is the difference between static and non-static variables?
    c. What is the difference between final and non-final variables?
    d. What is the difference between volatile and non-static variables?
    e. What is the difference between transient and non- transient variables?
    f. What is the difference between Parameter & local variables?
10. What are the modifiers allowed to variables?
11. What is the only modifier allowed to parameter & local variables?
12. What variables will have default value?
13. What are the rules on local variables?
    a. If we create a local variable without assigning a value, what is the value stored?
    b. Can we access local variable from other methods?
    c. Can we access local variable before its declaration statement?
    d. How can we use the value or object stored in one method in another method?
14. How can we access static, non-static variables within the class & in other classes?
15. Why static variable is called class variable?
16. Why non-static variable is called instance variable?
17. Why local variable is called auto variable?

JVM Architecture interview questions

1. What is JVM?
2. How many types of JVMs do we have?
3. Which JVM is used for running java program by default?
4. How can we use server JVM?

5. What will happen when we run java command?
6. What are the five phases we have in developing, compiling & executing a class?
7. What are different components existed in JVM for executing a class?
8. How many types of members we can create in a class?
9. Why Java has static & non-static members?
10. When static members are executed?
11. When non-static members are executed?
12. What are the members are automatically executed by JVM?
13. Explain five runtime areas of JVM?
14. What is the use of Class & ClassLoader?
15. Where & How class byte code is stored in JVM?
16. Where objects are created in JVM?
17. Where static and non-static variables are provided memory?
18. Where local variables are provided memory?
19. Where methods logic is executed in JVM?
20. Where non-static method is stored & where is it executed?
21. How classes are found by compiler and JVM?
22. Explain ClassLoader sub system architecture?
23. Explain about application classpath, extension classpath, bootstrap class path?
24. Explain Thread architecture?
25. What is a stack frame?
26. When a stack frame is created and destroyed?
27. Explain Stack Frame architecture?
28. In an expression, why byte, short, char data types are promoted to int type?
29. When a stack frame, thread, & JVM is destroyed?
30. How can decompile & see java bytecode format?
31. Why Java compiled code is named bytecode?
32. Draw JVM architecture with all 4 types of variables & objects memory diagram?
33. Draw JVM architecture for a method execution?
34. Whose stack frame is crated first is it for Static Block or for main method?
35. Where static block & constructor logic is stored & executed?

Static & non-static members execution flow interview questions

1. Including main method, what are the 12 members we can create in a class?
2. Like main method, will JVM executes other members declared in a class?
3. What is the basic rule of JVM to execute a class member?
4. Why these 12 members are called static & non-static members?
5. Why do we have static & non-static members in Java, what is the use of them?
6. When should we use static members & when should we use non-static members?
7. Who will provide memory to static & non-static members?
8. What is the meaning of class context?
9. What is the meaning of instance context?
10. Why static members are also called as class members?
11. Why non-static members are also called as instance members?
12. Explain the execution flow of all 4 static members?
13. Explain the execution flow of all 4 non-static members?
14. Which one is executed first--- static variable or static block?
15. Which one is executed first--- static block or main method?
16. Which one is executed first--- non-static variable or non-static block?
17. Which one is executed first--- non-static block or constructor?
18. Do you know when will you get CE: illegal forward reference?
19. How many times a class is loaded into JVM?
20. How many objects can be created from a class?
21. How many times static members are executed?
22. How many times non-static members are executed?
23. When should we use static variable & instance variable?
24. When should we use static method & instance method?
25. When should we use static block & instance block?
26. When should we use constructor & main method?
27. Can we execute a class without main method?
28. Can we execute a class with only non-static members?

29. In how many ways we can load a class into JVM, explain all 5 ways of loading a class?
30. In how many ways we can create object of a class, explain all 5 ways of object creation?
31. In all five ways of loading a class in which way only main method is executed?
32. What are the members executed when class is loaded into JVM?
33. What are the members executed when an object is created?
34. How a class can create its object by its own when it is loaded into JVM?

Static Keyword Interview Questions

1. What is the use of static keyword?
2. What is the meaning of *class context*?
3. When a member is called as static member?
4. To what are the 5 different members' static keyword is allowed?
5. Why static members are called as class members?
6. Can we apply static keyword to outer class & inner class?
7. Can we apply static keyword to a local variable/parameter?
8. What is the difference in apply static keyword to a variable & a method?
9. Why static keyword is not allowed to constructor?

New Keyword Interview Questions

1. What is the use of new keyword?
2. What is the meaning of instance context?
3. When a member is called as non-static member?
4. What are 5 different non-static members?
5. Why non-static members are called as instance members?
6. Will new keyword provide memory to static variables?
7. What is an object & instance; is instance and object both are same?
8. What is the object creation syntax?
9. In the statement "String str;" is "str" an object?

10. Why object is called instance of a class?
11. Explain all 4 activities involved in object creation process?
12. What is the Job of "new" keyword & what is the Job of constructor?
13. Who creates object of a class, is it new keyword or constructor?
14. Who will return object reference, is it new keyword or constructor?
15. Why new keyword will use constructor?

Constructor Interview Questions

1. What is a constructor?
2. What is the use of constructor?
3. What are the rules in creating a constructor?
4. What is the rule in invoking a constructor?
5. Why constructor should be called with new keyword?
6. What are the other ways of calling constructor?
7. If we call constructor using this() or using super() will new object created?
8. Why constructor name should be same as its class name?
9. Why return type is not allowed for constructor?
10. If we place return type in constructor declaration, is it leads to CE?
11. Can we define a method with class name, if yes, is it recommended, if no why?
12. How compiler and JVM can differentiate method and constructor blocks, if a method is also created with class name?
13. Why "return <value>;" is not allowed constructor?
14. Why "return;" is allowed in constructor?
15. Why 8 execution level modifiers are not allowed to constructor?
16. How many types of constructors will java supports?
17. What is a copy constructor; will Java supports its creation?
18. What is the alternative given in Java for copy constructor functionality?
19. What is a static constructor; will Java supports its creation?
20. Why is a default constructor?

21. What are the differences between default and no-arg constructor?
22. When will compiler provide constructor in a class?
23. Can we have both default & no-arg constructor in class?
24. Can we have both default & parameterized constructor in class?
25. Can we have both no-arg & parameterized constructor in class?
26. What is the constructor overloading?
27. If we invoke one constructor, will all other constructors are executed?
28. How can we execute overloaded constructors of a class without creating new object?
29. What is the difference between this() & this(5)?
30. How non-static variables will be differentiated from one object to another object in a constructor in its initialization logic?
31. How can we initialize non-static variables in different instances with different values at the time of object creation & after object creation?
32. Explain about setXXX() & getXXX() methods?
33. Explain about mutator & accessor methods?
34. Are setXXX, mutator method & getXXX & accessor methods same?

This Keyword Interview Questions

1. What is the definition and need of *this* keyword?
2. Is this keyword an *operator* or a *variable*?
3. What is the meaning of current object?
4. What will be the problem if *this* keyword is not existed in Java?
5. List out all 6 usages of this keyword?
6. What is the difference between "this." & "this()"?
7. Like non-static variables, will constructor & non-static method get separate copy of memory in every object?
8. How non-static variable is differentiated from multiple objects in a constructor and non-static method?
9. How can we develop code common to all objects of a class for executing it separately for each object for either setting/getting/printing values?

10. Why we allow to access non-static variable directly from a constructor or non-static method, why we are not allowed to access from static method?
11. What are the rules in using this keyword?
12. Prove that "this" variable stores current object reference?
13. Prove that "this" is a non-static variable?
14. Prove that "this" is a final variable?
15. What is the data type of "this" variable?
16. Prove that "this" variable data type is current enclosing class type?
17. Why we cannot access this keyword in a static method?
18. Can we access static variable or method using this keyword?
19. How can you pass current class object to another class method or constructor as an argument?

Blocks interview questions

1. What is a block?
2. How many types of block do we have in Java?
3. What is the difference between static block, instance block & local block?

Local block interview questions

1. What is local block?
2. What is the use of local block?
3. How can we restrict local variable scope in a method?
4. Can we declare a variable in a local block that is already declared in its method?
5. Can we declare a variable in a method that is already declared in a local block?
6. Can we create multiple local blocks?

Static block interview questions

1. What is a SB?
2. What is the use of static block?
3. Why static block does not have name?

4. What are the statements not allowed inside a SB?
5. Why "*return;*" is not allowed in a static block?
6. Will JVM execute SB automatically?
7. Can developer execute SB?
8. When static block logic is executed & how many times?
9. Where Static Blocks logic is stored and where its logic is executed in JVM?
10. How many static blocks we can define in a class?
11. What is the order of execution of all SBs?
12. If we create multiple static blocks why only one static block will be there in .class file?
13. Can we create a static block in side another static block?
14. If we nest static block what will happen, is it leads to CE?
15. What is the order of execution of SB and MM?
16. What is the output we get if we run a class only with SB?
17. How can we execute class logic without using SB & MM?
18. How can we execute main method at the time of class loading?
19. What is the order of execution of SV and SB?
20. When will we get CE: illegal forward reference?
21. Explain 4 steps involved in executing SV, SB, MM & SM when class is loaded into JVM?

Non-Static block interview questions

1. What is a NSB?
2. What is the use of NSB?
3. Why NSB does not have name?
4. What are the statements not allowed inside a NSB?
5. Why "*return;*" is not allowed in a NSB?
6. Will JVM execute NSB automatically?
7. Can developer execute NSB?
8. When NSB is executed & how many times?
9. Where NSB logic is stored and where its logic is executed in JVM?

10. How many NSBs we can define in a class?
11. What is the order of execution of all NSBs?
12. Can we create a NSB in side another NSB?
13. If we nest NSB, we don't get CE, why?
14. What is the order of execution of NSB and constructor?
15. What is the output we get if we run a class only with NSB?
16. Can we execute a class only with non-static members?
17. When we create a NSB in a class, why we will not find NSB in a .class file?
18. Where NSB code is placed inside .class file?
19. What is the order of execution of NSV and NSB?
20. When will we get CE: illegal forward reference?
21. Explain 4 steps involved in executing NSV, NSB, constructor & NSM when a class object is created?

Static variable Modifications Interview Questions

1. What is the use of static variable?
2. How many copies of memories are provided to a static variable?
3. How can we access a static variable with in the class & in other classes?
4. If we modify a static variable in one method is that modification remain in static variable after this method execution?
5. If we modify a static variable in one method will that modified value is accessible to all other methods of its class?
6. If we modify a static variable in some other class method, will that modified value is accessible to all other classes loaded in this JVM?

Non-static variable modifications Interview Questions

1. When a variable can be called as Non-Static variable?
2. What is the use of non-static variable?
3. Will JVM execute non-static variables automatically?
4. When and by whom memory location is provided and where and how?
5. How many copies of memories are provided to a non-static variable?
6. Draw JVM Architecture to show object structure?

7. How can we access a non-static variable with in the class & in other classes?
8. How many objects can be created from a class?
9. From where values are stored in second object, are they from class or from previous object?
10. Where compiler and JVM search the non-static variable in the statement "***e.x***"?
11. What is the difference between "**e1.x**" & "***e2.x***"?
12. If we modify non-static variable using one object is that modification affected to another object?
13. If we modify non-static variable in one method will that modified value is accessible to all other methods of its class?
14. If we modify a non-static variable in some other class method, will that modified value is accessible to all other classes loaded in this JVM?
15. How many referenced variables can point to a single object?
16. If we modifying object using one referenced, will that modification affected to another referenced variables those are pointing to the same object?

Null & NullPointerException interview questions

1. What is the default value of every referenced type variable?
2. null & NULL are both same?
3. What is type of null?
4. What is the type of "null"?
5. What are the possible values we can assign to a referenced variable?
6. How many types of referenced variables do we have based on the value assigned?
7. What is the difference between object referenced variable & null referenced variable?
8. What will happen if we call non-static variable or non-static method using null referenced variable?
9. In project when will we get NullPointerException, how should we solve this exception?

10. If we create a referenced variable at class level, what is the default value stored in it?

Empty referenced variable Interview questions

1. What is an empty referenced variable?
2. Where can we create empty referenced variable?
3. Can we create empty referenced variable at class level?
4. Can we use empty referenced to call variable or method?
5. Is it leads CE or RE, if CE, what is the CE, why?
6. What is the difference between
    1. empty referenced variable,
    2. null referenced variable &
    3. object referenced variable

Accessing static variable using referenced variable interview questions

1. Can we access a static variable using referenced variable?
2. Can we access a static variable using null referenced variable, will it leads to NullPointerException, if no why?
3. Can we access a static variable using empty referenced variable?
4. In how many ways we can access a static variable within the class members?
5. In how many ways we can access a non-static variable within the class members?
6. Where compiler and JVM search the static variable in the statement "***e.a***"?
7. What is the difference between "**e1.a**" & "***e2.a***"?
8. Where compiler and JVM search the non-static variable in the statement "***e1.x***"?
9. What is the difference between "**e1.x**" & "***e2.x***"?
10. Draw JVM architecture with static & non-static variables memory with multiple objects & null referenced variable?
11. If we modify static variable using one object will that modification effected to another object?

12. If we modify non-static variable using one object will that modification effected to another object?

Printing Object Interview questions

1. What is printed when object is printed?
2. How println() method can retrieve the given object's classname@hashcode?
3. In which class toString() method is defined?
4. How can we print an object's data, when an object is printed?
5. When we print String class object, why its content is displayed?
6. When we print our class object, why its classname@hashcode is displayed?
7. What is printed when we print object referenced variable?
8. What is printed when we print null referenced variable?
9. What is printed when we print empty referenced variable?

Static & non-static methods

1. What is a static method?
2. What is a non-static method?
3. Will JVM executes static or non-static methods automatically?
4. Inside JVM where static & non-static method logic is stored & where is it executed?
5. What is the difference in memory location for non-static variable & non-static method, explain with JVM architecture?
6. When should we create a method as static or non-static method?
7. Can we access static variables directly in static & non-static methods?
8. Can we access non-static variables directly in static & non-static methods?
9. How compiler & JVM will process the statement e.m1()?
10. What is the difference in e1.m1() & e2.m1()?
11. In a non-static method or constructor if we modify non-static variable using this keyword in which object those modified values are stored?
12. What will happen if we modify NSV using local object referenced variable, is that modification effected to current object?
13. Will *this* keyword exist in a static method?

14. How can we differentiate non-static variable from multiple objects In a static & non-static method?
15. How can we pass input values /objects to a method for its logic execution?

Pass-By-Value interview questions

1. What is the meaning of pass-by-value?
2. Will Java supports Pass-By-Reference?
3. In how many ways we can pass primitive value into a method?
4. In how many ways we can pass object into a method?
5. When we pass object as a current object, inside that method where is it stored?
6. When we pass object as an argument object, inside that method where is it stored?
7. Can we pass an object as current object to a static method?
8. Can we pass one class object to another class as current object?
9. When we pass primitive value as an argument to a method, if we modify it using method parameter, will that modification affected to passed-in variable?
10. When we pass object reference as an argument to a method, if we modify that object values using method parameter, will that modification affected to passed-in variable?
11. Assume we pass object reference as an argument to a method, we assigned new object to parameter, then if we modify that object values using method parameter, will that modification affected to passed-in variable?
12. When we pass object as a current object to a method, if we modify that object values using this keyword, will that modification affected to passed-in variable?

Final keyword interview questions

1. What is a final keyword?
2. What is the use of final keyword?
3. To what programming elements final is allowed & not allowed?
4. Why final keyword is not allowed to constructor, block, Abstract Class, interface, enum & annotation?
5. What is the rule on final variable, final method, final class?
6. What is the difference between final, finally & finalize?

Final variable interview questions

1. What is a final variable? Is it possible to modify final variable value?
2. What is the diff in declaring primitive variable as final & referenced variable as final?
3. How many types of final variables do we have in Java?
4. When should we declare static variable as final & non-static variable a final?
5. When should we declare local variable or parameter as final?
6. Where can we initialize static final variable & non-static final variable?
7. What is right procedure in initializing non-static final variable?
8. Can we initialize static final variable in all static blocks?
9. Why we must initialize non-static final variable in all constructors?
10. Where can we initialize final local variable & final parameter?
11. Why we must initialize static & non-static final variables before program execution is started, where as we can leave local final variable as uninitialized?
12. Why Compiler removes final variable declaration statement?
13. Why class will not be load into JVM, when we access static final variable?
14. Why class will be loaded into JVM, when we access static final variable that is initialized in static block?
15. Prove it Static final or non-static final variable will also contain default value?
16. If we declare a referenced variable as final will its object also final?
    1)If we declare array variable as final, can we modify values in array locations?
    2)If we declare class variable as final, can we modify its object values?
17. How can we create immutable variable & immutable object?

OOP interview questions

1. What is OOP; is it a language or technology or technique?
2. Why OOP concepts?
3. What is the meaning of representing an object?
4. What are the problems with structure programming language?
5. What is the difference between Object-based & Object-oriented programming?
6. What are the building blocks of object?
7. What is an object, what is a class, what is an instance?
8. Is instance and object both are same?
9. Why class is called blue print of an object?
10. Why object is called instance of a class?
11. What are the object characteristics?
12. How can we create object properties in a class?
13. How can we implement object behaviours in a class?
14. What is a hash code, how can we generate it?
15. What are the 5 steps we must follow in designing an object?
16. Draw object design block diagram?
17. What is the difference between business object & container object?
18. What is the meaning of *object class*, *user class*, & *executable class*?
19. Will we create main method in every class, if not, why?
20. Create a class to store an Employee object data in Java program?
21. Create a class Company for creating different employees of this company?
22. Create a project to store & display Student object data in JVM?

Object design models interview questions

1. How will you write a class for representing multiple objects of same type has same properties & same behavior implementation?
2. How will you write a class for representing multiple objects same type has different properties & different behavior implementation?

Object relations interview questions

1. How many relations we can develop between objects?
2. When should we create IS-A. Has-A, USES-A relation between objects?
3. How should we develop code for creating IS-A, HAS-A, USES-A relation?
4. What are the keywords we use for developing IS-A relation?
5. What is inheritance, composition, association, aggregation?

OOP Principles interview questions

1. What are the OOP principles?
2. Will OOP principles provide any predefined code?
3. Is abstraction an OOP principle?
4. What are the benefits we will get with OOP principles?
5. What is encapsulation?
6. What is inheritance?
7. What is polymorphism?
8. What is abstraction?
9. Explain all 4 principles with real-world examples?

Encapsulation interview questions

1. What is encapsulation?
2. How can we develop encapsulation in Java?
3. What are the benefits of encapsulation?
4. What is the meaning of data hiding?
5. What is the meaning of implementation hiding?
6. What is problem if we do not implement encapsulation in projects?
7. What is meaning of tight encapsulation?
8. What is the difference between encapsulation & abstraction?

Inheritance interview questions

1. What is inheritance?
2. What are the advantages of inheritance?
3. What is the meaning of reusability?

4. How can we develop inheritance in Java?
5. What are the differences between extends & implements keywords?
6. Why we cannot derive an interface from a class?
7. What is the meaning of "superclass members are inherited to subclass"?
8. What are the members not inherited from superclass?
9. How can we reuse super class static & non-static members in subclass?
10. How many subclasses we can create from a superclass?
11. If we change a variable value or method logic in superclass will it affected to all its subclasses?
12. What type of logic we must place in super class & subclass?
13. What is the meaning of generalized logic & specialized logic?
14. How many types of inheritance Java supports?
15. Why Java does not support multiple inheritance with classes?
16. Why Java do not have problem in developing multiple inheritance with interface?

Static & non-static members' execution flow in inheritance

1. What is the order of compiling, loading and instantiating classes in inheritance?
2. When we load subclass will all its super classes also loaded into JVM?
3. When we load superclass will all its sub classes also loaded into JVM?
4. How can JVM load super class when sub class is loaded?
5. Why JVM will not load sub classes when super class is loaded?
6. Can we call super class variable & methods in sub class directly by their name?
7. Can we call sub class variable & methods in super class directly by their name?
8. Explain static members' execution flow from super class to subclass?
   - Will static variable & static blocks are executed from both super & sub classes?
   - Will main method executed from both superclass & subclass?
   - If we access subclass static variable in superclass static block what is the value return? Is it assigned value or default value?

- Is it possible to access subclass variable from superclass directly by its name?

9. Explain non-static members' execution flow from super class to subclass?
   - Will non-variable & blocks are executed from both super & sub classes?
   - Will constructor executed from both super class & sub class?

Sub class object & constructor execution in inheritance interview questions

1. What a subclass object contains?
2. Will super class object also created, when subclass object is created?
3. Prove that super class object is not created when subclass object is created?
4. When super class object is not created, why constructor is executed from superclass?
5. How superclass constructor is called from subclass constructor in its object creation?
6. What is the use of super()?
7. Who will place super() in every subclass constructor?
8. What is the constructor by default executed from superclass, why?
9. How can we execute superclass parameterized constructor from subclass?
10. Which constructor is executed from super class for super() and super(5) call?
11. What is the first check point we must verify in solving inheritance bits?
12. If super class have only parameterized constructor & subclass is empty class, then will that subclass compiled successfully?
13. If super class has only private constructor, will inheritance possible?

super keyword interview questions

1. What is the use of *super* keyword?
2. Is *super* keyword an *operator* or a *variable*?
3. What is the type of super keyword?
4. What are the 2 usages of super keyword?

5. What is the difference between "super." & "super()"?
6. Why "super()" must be the first statement in a constructor?
7. Can we place "super()" in a method?
8. Can we place "super()" at any line in a constructor?
9. Can we place "super." in a constructor?
10. Can we place "super." at any line in a method or constructor?
11. What are the differences between *this* & *super* keywords?
12. What is the data type of *this* & *super* keywords?
13. To whom *this* keyword & *super* keyword is referring in subclass object?
14. What are the rules in using super keyword?
15. Can we use super keyword in static method?
16. In a non-static method, can we call static method using super keyword?
17. Can we print super keyword?
18. Can we return super keyword from a method?
19. Can we pass super keyword as a method argument?
20. Internally, in a non-static method how many referenced variables are created?
21. Draw subclass object memory diagram with super class, sub class non-static variables & with their referenced variables this & super keywords?
22. What is stored in super keyword,
    is it subclass object reference or superclass object reference?
23. Prove that super keyword contains sub class object reference?

Variable hiding interview questions

1. What is the meaning of variable hiding?
2. Can we create a variable in the sub class with a super class variable name?
3. Give me object memory diagram of a sub class object with variable hiding?

4. What is the difference in storing
   a. sub class object reference in sub class type referenced variable &
   b. sub class object reference in super class type referenced variable
5. What are the 2 ways we have for differentiating a variable from superclass & subclass?
6. **When we access a variable, its value is read whether from super class or from subclass. Is it resolved based on referenced variable type or based on object type?**
7. Assume the variable "x" is defined in both super class "Example" & in its subclass "Sample", from which class that variable "x" is accessed in the below cases

```
Sample s = new Sample();
Example e = new Sample();
```

x variable is accessed from

1. s.x
2. e.x
3. this.x
4. super.x

8. Can we access static variable using super keyword?
9. How can we access super class static variable from subclass static methods?
10. Can we create a variable in subclass with super class name but with different data type and modifiers?
11. How super class static variable is shared to all its subclasses?
12. How super class non-static variable is shared to all its subclasses?
13. At execution time, if we modify super class static variable value in one subclass, is it affected to all other subclasses?
14. At execution time, if we modify super class non-static variable value in one subclass, is it affected to all other subclasses?

Method hiding & overriding interview questions

1. What is the meaning of method hiding?
2. What is the meaning of method overriding?
3. When should we override super class method in sub class?

4. In subclass when a method is considered as overriding method?
5. What are the 4 rules we must follow in method overriding?
   1. Can we change return type of overriding method?
   2. What is co-variant return method?
   3. Can we add or remove static keyword of overriding method?
   4. Can we decrease accessibility modifier of overriding method?
   5. Can we override private method, if we override will it leads to CE?
   6. If super class method is not throwing a checked exception, can we throw checked exception in overriding method?
   7. If super class method is throwing a checked exception, is it mandatory we should throw this checked exception in overriding method also?
6. Why overriding rules are created, what will be the problem if we do not follow overriding rules?

7.	Can we override	private method	as non-private?
		Non-private method	as private?
8.	Can we override	public method	as non-public?
		Non-public method	as public?
9.	Can we override	Static method	as non-static?
		Non-static method	as static?
10.	Can we override	final method	as non-final?
		Non-final method	as final?
11.	Can we override	Abstract method	as non-abstract?
		Non-abstract method	as abstract?
12.	Can we override	native method	as non-native?
		Non-native method	as native?
13.	Can we override	synchronized method	as non-synchronized?
		Non-synchronized method	as synchronized?

14. Can we override strictfp method as non- strictfp?
    method as strictfp?
15. When we call static and non-static methods, their execution is resolved whether based on referenced variable type or based on the object stored in the referenced variable?
16. From subclass, how can we execute an overridden method from superclass?
17. From a user class, how can we execute an overridden method from superclass?
18. Assume the method "m1()" is defined in both super class "Example" & in its subclass "Sample", then from which class this method "m1()" is executed in the below cases

	Accessed from	
	if m1() is static	if m1() is non-static
1.	s.m1()	
2.	e.m1()	
3.	this.m1()	
4.	super.m1()	

19. Why static method is executed from superclass & why non-static method is executed from subclass, when we call them using super class referenced variable by storing subclass object reference?
20. How can we execute a non-static overridden method from superclass?
21. How can we execute a static hidden method from superclass?
22. How can you write a program to develop below scenarios
    When we call an overridden method using subclass object
    1. It should also execute from superclass
    2. First it should be executed from superclass
    3. First it should be executed from subclass
    4. Execute superclass method logic in middle of overriding method
    5. Add some more logic to super class method implementation
23. Assume we have main method only in superclass, in this case what will happen if we run subclass, is it leads to Error: main method not found?

24. Assume we have main method in both superclass & subclass, in this case how can we execute main method from both classes?

Method overloading interview questions

1. What is a method overloading?
2. When a method is considered as overloading method?
3. When we should overload a method, what are the advantages?
4. In overloading, why we should create methods with same name, if we create them with different names what will be the problem, will it be there any CE or RE?
5. What are the rules we must follow in overloading method?
    1. Is it possible to overload a method with the same parameter?
    2. Is it possible to overload a method by just changing return type of the method?
    3. Is it possible to overload a method with different return type & modifiers?
    4. Is it possible to overload a static method as non-static method?
    5. Is it possible to overload a non-static method as static method?
    6. Is it possible to overload a private method?
    7. Is it possible to overload a final method?
    8. Is it possible to overload a method in the same class?
    9. Is it possible to overload a method in subclass?
    10. Is it possible to override a overloaded method?
6. In how many ways we can show differences in parameter in overloading method?
7. On what basis overloaded method execution is resolved?
8. Identify what is the output in the below cases?
    1) Assume we have overloaded a m1() method with char & int parameters, then
        a. for m1('a') call which parameter method is executed?
        b. for m1(97) call which parameter method is executed?
        c. for m1('a' + 'b') call which parameter method is executed?

d. for m1((char)'a'+'b') call which parameter method is executed?
e. for m1((char)('a'+'b')) call which parameter method is executed?

2) What is the output from below program?

```
class CalleeImpl{

        public void foo(Object o) {

                System.out.println("Object parameter");

        }

        public void foo(String s){

        System.out.println("String parameter");

        }

        public void foo(Integer i){

        System.out.println("Integer parameter");

 }

}

public class OverloadingMystery{

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

9. If the given argument types parameter method is not found, Is it leads to CE?
10. If the given argument is matched with widening parameter type, what will compiler do?

11. Will compiler allow us to call narrow parameter type method?
12. How can we call narrow parameter method?
13. Identify what is the output from below cases?
    1) Assume we have overloaded m1() method with int & float parameters, then for m1('a') & m1(50L) calls which method is executed?
    2) Assume we have overloaded m1() method with byte & short parameters, then for m1(5) call which method is executed?
    3) Assume we have overloaded m1() method with byte & int parameters, then for m1(5) call which method is executed?
    4) Assume we have m1() method with only byte parameter, What will be the output fro the methd calls m1(5), m1('a') & m1((byte)5), m1((byte)'a')?
14. When will we get ambiguous error in calling overloading methods?
    1) What will happen if the method arguments are matched with widening parameters in multiple overloaded methods with different order?
    2) Assume we have overloaded m1() method with String & Integer parameters, then for m1(null) which parameter method is executed?
    3) Assume we have overloaded a m1() method with Object & String parameters, then for m1(null) which parameter method is executed?
15. How compiler will resolve the overloaded method execution, when it is also overloaded & overridden in subclass?
    1. Assume we called a overload method using subclass type referenced variable, then identify from which class this method is executed, also identify will you get any CE in the below cases
        1. given argument type parameter method is only available super class
        2. given argument type parameter method is in found in subclass also in super class

3. given argument type parameter method is in found in super class, widening parameter method found in subclass

2. Assume we called a overload method using superclass type referenced variable, then identify from which class this method is executed, also identify will you get any CE in the below cases
   1. Given argument type parameter method is available in superclass
   2. Given argument widening type parameter method is available in superclass, given argument type parameter method available in subclass
   3. Given argument narrow type parameter method is available in superclass, given argument type parameter method available in subclass
   4. Given argument widening type parameter method is overridden in subclass
   5. Given argument widening type parameter method is overridden in subclass, in subclass also we have given argument type parameter method

Ex:

Assume in parent class we have m1(long) method & in child class we have m1(int) method, then which method is executed for below method calls	Assume in parent class we have m1(int) method & in child class we have m1(long) method, then which method is executed for below method calls
1. child.m1(50) 2. child.m1('a') 3. child.m1(50L) 4. parent.m1(50) 5. parent.m1('a') 6. parent.m1(50L)	1. child.m1(50) 2. child.m1('a') 3. child.m1(50L) 4. parent.m1(50) 5. parent.m1('a') 6. parent.m1(50L)

13. List out all differences between
    i. Method overriding & method hiding
    ii. Method overriding & method overloading
14. Who will resolve hidden method execution, will it compiler or JVM?
15. Who will resolve overridden method execution, will it compiler or JVM?
16. Who will resolve overloaded method execution, will it compiler or JVM?
17. Why compiler resolves hidden & overloaded methods?
18. Why JVM resolves overridden method execution?
19. Identify valid *overriding* and *overloading* methods from the below methods list

```
class Example
{
  void m1(int a , int b){}

  void m2(){}

  private void m3(String s){}

  protected void m4(double d, float f){}

  public void m5(){}

  public void m6(int x , float f){}

  static int m7(){return 10;}

  void m8(){}

}
```

```
class Sample extends Example
{
  void m1(int x , int y){}

  public void m2(){}

  public int m3(String s){return 10;}

  public float m4(double d, float f){return 20.34f;}

  void m5(){}

  public static void m6(float f , int x){}

  int m7(){return 30;}

  static void m8(){}

}
```

Constructor overloading & chaining Interview Questions

1. What is constructor overloading?
2. What is constructor chaining?
3. Can we override constructor?
4. If we overload constructors, which constructor is executed?
5. When should we overload constructor?
6. What are the rules we must follow in overloading a constructor?

7. How can we chain subclass constructors with super constructors?
8. How can we chain subclass overloaded constructors?
9. When we invoke a constructor using this() or using super(), will new object created?
10. Explain this() execution flow?
11. Write a program to count number of objects created from a class?
12. What are the rules we must follow in constructor chaining?
    1) Why this() or super() call should be placed as first statement in a constructor?
    2) Can we place both this() & super() in the a constructor?
    3) Can we place this() or super() twice in a constructor?
    4) Can we place this() or super() in a method?
    5) Can we place this() call in the same matching parameter constructor?
    6) When will you get CE: recursive constructor call?
    7) For example in a no-arg constructor, why we cannot place *this()* call but we can place constructor call with new keyword?
    8) In a subclass, why we should not place this() in all constructors, at least one constructor should contain super()?

Final method interview questions

1. What is final method, what is the rule on final method?
2. What is the use of final method?
3. When should we declare a method as final?
4. Will final method inherited to subclass?
5. Can we call and execute final method in subclass?
6. Can we override a final method in subclass?
7. Can we overload a final method in subclass??
8. Can we override a non-final method as final method in subclass?
9. What is the difference between private method and final method?
10. Can we declare main method as final?
11. If we declare main method as final, can we define main method in its subclass?
12. How can we execute static & non-static members of this subclass from this subclass?

Final class interview questions

1. What is a final class, what is the rule on final class?
2. What is the use of final class?
3. When should we declare a class as final?
4. Can we create object from a final class?
5. If a class is declared as final, are all its members final?
6. Can we modify final class variables?
7. Can we override final class methods?
8. How can we stop create a subclass from a class without declaring it as final?
9. In how many ways we can stop creating a subclass from a class?
10. What is the difference between declaring a class as final and declaring all its constructors as private?

Abstraction interview questions

1. What is an abstraction, advantages of abstraction?
2. What is the meaning of hiding implementation details?
3. Is abstraction an OOP principle?
4. In which stage of project lifecycle abstraction is used?
5. How many types of classes we will develop as per abstraction in object design?
6. What is will be the problem if we do not follow abstraction in an object design?
7. How can we develop abstraction in java?

Abstract Keyword Interview Question

1. What is an abstract keyword?
2. What is the use of abstract keyword?
3. To what programming elements abstract keyword is allowed & not allowed?
4. Why abstract keyword is not allowed to variable, constructor, final class & enum?
5. What is the difference in declaring method as abstract & class as abstract?
6. What are the modifiers not allowed in combination with abstract keyword?

Abstract Method Interview Questions

1. What is an abstract method?
2. What is the use of abstract method?
3. What are we doing by declaring a method as abstract?
4. In what type of class the abstract method is allowed?
5. What are the rules we must follow in creating an abstract method?
    1) Can we create a method without body?
    2) Can we provide body to abstract method?
    3) Can we declare abstract method as
       private, static, final, native, synchronized, strictfp?
    4) What are the modifiers only allowed to abstract method?
    5) Can we use abstract keyword two times?
    6) Can an abstract method have return type other than void?
    7) If abstract method as non-void return type, who will return value?
    8) Can an abstract method throw exception?
    9) Can we call abstract method?
    10) Can we execute abstract method?
6. In which class abstract method contains its logic?
7. How can we provide choice to subclass in overriding a method?
8. How can we force subclass to override a method?
9. How can we stop overriding a method in subclass?
10. Why we must declare a class as abstract if it contains abstract method?

Abstract Class Interview Questions

1. What is an abstract class?
2. What is the use of abstract class?
3. What are we doing by declaring a class as abstract?
4. When should we declare a class as object?
5. When should we use abstract class in project?
6. What are the rules we must follow in declaring abstract class?
    1) Can we declare abstract class as final, strictfp?
    2) What are the modifiers only allowed to abstract class?

7. In addition to abstract method,
   What are the other members we can create in abstract class?
8. Can we compile & execute abstract method?
9. Can we instantiate abstract class?
10. Why cannot we instantiate abstract class?
11. Can we create referenced variable using an abstract class?
12. What is the use of abstract class referenced variable, when we cannot instantiate it?
13. Can abstract class contain constructor?
14. Why constructor is allowed in abstract class when it cannot be instantiated?
15. How can we execute non-static members of abstract class?
16. What are the rules we must follow in creating subclass deriving from abstract class?
    1) Is mandatory to override all abstract methods declared in abstract class?
    2) How can we create subclass by not overriding some the abstract methods?
    3) What is the meaning of concrete subclass?
17. Explain Abstract class's subclass execution flow?
    1) When we load subclass will static members executed from abstract class also?
    2) When we create subclass object will non-static members executed from abstract class also?
    3) Give me the memory diagram of an abstract class's subclass object?
18. Is it mandatory that abstract class should contain abstract method?
19. Can we declare concrete class as abstract, what is the use of it?
20. Can you give me predefined classes which are declared as abstract without having abstract methods?
21. How should we define a class to allow subclass also to allow instantiation?
22. How should we define a class not to allow subclass but should allow instantiation?
23. How should we define a class to allow subclass but not to allow instantiation?

24. How should we define a class not to allow subclass and not to allow instantiation?
25. In an abstract class can we declare all its constructors as private?

Interface Interview Questions

1. What is an interface?
2. What is the use of interface?
3. How can we develop full abstraction?
4. When we have abstract class to declare abstract method, why do we need interface?
5. When should we choose interface, abstract class, concrete class, & final class?
6. What are the rules in declaring interface?
    1) What is the keyword we must use for creating interface?
    2) What are the members allowed inside an interface?
    3) How can we save interface?
    4) Can we compile interface source file?
    5) Can we execute interface?
    6) What are the modifiers allowed to interface?
    7) Can we declare interface as final & strictfp?
    8) Can we create interface deriving from other interface?
    9) Can we create interface deriving from a concrete class or abstract class?
    10) Is it mandatory to declare method as public, abstract in interface?
    11) Is it mandatory to declare interface variable as public, static, final?
    12) Like java.lang.Object, is there any default super interface for every interface?
    13) What is the default accessibility modifier of interface & its members?
    14) Can we declare interface method as private or protected?
    15) **Why interface members are by default public why not protected?**
    16) Can we create a variable without assigning a value in interface?

    17) Can we create constructor in interface?
    18) Can we create main method in interface?
    19) Can we create static or non-static method in interface?
    20) Can we create static variable in interface?
    21) Can we create non-static variable in interface?
    22) If we create non-static variable in interface, will it leads to CE?
    23) Can we create inner class or inner interface?
    24) Can we instantiate interface?
    25) Can we create a referenced variable using interface, what is the use of it?
7. What is the rule in creating subclass from an interface?
    1) What is the keyword we must use in deriving subclass?
    2) Is it mandatory we must implement all abstract methods of interface?
    3) When we provide logic to interface methods in subclass, why we must use the word implements why should not use override?
    4) How can we create a subclass from an interface by not implementing all its methods?
    5) What is the accessibility modifier we must use in implementing interface method in subclass, why?
8. What is the meaning of adapter class?
    1) How can we create adapter class?
    2) Is adapter a keyword?
    3) What are the advantages with adapter class?
    4) In which package we will find predefined adapter classes?
9. Can we create an empty interface, what is the purpose of it?
10. What is the marker interface?
    1) What are the predefined marker interfaces?
    2) What is a custom marker interface?
    3) Can we create custom marker interface?
    4) What is the procedure to create custom marker interface?
11. What are the similarities and differences between interface and abstract class?
12. What is a default method in interface (Java 7 enhancement)?

Polymorphism Interview Questions

1. What is a polymorphism?
2. How can we develop polymorphism in Java?
3. What is the difference in implementing polymorphism
   1)using method overloading &
   2)using method overriding
4. In what stage of project development we implement polymorphism?
5. How many types of polymorphisms Java supports?
6. What is compile-time polymorphism (or) static polymorphism (or) early binding?
7. What is run-time polymorphism (or) dynamic polymorphism (or) late binding?
8. How can we find a method call comes under compile time (or) runtime polymorphism?
9. Why static method and overloaded method comes under compile-time polymorphism?
10. Why non-private, non-static method comes under run-time polymorphism?
11. If a method is overridden in subclass will it always comes under run-time polymorphism?
12. What are the conditions should meet to achieve runtime polymorphism?
13. What is the meaning of dynamic method dispatching?
14. **Why interface guarantees runtime polymorphism?**
15. **Why every business object design should starts with interface?**

LC-RP architecture based project development interview questions

1. How many types of objects will exist in real-world business?
2. What are the three types of classes we must develop for an object as per business?
3. When should we create super class & subclass in project?
4. What is the meaning of specification, service provider, and service user?
5. Explain LC-RP architecture in designing, developing & using object operations?

6. What is Coupling and cohesion?
7. What is the meaning of tight coupling & loose coupling?
8. What is the meaning of high cohesion & low cohesion?
9. Why we should develop low coupling & high cohesion in project?
10. When exactly superclass, subclass should be defined?
11. Design and develop Mobile-SIM objects using LCRP architecture?
12. Design and develop ATMMachine-ATMCard objects using LCRP architecture?
13. Understand Gmail project architecture using MVC and LCRP?
14. What is the right design to enhance the project to add more operations to an existed business object (or) to modify an existing operation of an object?
15. *For example* what is the right design to add 3G operation to SIM object?

Types Of Inheritance Interview Questions

1. How many types of inheritances Java supports?
2. Will Java supports multiple inheritance?
3. What are the keywords we use to develop inheritance?
4. What are the differences between extends & implements?
5. Can we use implements keyword to derive a class from another class?
6. What is the default super class to all java classes?
7. What is single level inheritance?
   1) Can we derive a class from another class?
   2) Can we derive a class from an interface?
   3) Can we derive an interface from another interface?
   4) Can we deriver an interface from another class?
   5) Is it possible to develop a class without inheritance?
   6) If we develop single class, is it really single class?
8. What is multi level inheritance?
   1) What is the meaning of direct super class?
   2) What is the meaning of indirect super class?
   3) In a subclass super keyword references to which superclass?
   4) Is it possible to execute indirect superclass method that is overridden in direct superclass?

5) If we create single level inheritance, is it really single level inheritance?

9. What is hierarchical level inheritance?
   1) How many subclasses can we develop from a single class?
   2) As a subclass developer can we stop multilevel inheritance?
   3) As a subclass developer can we stop hierarchical inheritance?
   4) What is the default inheritance you develop unknowingly in your practice?
   5) Assume you have developed 3 classes A, B, C as normal classes, is there any inheritance?
10. What is hybrid inheritance?
   1) What is the default inheritance we develop in project unknowingly?
11. What is multiple inheritance?
   1) Why Java does not support multiple inheritance with classes?
   2) How inheritance will solve multiple inheritance ambiguous problem?
   3) What is the syntax to create multiple inherence with interfaces?
   4) Can we derive a class from multiple classes?
   5) Can we derive a class from multiple inheritances?
   6) Can we derive a class from a class & interface?
   7) What is the syntax to derive a class from a class & interface?
   8) Why extends keyword should use before implements keyword?
   9) If we derive a class only from interface is it also deriving from a class?
   10) Why compiler will allow us to cast a class referenced variable to interface even through it is not subclass, where as it is not possible to cast final class variable to interface why? Identify CE in below case:

```
interface A{}

class B{}
```

```
B b = new B();

A a1 = b;
```

```
C c = new C();

A a1 = c;
```


11) Why compiler allow us to call java.lang.Object class method using interface type referenced variable, even though interface is not a subclass of Object class?
For example: A a = null; a.toString(); here we do not get CE?

12. Identify is multiple inheritance is possible in the below cases?
   1) **both interfaces have a method with same prototype**, how many times we must override it?
   2) both interfaces have a method with same name, but with different parameters
   3) both interfaces have a method with same signature, but with different return type as below
      i. return types are primitive types
      ii. return types are compatible referenced types
      iii. return types are incompatible referenced types
   4) both interfaces have a method with same signature, but with different checked exceptions type
   5) **both class and interface have method with same prototype,** how many times we must override it?
   6) both class and interface have a method with same signature but with different accessibility modifiers
   7) class method is declared as static?
13. How can we access interface variable in subclass?
14. How can we solve ambiguous error if both interfaces have a variable with same name?

Garbage Collection Interview Questions

1. What is the meaning of garbage?
2. What is the meaning of garbage collection?
3. When will we get *java.lang.OutOfMemoryError?*
4. How can we solve OOME, is it by using try-catch?
5. Do we have destructors in Java to destroy objects?
6. How can we destroy an object?
7. What is a garbage collector?
8. What type of thread is Garbage collector thread?

9. When gc will destroy the object?
10. When an object is said to be eligible for garbage collection?
11. What are different ways we have to unreference the object?
12. What is the meaning of Island of Isolations?
13. Garbage Collector is controlled by whom?
14. Can we force Garbage Collector to start?
15. How can we request JVM to start Garbage Collector?
16. How is the difference between System.gc() & Runtime.gc()?
17. What is the algorithm JVM internally uses for destroying objects?
18. Which part of the memory is involved in Garbage Collection? Stack or Heap?
19. What is the need of finalize() method, in which class it is defined?
20. What is the meaning of clean-up code?
21. Can we call finalize method, if yes, then will the object destroyed?
22. If an exception is thrown during the execution of the finalize() method, will the object destroyed?
23. An unreachable Java object, can it become reachable again?
24. How many times will the garbage collector call the finalize() method for an object?
25. If an object becomes eligible for Garbage Collection and its finalize() method has been called and inside this method the object becomes accessible by a live thread of execution and is not garbage collected. Later at some of point the same object becomes eligible for Garbage collection, will the finalize() method be called again?
26. How to enable/disable call of finalize() method?

Array Interview Questions

1. What is an array, what is the use of an array?
2. What are the limitations of an array?
3. What are the 3 syntaxes to create an array?
4. In Java, is array a primitive type (or) referenced type?
5. In Java, how can we create a heterogeneous array object?
6. In Java, where should we specify array size?
7. What is the meaning of dimension?
8. How can we find the length of an array?

9. What is the differences between length & length()?
10. Explain difference in storing primitive values and objects in arrays?
11. Can we cast array objects?
12. Explain various situations we get below errors?
    1) missing dimension
    2) possible loss of precision
    3) incompatible types
    4) NegativeArraySizeException
    5) ArrayIndexOutOfBoundsException
    6) ArrayStoreException
13. How can we pass multiple values as argument & return type from a method?
14. What is meaning of an anonymous array, how can we create it?
15. Where anonymous array is useful?
16. If we unreference an array how many object are eligible for garbage collection?
17. If we declare array referenced variable as final, will array object also final?
18. Can we declare array object as final?
19. Explain all four types of array referenced variables?
20. How many types of arrays java supports?
21. What is the meaning of array of arrays?
22. What is the difference in the below two lines?
    1) int[] ia = new int[3][2];
    2) int[] ia = new int[3][];
    3) int[] ia = new int[][];
23. What is a Jogged Array?
24. What is a Var-arg method, what are the various rules applicable on Var-arg parameter?
25. Is Var-arg is the replacement of single dimensional array in method parameter?
26. In Var-arg why only three dots(...) are allowed?
27. What is the diff in between array parameter method & Var-arg parameter method?

Package Keyword Interview Questions

1. What is a package & sub package?
2. What is the use of package & sub package?
3. What is the syntax to create package & sub package?
4. What is the difference between?
   1) package p1;
   2) package p1.p2;
5. What is the folder structure created for below package statement
   1) package p1.p2.p3;
6. Can we place package statement anywhere in the source file?
7. Can we place more than one package statement?
8. Will package keyword create package fo lder in hard disk, what will it do?
9. When should we call a folder is a package?
10. What is the use of –d option of *javac* tool?
11. If java file does not have package statement will –d option throws any error?
12. What is the right syntax to compile and execute a packaged class?
13. What is the difference between
    1) javac Example.java
    2) javac –d . Example.java
    3) javac –d C:\test Example.java
14. What is the difference between
    1) java Example
    2) java p1.Example
15. Assume we have created two java files with same class name without package statement, after compiling java files how many .class files are existed in the current working directory and what is the output we get when we run first class after second java file compilation?
16. In the above scenario how can we have both .class files?
17. We know that pubic class name & its java file name should be same, then how can we create two java files with same public class name?
18. When *should* we create *package manually*?
19. How can we create two different public classes with same name in same package?

Import Keyword Interview Questions

1. How can we access one package classes from other package classes?
2. What is the difference between fully qualified name & import statement?
3. Can we place import statement anywhere in the source file?
4. What is the difference between
   1) import p1.*;
   2) import p1.A;
5. Why "import p1.A;" syntax is recommended in project?
6. If we place "import p1.A;" can we create a "class A{}" in the current source file?
7. Assume in the current package p1 & imported package p2 we have "A.class", then for the below import statements from which package "A.class" is loaded
   1) for import p2.*; Î
   2) for import p2.A; Î
8. Will import statement load a class into JVM?
9. Can we place more than one import statement?
10. What is the default predefined java package that we no need to import?
11. What is meaning of class naming conflict?
12. When should we use both import statement & fully qualified name of a class?
13. What is the difference between import statement & classpath?
14. In how many ways we can set classpath?
15. What is the difference in below options in setting classpath
    1) java option –cp
    2) java option –classpath
    3) set classpath
    4) permanent settings
16. If we import parent package will sub package classes also imported?
17. What is the difference between
    1) import p1.*;
    2) import p1.A;
    3) import p1.p2.*;
    4) import p1.p2.A;

18. What is the compiler algorithm for searching .class files?
19. Why public class name and java file name should be same?
20. What is the importance of "." in classpath variable?
21. Assume we have placed Example.class file in Current working directory & C:\test folder, then identify from which folder compiler & JVM loads Example.class file for the below classpath settings

    Loaded from which folder?

    1) set classpath=C:\test
    2) set classpath=.
    3) set classpath=.;C:\test
    4) set classpath=C:\test;.
    5) set classpath=NareshIT;C:\test
    6) set classpath= NareshIT;;C:\test
    7) set classpath=NareshIT;C:\test;
    8) set classpath=NareshIT
    9) set classpath=NareshIT;
22. What is difference in placing ; in the middle of the paths & at end of the path?
23. What is difference in placing single ; and multiple ;s in the middle of the path?
24. Can we access a packaged class from a non-packaged class?
25. Can we access a non-packaged class from a packaged class?
26. Can we access a non-packaged class from a non-packaged class?
27. Can we access a packaged class from a packaged class?

Static Import Interview Questions

1. What is the need of static import?
2. What is the syntax of static import?
   Is it "static import" or "import static"
3. In which version static import is added?
4. What is the difference between import & static import?
5. Using static import can we access non-static members of class?

6. How can we differentiate imported class static members from local class members if both have same name?
7. What is the difference between below statements?
   1) import p1.*;
   2) import p1.A;
   3) import static p1.A.*;
   4) import static p1.A.a;
   5) import static p1.A.m1;
8. In the above import statement, if m1 is a non-static method in class A, will compiler throw error?
9. Can you write a program to print data using only "out.println()"?
10. If we use static import, will it give access class & its non-static members?
11. **In the below program at which line number CE is raised?**

```
1)  package p1;
2)  public class A{
3)    public static void m1(){}
4)  }

5)  import p1.A.*;
6)  class WrongAccess{
7)      public static void main(String[] args){
8)            m1();
9)            A.m1();
10)           p1.A.m1();
11)          A a1 = new A();
12)          p1.A a1 = new p1.A();
13)   }
14) }
```

12. **Find out valid syntaxes from the below list**

1) import java.lang.*;
2) import java.lang.System;
3) import java.lang.System.*;
4) import java.lang.System.out;
5) import static java.lang.System.*;
6) import static java.lang.System;
7) import static java.lang.System.out;
8) import static java.lang.System.out.*;
9) static import java.lang.System.out;

Jar File Interview Questions

1. What is a jar file?
2. What are the different types of compressed files?
3. What is the difference between jar, war, ear files?
4. What is the difference between jar, zip file?
5. What is the use of *jar* command, where is it available?
6. What are the different options of *jar* command?
7. What are the options to create jar file?
8. What are the options to extract jar file?
9. What are the options to list jar files?
10. What is the importance of character '*' in jar creation?
11. How can we access classes from a jar file?
12. How can we set jar file in classpath for accessing classes in all packages of project?
13. What is classpath of a war file; means where should we copy *jar* file in project war?

Executable Jar File Interview Questions

1. What is an executable jar file?
2. What is difference between executable jar file and normal jar file?
3. How can we create executable jar file?
4. What is the need of *manifest.mf* file in jar file?
5. What are the different property names available in manifest.mf file?
6. What is the property name to configure class name in manifest.mf file?
7. What is the *java* command *option* to execute executable jar file
8. How can we execute *jar* file using a batch file?
9. How can we execute jar file by double clicking the jar file?

API interview questions

1. What is the meaning of API and API Documentation?
2. When a class is called API?
3. What are the differences between API & API documentation?
4. Why API documentation file format is *.html* file?

5. What is the use of *javadoc* command?
6. How API is distributed with software?
7. How API Documentation is distributed?
8. For whom sake API and API documentation is generated?
9. What files should be updated in Classpath environment variable?
10. How can we generate API documentation for our user defined classes?
11. What is the command and syntax to generate API documentation for our classes?
12. Should we compile source file to generate API Documentation file?
13. Can we generate API documentation for non-public class?
14. What members of the class are included in API documentation of this class?
15. Why private & default members of the class are not included in API documentation?
16. Can explain API Documentation file structure?
17. How many html files are generated by javadoc tool for the given class?

java.lang package interview questions

1. Why java.lang package is called default package?
2. Why we no need to import java.lang package when access its classes?
3. If we use the class String in our class, how compiler will link it from java.lang package?
4. Given me some of the important classes from java.lang package?
5. What is the use of System class?
6. What is the use of Runtime class?
7. What is the use of Class class?
8. What is the use of ClassLoader class?
9. What is the use of Process class?
10. What is the use of Math & StrictMath classes?
11. What is the use of Comparable interface?
12. What is the use of Cloneable interface?
13. Using Runtime class method, Write a program
    1) to start "notepad" from java program?
    2) to find JVM memory status using Runtime class method?

14. Using Math class method, write a program
    1) to find square root of a given number?
    2) to find cube root of a given number
    3) to find a power of a number with the given number?
    4) to find max of given two numbers
    5) to find min of given two numbers

java.lang.Object class interview questions

1. Why Object class is the super class for all java classes?
2. Why its name is chosen as Object?
3. How many methods Object class contains?
4. Explain the purpose of all 11 methods of Object class?
5. How many methods we can override in subclass from Object?

1. **hashCode() method interview questions**

1. What is hashCode?
2. Is hashCode & reference both are same?
3. How can we retrieve hashCode of an object?
4. How many ways we can generate hashCode of an object?
5. What is returned from hashCode() method from Object class?
6. How can we generate hashCode() of an object using its data?
7. What is hashing algorithm?
8. If we override hashCode() method, will object also contains referenced based hashcode? How can we get JVM generated hashcode?
9. What is the difference between hashCode() & System.identityHashCode() methods?
10. Can two objects have same hashCode?
11. If we change object data will object hash code also be changed?
12. In what programming hashCode() method is called internally?

2. equals() method interview questions

1. What is the use of equals() method?
2. What are the different ways we can compare two objects of a class?

3. How can we compare two objects using their references?
4. How can we compare two objects using their data?
5. How equals() method will compare objects in Object class?
6. How can we compare objects using their data using equals() method?
7. What are the different rules we must follow in overriding equals method?
    1) What value should return for incompatible objects comparison?
    2) What value should return when null is passed as argument?
8. What is the contract between hashCode() & equals() method overriding?

9. Why equals() method is defined when we have == operator?
10. What will happen when we compare incompatible objects using "==" operator and equals method?
11. What is the output we get when we compare

	using == operator	using equals method
1) two nulls	Æ	
2) null and object	Æ	

12. Can we use equals() method for comparing primitive values?
13. List out all differences between "==" operator & equals() method?
14. In what programming equals() method is called internally?

3. getClass() method interview questions

1. What is the use of getClass() method?
2. What is the need of ***java.lang.Class?***
3. What is meaning of runtime class and runtime class object?
4. How can we get object's runtime class object reference?
5. Write a program to get any given object's class name?

4. toString() method interview questions

1. What is the use of toString() method?
2. What is printed when we print object?
3. How println method retrieves current printing object information?
4. What is the default implementation of toString() in java.lang.Object class?
5. When should we override toString() method in subclass?
6. How can we return all values of an object from toString() for displaying on console?
7. When we print String class object its data is printed, when we print our class object why classname@hashcode is printed?
8. **When we must override hashCode(), equals(), toString() methods in subclass?**

5. clone() method interview questions

1. What is the use of clone() method?
2. What is the meaning of cloning?
3. What is the rule in cloning an object?
4. What is the use of Cloneable interface, why is it empty interface?
5. Is clone() method defined in Cloneable interface?
6. Will clone() method return new object or same object, will new reference & hashCode?
7. Why clone() method is declared as protected?
8. What are the programming rules we must follow in calling a clone() method?
    1) Can we call clone() method on an object in its own class?
    2) Can we call clone() method on an object in another class?
    3) Why should we handle CloneNotSupportedException in calling clone() method?
    4) If the object is not Cloneable type, what will happen when we all clone() method using this object?
    5) Why should we cast clone() method returned object to current object's class?
9. **What is the procedure to override clone() method?**

10. What is the difference in cloning an object
    1) by containing primitive variables &
    2) by containing referenced variables
11. When we clone an object with HAS-A relation, will its internal object also cloned?
12. When we clone a subclass object, is its super class also should be Cloneable type?
13. When we clone a subclass object, will internal objects of superclass also cloned?
14. What are the 2 types of cloning mechanisms?
15. What is a shallow cloning?
16. What is a deep cloning, how can we implement it?
17. What type of cloning is implemented in Object class's clone() method?
18. **What are the two situations force you to override clone() method in project?**

7. finalize() method interview questions

1. What is the use of finalize() method?
2. What is the meaning of clean-up code or resource releasing logic code?
3. Give me some project scenarios to explain about clean-up code?
4. What is the default implementation given for finalize() method in Object class?
5. When should we override finalize() method in subclass?
6. Why finalize() method is declared as protected?
7. Who will call finalize method, and when?
8. Can programmer call finalize() method explicitly?
9. If we call finalize() method, will object is destroyed?
10. How many times finalize() method is called for an object?
11. How can we enable or disable calling a finalize() method?
12. Why finalize() method has "throws Throwable"?
13. Can an unreachable Java object become reachable again?
14. If an object is become referenced after finalize() method execution, will gc destroy it?

8, 9, 10, 11 check Multithreading interview questions

Why wait(), notify(), and notifyAll() methods are given in Object class why not in Thread class?

String Handling interview questions

String Literal Interview Questions

1. What is a string?
2. What is string handling?
3. In how many ways we can store string data?
4. Why String class is given when char array is already available?
5. Why StringBuffer class is given when we have String class to store string data?
6. What is meaning the immutable & mutable objects?
7. Can we create custom (user defined) immutable object?
8. Why StringBuilder is given when we have StringBuffer?
9. What is the meaning of thread-safe and non-thread safe objects?
10. Can we create thread-safe and non-thread safe objects?
11. Is String object thread safe?
12. What is the definition of String, StringBuffer, and StringBuilder?
13. What is the relation between Sting, StringBuffer and StringBuilder classes?
14. Why String, StringBuffer classes are created as siblings, why not with IS-A relation?
15. Why StringBuffer, StringBuilder classe s not created with IS-A relation?
16. Can we assign string literal directly to StringBuffer or StringBuilder type variables?
17. How can we store sting literal in StringBuffer and StringBuilder objects?

String Class Object Creation Interview Questions

18. What are the different ways to construct String Object?
19. **What are the differences in creating String object using the String literal and using the constructor?**

20. How many objects are created in "literal" assignment & in constructor assignment?
21. When should we use constructor for creating String object?
22. On what basis can we count how many string objects are created in JVM?

String pooling interview questions

23. What is String Pooling; and Why String pooling?
24. Where String literal objects are stored and where String constructor objects are stored?
25. Draw JVM Architecture to show complete memory structure of String objects creation in both approaches?
26. Why String object is created as immutable?
27. Why String poling is implemented for String objects only?
28. Is String referenced variable immutable?
29. How can we make string reference variables as immutable?

String Constant Pool Area (*SCPA*) interview questions

30. What is a String Constant Pool Area?
31. Who creates SCPA object & When?
32. Is SCPA created separately for every cl ass in the project or is it created one per JVM? Prove your answer with a program?
33. Is SCPA eligible for garbag e collection, if not why?
34. Is String literal object eligible for gc, if not why?
35. Is String constructor object eligible for gc, if yes why?
36. Prove that string literal object is not eligible for gc even if it is unreferenced explicitly?

String class constructors interview questions

37. How many important constructors do we have in String class?
38. What happen if we pass ***null*** as argument to String class constructor?
39. Can we create null string object?
40. Can we create null string referenced variable?
41. Can we create empty string object?
42. Are both empty string & null string objects same?

Custom Immutable and mutable objects creation interview questions

43. What is immutable and mutable object?
44. Given some of the inbuilt immutable & mutable objects?
45. Can we develop custom immutable object, if yes what is the procedure?
46. Can we develop immutable variable, if yes how?
47. Assume we called a *mutator method* on *an immutable object*. Then it definitely returns a new object with result. Then what is the output we get in the below cases:
    - Case #1: Returned object is not assigned to any variable
    - Case #2: Returned object is assigned to new referenced variable
    - Case #3: Returned object is assigned to the same current object's variable

String operation methods interview questions

48. How can we find is string empty or not?
49. How can we find length of the String?
50. What is the difference between length & length()?
51. What is printed when we print String object, is it String@hashCode or its data?
52. In how many ways we can compare string objects?
53. How String Objects must be compared for equality?
54. What is meant by comparing string objects lexicographically, how it can be done?
55. When should we compare string objects for equality and lexicographically?
56. How can we read characters from the String?
57. How can we find the character case in the String object?
58. How can we find the position of a character or sub String?
59. If the given character is not available what value is returned from indexOf() method?
60. How can we find a character or a substring in the given string?

61. How can we find the string is startsWith or endsWith the given substring?
62. What is the difference between equals, contains, startsWith and endsWith methods?
63. How can we retrieve substring from the String?
64. If the start index and end index is same then what is returned from substring() method?
65. What will be returned from substring method if startIndex > endIndex?
66. How can we convert primitive values and objects to String object?
67. What is the limitation of String class object?
68. What is meant by concatenation, how Strings can be concatenated?
69. How can we convert string case to lower and upper case in the String?
70. How can we replace a character or substring in this String?
71. How can we remove string leading and trailing spaces?
72. When we call concat()/toLowerCase()/toUpperCase()/replace() & trim() methods, if current string object data is modified, will the result is stored in this current string object (or) will it return in the new object?
73. How can we split string into tokens?
74. How can we spit given string into words?
75. How can we find number of words in a string?
76. Write a program to print number of characters & words present in a given string?
77. What is returned in String[], if we pass empty string as argument to split method?
78. When is empty string is included as token in string[]?
79. What is returned in String[], if the argument string is not present in current string?
80. What is returned in String[], if the we pass same current string as argument?
81. Write a program to reverse words in the given string?
82. What are the operations we cannot perform on string using String object?

StringBuffer, StringBuilder classes interview questions

83. What is definition of StringBuffer, and StringBuilder?
84. What is need of StringBuffer, and StringBuilder?
85. What is the difference between StringBuffer & StringBuilder class?
86. When should we choose StringBuffer class & StringBuilder class?
87. What are the special operations can be performed on StringBuffer; those cannot be applied on String?
88. In how many ways we can create StringBuffer object?
89. In SB object creation what will happen if we pass *–ve* number or *null* as arguments?
90. What is the default capacity of String, StringBuffer and StringBuilder?
91. What is the meaning of default capacity?
92. What is the default capacity of StringBuffer, If its object is created
    1) using StringBuffer() constructor?
    2) Using StringBuffer(String) constructor?
93. What is the difference between capacity and length?
94. How can we find StringBuffer capacity and length?
95. How can we append, insert, delete, reverse, & override characters in the StringBuffer?
96. What happened when we append/insert characters more than its capacity?
97. How much the capacity is increased when size is greater than current capacity?
98. **In the SB capacity incrementing formula ((CurrentCapacity * 2) + 2) Why the SB incremented capacity result is added by 2?**
99. How can we control StringBuffer capacity?
100. What is the use of ensureCapacity() method?
101. What is the use of trimToSize() method?
102. What is the use of setLength() method?
103. What are the 7 differences between String and StringBuffer?
104. What is the right design to store and manipulate string data?

Wrapper Classes Interview Questions

1. What is the meaning of wrapping?
2. What is the use of wrapper class?

3. How many wrapper classes do we have?
4. What is a Number class, why is it an abstract class?
5. What is the super class for Character & Boolean classes?
6. What are the 6 conversions we can do using wrapper classes?
7. What are the constructors and methods do we have in all wrapper classes for performing all 6 conversions?
8. What are the constructors commonly available in all wrapper classes?
9. Why Character class does not have String parameter constructor?
10. When will you get java.lang.NumberFormatException?
11. Will we get NumberFormatException in Boolean class?
12. Can perform casting between wrapper classes?
13. How wrapper classes objects are compared using equals() method & == operator?
14. What is the hashCode returned from hashCode() method for wrapper classes object?
15. When we print wrapper class object, why its data is printed?
16. Explain Character class special methods to perform operations on characters?

Auto Boxing And Unboxing Interview Questions

1. What is the meaning of auto boxing and unboxing?
2. In which version this feature is added?
3. For what programming this feature is added?
4. Will JVM know about auto boxing & unboxing?
5. On what basis compiler changes primitive type to wrapper class object?
6. On what basis compiler changes wrapper class object to primitive type?
7. What is the method used in auto boxing?
8. What is the method used in auto unboxing?
9. What are the different cases in overloading a method with widening, AB & var-arg?
10. When we call a method by passing primitive value as argument, which method has highest priority is it widening parameter type or widening type or var-arg type?

11. Identify which parameter method is executed for the method m1(5) call?
    1) method is overloaded with (int) & (Integer) parameters
    2) method is overloaded with (long) & (Integer) parameters
    3) method is overloaded with (long) & (int...) parameters
    4) method is overloaded with (Integer) & (int...) parameters
    5) method is overloaded with (Integer...) & (int...) parameters
    6) method is overloaded with (Object) & (int...) parameters
12. Can we pass wrapper class object as argument to switch?
13. Can we pass String class object as argument to switch?
14. Can we do arithmetic operations using wrapper classes?
15. Write a program to add two integer numbers without using primitive variables?
16. In place of primitive types, Is it recommended to use wrapper classes in arithmetic operations?

Exception Handing Interview Questions

1. What is an exception & exception handling?
2. What is represented by an exception in Java?
3. What is the super class of all exception classes?
4. What is the difference between Error and Exception?
5. What is the difference between RuntimeException and Exception?
6. What is the exception message format printed by JVM?
7. What is the reason for raising an exception?
8. Why program is terminated when an exception is raised?
9. How can we print user understandable message to take correct value from user?
10. How can we stop abnormal termination?
11. How many keywords are given to work with exception?
12. What is the meaning of handling an exception?
13. What are the keywords used for handling an exception?
14. Explain when will we use try/catch/finally & throw/throws keywords?
15. Can finally catch an exception?
16. When should we use try/catch, try/finally & try/catch/finally?

17. Which of the combinations are correct?
    1) Is it possible to place try/catch, try/finally, try/catch/finally, try/finally/catch?
    2) Is it possible to place try with multiple catches?
    3) Is it possible to place try with multiple finally blocks?
    4) Is it possible to place only try, only catch, only finally?
    5) Is it possible to place a statement between try/catch/finally?
    6) Is it possible to place try/catch/finally at class level?
    7) Is it possible to place catch block with String or some other class as parameter?
    8) Is it possible to place multiple "try/catch/finally" blocks in the same method?
    9) Is it possible to place only try with resource?
    10) Is it possible to place catch with multiple parameters?
    11) Is it possible to place finally with parameter?
18. What are the rules in placing multiple catch blocks?
    1) Why do we need to create a multiple catch blocks?
    2) What is the parameter we must use to catch all exceptions using single catch?
    3) Can we place multiple catches in any order?
    4) Can we place multiple catches with same parameter block?
    5) Can we place catch block with parent class before the catch block with child class exception?
19. What are various methods given in Throwable class to print exception message?
    1) What is the difference between printStackTrace(), toString() & getMessage()?
    2) What is the difference in printing exception by JVM and printStackTrace()?
    3) Based on the printed message, how can you identify is the program terminated normally or abnormally?
20. Explain about exception default handler (EDH)?
21. What is the use of *finally* block?
22. What type of code we will write in finally block?
23. **When finally block is not executed?**
24. Explain *various cases exist in try/catch/finally* execution control flow?

25. Identify in the below cases will finally is executed or not?
    1) exception is not raised in try block?
    2) exception is raised in try block, catch block is matched?
    3) exception is raised in try block, catch block is not matched?
    4) exception is raised in try block, catch block is matched, also exception is raised in catch block?
    5) If try block has return statement will finally execute?
    6) If catch block has return statement will finally execute?
    7) try/finally without exception
    8) try/finally with exception
    9) In try block we have System.exit(0) will finally executed?
26. What is the effect if finally block has return statement?
    1) Can we place statements after finally block?
    2) If finally has return statement, how can we place statements after finally?
    3) How can we suppress exception raised in try block without using catch block?
    4) Exception is raised in try block, also exception is raised in finally block which exception is sent to calling method?
    5) Exception is raised in try block, finally has return statement will exception is sent to calling method?
    6) From try block value 30 is returned, also value 50 is returned from finally block, then which value is returned to calling method?
27. What is the Inner try/catch block?
    1) How can we catch an exception within the try or catch or finally block?
    2) The exception that is raised in the catch block, will it be caught by the same catch block?
28. Explain various cases exist in inner try catch execution control flow?
29. Identify which catch block will catch exception in the below cases?
    1) Exception is raised in outer try, inner catch & outer catch are matched
    2) Exception is raised in inner try, inner catch & outer catch are matched

    3) Exception is raised in inner try, inner catch is not matched, but outer catch is matched
    4) Exception is raised in inner try, inner catch and also outer catch are not matched
30. Explain various cases existed in Inner and outer finally block execution?
    1) No Exception is raised in outer try or in inner try blocks
    2) exception is raised in inner try, either inner catch or outer catch is not matched
    3) exception is raised in inner try, inner catch and outer catch are matched
    4) exception is raised in inner try, inner catch is not matched but outer catch is matched
    5) exception is raised in inner try, inner catch is not matched, but outer catch is matched, but inner finally has return statement
    6) return statement is placed in inner finally
    7) return statement is placed in inner finally and also outer finally
31. By writing try block what are we informing to compiler?
32. By writing catch block what are we informing to compiler?
33. By writing finally block what are we informing to compiler?
34. What are the various cases compiler considers in compiling a try/catch/finally blocks?
35. What is the output we get in below cases?
    1) A variable is created in try block & accessing it outside try block?
    2) A variable is declared outside try, initialized in try block, accessed it after try
    3) A variable is declared outside try, initialized in both try & in all catch blocks, accessed it after try/catch
    4) A variable is declared outside try, initialized in both try & in all catch blocks, accessed finally
    5) A variable is declared outside try by assigning value, reassigned only in try accessed it after try/catch

6) A variable is declared outside try by assigning value, reassigned only in try accessed in finally
7) return statement is placed only in try
8) return statement is placed in try & in all catch blocks
9) return statement is placed in try, in all catch blocks also in finally

36. What is exception propagation?
37. How long exception is propagated?
38. What is the use of throw keyword?
39. What is the use of throws keyword?
40. In a method where should we use throw and throws keywords?
41. What are the differences between throw and throws?
42. How many kinds of exceptions do we have in java?
43. What is checked and unchecked exception?
44. What are the exceptions comes under checked and unchecked exceptions?
45. What is the difference between checked and unchecked exceptions?
46. What is the difference between pure and partial checked exceptions?
47. What is the rule on pure checked exception and partially checked exceptions?
48. What are various rules on checked exception?
    1) We throw a checked exception and if we do not report or catch it then what will happen, will leads to CE or RE?
    2) Without throwing a checked exception from a try block, can we place a catch block with checked as parameter?
49. What are the various rules in overriding a method if it is throwing a checked exception
    1) Can we ignore throwing this exception?
    2) Can we throw the same checked exception?
    3) Can we throw its subclass exception?
    4) Can we throw its super class exception?
    5) Can we throw its sibling checked exception?
    6) Can we add more checked exception?
    7) Can we throw unchecked exception?

8) How should we override the method if overriding method required throwing some other checked exception?
9) What is the meaning of exception chaining?
10) What is the meaning re-throwing an exception?

50. What are the various rules we must follow in deriving a subclass if super class constructor throwing a checked exception
    1) If super class constructor throwing a checked exception, should subclass also throw same exception?
    2) Instead of throwing this exception, can we catch this exception in subclass constructor?
    3) Can we place super() in a try block?
    4) If a constructor throws exception, using this constructor can we create object at class level?
51. What is custom exception?
    1) What is the procedure for creating custom exception?
    2) Why custom exception should not be subclass of Throwable or Error?
    3) Why should we create custom exception as subclass of Exception or RuntimeException?
    4) How can we create custom exception as checked or unchecked exception?
    5) When we throw custom exception, is it mandatory to catch or report it?
52. Develop a Bank project with the custom exceptions *InvalidAmountException* & *InsufficientFundsException* to handle wrong operations done by customer in *deposit* and *withdraw* operations
53. What are the enhancements added in Java 7 for try & catch blocks?
    1) Explain The try-with-resources Statement
    2) Explain Multiple Exception Handling Syntax
54. Identify when will you get below 18 compile time errors
    1) try without catch, finally or resource
    2) catch without try
    3) finally without try
    4) ')' expected
    5) exception has already been caught

6) unreachable statement
7) unreported exception must be caught or declared thrown
8) exception has never thrown from corresponding try statement
9) try-with-resources not applicable to variable type
10) <identifier> expected
11) illegal start of type
12) incompatible types
13) cannot find symbol
14) variable might not have been initialized
15) missing return statement
16) illegal start of expression
17) ';' expected
18) method in subclass is not overriding method in superclass

55. What is the difference between exception & System.exit();
56. What is the difference between try-catch & if-else?

Multithreading Interview Questions

1. What is a thread?
2. What is multithreading; explain when should we develop multithreading?
3. What are the advantages of multithreading?
4. Give different real world applications of multithreading?
5. What is the difference between sequential, parallel and concurrent flow of execution?
6. What is the difference between multitasking and multithreading?
7. How many threads are created by default in JVM?
8. Is Java by default multithreaded programming language?
9. How can we execute multiple tasks concurrently?
10. What are the different ways we have to create custom thread?
11. What are the steps we must follow in creating custom thread extending from Thread class and implementing from Runnable interface?

12. Why custom thread implementing from Runnable interface approach is recommended?
13. What is the use of start() method & run() method?
14. When we call *start* method, will thread execution started immediately?
15. How can we execute logic in custom thread & main thread?
16. What will happen if we do not override run() method?
17. What is the mandatory way to override run() method in the below cases?
    1) When extends from Thread class?
    2) When implementing from Runnable interface?
    3) extends from Thread class & also implementing Runnable interface
18. Can we overload run() method, then will it be executed?
19. If we only overload run() method, in which case we get CE?
    1) Subclass is derived from Thread class?
    2) Subclass is derived from Runnable interface?
    3) Subclass is derived from Thread class & also from Runnable interface?
20. Can we override start() method in Thread subclass?
    1) If we override start() method, will custom thread created?
    2) What is the right design to override start() method?
    3) What is situation force us to override start() method?
21. What is the meaning of single thread model and multithread model application?
22. Can we call run method directly, if so what will happen?
23. What is the difference between th.start() & th.run()?
24. Can we call run and start method both at a time, if so what will happened?
25. Is it possible to start a thread more than once?
26. How can we create multiple custom threads?
27. Explain lifecycle phases of a thread?
28. What is the meaning of active thread & live thread?
29. How can we find how many active threads exist in JVM?
30. How can we find a given thread is live thread or not?
31. If we call yield() method in which state thread will enter?

32. What are the algorithms used in Threads execution process?
    1) Will a newly born thread enter into running state directly?
    2) From block state will thread directly enter into running state?
33. What is thread priority?
    1) What is the default priority of every new thread?
    2) What is the priority of main thread?
    3) Why every thread default priority is 5?
    4) What is the range of thread priority?
    5) How can we get & set thread's priority?
    6) What will happen when we set thread priority to 15?
    7) If two threads have different priority which thread will execute first?
    8) If two threads have same priority which thread will execute first?
    9) When a new thread is born, how can we start its execution immediately?
34. Explain thread class constructors Thread() & Thread(String)?
35. What is the default name of the thread?
    1) How can we get thread name?
    2) What are the different ways we can change the default name of the thread?
36. Can we change name & priority of thread after it is started?
37. What is the meaning of currently running thread & current thread object?
    1) How can we get currently running thread reference?
    2) In the below cases which thread object reference is return
        i. Thread.currentThread() is placed in main method
        ii. Thread.currentThread() is placed in run() method
        iii. Thread.currentThread() is placed in m1() method
    3) If we place th.run() method call in main method, what is the currently running thread object & current thread object of run() method?
    4) What is the output printed for the below statements in run() method when we call run() method from main method
        i. Thread.currentThread().getName() Æ
        ii. this.getName(); Æ

    5) Can we change main thread name & priority, write code for it?
    6) Write a program to prove static block is executed in main thread?
38. What is the meaning of thread group?
    1) What is the default group?
    2) How can we get thread group name?
    3) How can we create our own group for custom threads?
39. What is printed when a thread class object is printed?
40. How many types of threads we can create in JVM?
    1) What is a Non-daemon daemon thread?
    2) What is a Daemon thread?
    3) JVM wait for which thread to complete?
    4) Why main thread is a non-daemon thread?
    5) Why gc is a daemon thread?
    6) Why every custom thread is created as non-daemon thread?
    7) How can we create a custom thread as daemon thread?
    8) What is the rule in calling setDaemon() method?
    9) Is it possible to change the demon property after starting thread?
    10) If we create custom thread as daemon can we guarantee its full execution?
    11) How can we find the given thread is a daemon or not?
41. What are the different ways to pause thread execution?
    1) How can we pause thread execution to allow other thread to execute?
    2) What is the difference between yield() & sleep() methods?
    3) How can we pause a thread execution until some other thread execution is completed?
    4) What is the difference between join() & join(5) & sleep(5)?
    5) Why suspend() & resume() methods are deprecated?

42. Explain the use of below Thread class methods?
    1) start()
    2) run()
    3) isAlive()
    4) activeCount()
    5) setName(String)
    6) getName()
    7) setPriority(int)
    8) getPriority()
    9) currentThread()
    10) toString()
    11) setDaemon(true)
    12) isDaemon()
    13) yield()
    14) sleep(int)
    15) join()
    16) join(int)
    17) interrupt()
    18) interrupted()
    19) isInterrupted()
43. What is synchronization?
    1) When should we develop synchronization?
    2) Explain different applications area of synchronization?
    3) How can we develop synchronization in Java?
    4) Synchronized keyword either applied to method or object?
    5) What JVM will do when we call synchronized method?
    6) What is meaning of object lock & class lock?
    7) When a thread is called as monitor?
    8) When will the object is unlocked?
    9) How JVM will mark an object is locked & unlocked?
    10) When an object is locked by a thread,
        i. Can we use it to call synchronized method from other thread?
        ii. Can we use it to call non-synchronized method from other thread?
    11) How can we lock argument object of method?
    12) What are the differences between synchronized method & synchronized block?
    13) What are the advantages of synchronized block over synchronized method?
    14) What is the difference in declaring static & non-static method as synchronized?
    15) Can we define multiple synchronized blocks in a method?

    16) When should we develop multiple synchronized blocks instead of declaring complete method as synchronized?
    17) If threads use different objects, will synchronization applied?
    18) What is meaning of thread-safe & non-thread safe objects?
    19) How can we develop thread-safe objects?
    20) Create BankAccount object to solve data corruption problem?
    21) Create Train object in multithreading environment to issue thickets?
    22) What is the difference between volatile & synchronized keywords?
44. What is a deadlock?
    1) When will we get deadlock situation?
    2) What is the keyword causes deadlock?
    3) What is the difference between deadlock & starvation?
    4) Given me different code snippets to get deadlock?
    5) How can we overcome from deadlock situation?
45. What is Inter thread communication?
    1) How two thread can communicate?
    2) In addition to synchronized method what else we should use?
    3) What is the use of wait(), notify() & notifyAll() methods?
    4) What is the code template to develop inter thread communication?
    5) When should we develop two threads with communication?
    6) Develop code for Producer-Consumer application?
46. Why wait, notify & notifyAll are defined in Object class, but not in Thread class?
47. What are the various rules we must follow in calling wait(), notify() & notifyAll()?
    1) What is the exception we must handle in calling wait() method?
    2) Can we call these three methods in a non-synchronized method?
    3) If we call these three methods in a non-synchronized method, Why it leads to exception *IllegalMonitorStateException*?
    4) Even if notify()/notifyAll() methods are not called how can we bring out the thread from waiting state?

5) If we call wait(100), will thread starts its execution immediately after 100 milliseconds?
6) What is the difference between wait() & wait(100)?
7) What is the difference between sleep(100), join(100) & wait(100)?

48. What is inline thread?
49. What is the use of ThreadGroup class?
50. How can we add our threads to our custom thread group?
51. What is the use of ThreadLocal class?

File IO Interview Questions

1. What are different places we have to store data permanently?
2. How can we store data permanently in a file?
3. What is the use of java.io package?
4. To access this package class should we place import statement?
5. What is a Stream, How many types of Streams java supports?
6. What is the difference between binary stream & character streams?
7. In which version character streams are introduced, & why?
8. What is the difference between inputstream & outputstream?
9. What are the super classes to all types of inputstream & outputstream classes?
10. What are the super classes to all types of reader & writer classes?
11. List out the important subclasses & methods of InputStream class?
12. List out the important subclasses & methods of OutputStream class?
13. List out the important subclasses & methods of Reader class?
14. List out the important subclasses & methods of Writer class?
15. What is the difference between in.read() & in.read(byte[]) methods?
16. What is the difference between out.write() & out.write(byte[]) methods?
17. How many bytes of data we can write & read at a time from a file?
18. What is the use of FileInputStream & FileOutputStream classes?
19. What is the use of FileReader & FileWriter classes?
20. When will get FileNotFoundException?
21. When will get IOException?

22. What is the value fis.read() method returns when control reached end of file?
23. Write a program to copy one File data to another file?
24. Writer a program to copy Multiple Files data to a single file?
25. What is the use of SequenceInputStream class?
26. What is the use of BufferedReader class?
27. What is the difference between FileReader & BufferedReader classes?
28. How can we read one line at a time from a file?
29. What is the value br.readLine() method returns when control reached end of the file?
30. What is the difference between PrintStream class from other stream class?
31. How can we write the given data as it is in a file?
32. Is PrintStream an InputStream class or OutputStream class?
33. Why PrintStream class name is not PrintOutputStream?
34. Explain System.out:println() statement?
35. Is println() method a overloaded method?
36. What is the difference between print(), println(), printf() methods?
37. When we call println(char[]) parameter method by passing null, why will we get NullPointerException?
38. How many streams are by default opened in JVM, what are they?
39. How can we access these three internal streams?
40. What is the difference between System.in, System.out, System.err?
41. What should we use System.out & System.err?
42. What is the internal method used in printStackTraceMethod() to display exception?
43. How can we redirect STDOUT data to a file?
44. What is the use of System.setIn(), System.setOut(), System.setErr() methods?
45. What is the problem in using System.in.read() statement to read data from keyboard?
46. What is the right procedure to read data from keyboard?
47. What is the use of InputStreamReader class?
48. Can you explain below statement

```
BufferedReader br = new BufferedReader(new
InputStreamRader(System.in));
```

49. What are the different ways we have to read data from keyboard?
50. When will we use wrapper classes in reading data from keyboard?
51. What is the use of Console class, can we create its object use its constructor?
52. How can we read password from console by hiding its characters?
53. Why console.readPassword() method returns password as char[], but not as String?
54. What is the meaning of filtering?
55. What is the use of DataInputStream & DataOutputStream classes?
56. fis.write() can only write one byte data only in int range, then how can we writ other primitive values?
57. What is the difference between fos.write() & dos.writeInt() methods?
58. What is the difference between fis.read() & dis.readInt() methods?
59. What is rule in calling writeXXX() & readXXX() methods?
60. When will get EOFException?

Serialization Interview Questions

1. What is Serialization and deserialization?
2. How can we save & restore object data from file?
3. What is the use of ObjectOutputStream & ObjectInputStream classes?
4. What are the rules we must follow in serialization and in deserialization?
5. When will we get NotSerializableException?
6. What is the use of java.io.Serializable interface?
    1) Why Serializable interface does not have methods?
    2) Why Serializable interface is called marker interface?
7. Explain all steps involved in object serialization?
8. Explain all steps involved in object deserialization?
9. Why file extension in which we are saving object data should be ".ser" can it be ".txt"?
10. What is the information stored in .ser file?
11. When will ois.readObjed() method throws ClassNotFoundException?
12. When will ois.readObject() method throws InvalidClassException?

13. What is the use of serialVersionUID variable?
    1) What is serialVersionUID, who will create it in .class file?
    2) On what basis compiler generates serialVersionUID value?
    3) If we add or remove class members will serialVersionUID value also changed?
    4) What is the use of serialver command, where is it available?
    5) For what type of class only serialVersionUID is generated?
    6) Can we create serialVersionUID variable in a class explicitly?
    7) What should we do to complete deserialization successfully even if the class is modified after serialization?
    8) What is the complete syntax of serialVersionUID?
    9) Is it possible to remove private, static, final keywords in creating serialVersionUID variable?
14. Assume the variable serialVersionUID is created in the class by programmer, serialization is completed, then will deserialization is complete in the blow cases?
    1) New variable is added in this object class?
    2) Serialized variable is removed from this object class?
    3) Serialized variable's data type is changed?
    4) class is derived from a class
    5) New method is added
15. What is a transient keyword & what is the use of it?
    1) How can we stop saving object data?
    2) To what variables transient keyword is allowed?
    3) Is there any use of declaring static variable as transient?
16. What are the variables not participate in Serialization and deserialization?
17. What is the value stored in transient variable & static variable in deserialization?
18. What is the correct procedure to store & read confidential data of an object from file?
19. Let say object has 100 variables, among them how can we save only 10 variables, & among the saved 10 variables how can we read only 5 variables?
20. The meaning of customizing serialization?

1) What is the purpose of private writeObject() & private readObject() methods?
2) What is the purpose of Externalizable interface?
3) What are the differences between Serializable and Externalizable interfaces?
4) Is Externalizable interface a marker interface?
5) Explain the use of Externalizable interface methods?
6) Who will execute private writeObject()& writeExternal() methods, at what time?
7) Who will execute private readObject() & readExternal() methods, at what time?

21. Explain various steps involved in serializing IS-A relation object?
    1) If super class is Serializable, will the complete object graph is saved & restored?
    2) If super class is not Serializable, will the complete object graph is saved & restored? What is the rule we must follow if super class is not Serializable type?
    3) Why root super class should not be subclass of Serializable interface?
    4) Why Collection interface is not subclass of Serializable & Cloneable?
    5) If super class is a Serializable type how can we stop serializing subclass object?
22. Explain various steps involved in serializing HAS-A relation object?
    1) To serialize an object, is it mandatory that its internal object also should be Serializable type?
    2) How can we save an object if its internal object is not Serializable type?
23. How can we save & restore multiple objects from a file?
24. What are the different ways to create an object?

File Class Interview Questions

1. What is the use of File class?
2. What are the various methods available in file class?
3. How can we create new files and directories from Java program?
4. What is the difference between file.mkdir() & file.mkdirs() methods?

5. What is the use of file.exists() method?
6. How can we find the given file is a normal file or directory?
7. How can we find absolute path of a given file?
8. How can we list all files and subdirectories of a given folder?
9. While listing the files how can we filter them?
10. How can we delete files and directories?
11. What is the rule in deleting a directory?
12. How can we retrieve OS dependent directory separator?

Collections Framework interview questions

1. What is a collection, Collection API & Collection Framework?
2. In what situation we will use collection object?
3. What are the different operations we can perform using collection?
4. What are the THREE problems of array object for creating collection API?
5. What is the inbuilt object used in all collections to store heterogeneous objects?
6. When we should use array object and collection object, which will give us high performance in storing objects?
7. In which package collection API is given?
8. Why java.util package classes are called Collection API classes?
9. What is a container object why collection objects are called container objects?
10. What is the difference between container objects, cursor objects, utility objects?
11. From which version onwards collection API has been called as Collection Framework?
12. What is the meaning of legacy?
13. What are the legacy collection classes & interfaces available?
14. Why the word legacy is only used with collection objects?
15. What are the primary advantages of Collection API?
16. In how many formats we can store objects in collection?

17. What are the two collections given by SUN in java 1.0 to store objects in array format & in (key, value) pair format?
18. What is the difference between Vector & Hashtable?
19. What are the 4 problems of Vector, & its 4 alternative classes given java 1.2?
20. What are the 2 problems of Hashtable, its 2 alternative classes given in java 1.2?
21. What are the TWO root interfaces of in Collection Framework?
22. What is the difference between Collection and Map?
23. Why Map is not created as sub interface of Collection?
24. Why Collection doesn't extend Cloneable and Serializable interfaces?
25. What are the important subclasses of Collection & Map interfaces?
26. What are the new collection classes added in java 1.2, 1.4, 1.5, 1.6, 1.7, & 1.8?
27. How many types of collections java supports?
28. What is the data structure used in List type collections implementation?
29. What is the data structure used in Set and Map collections implementation?
30. What is the difference between the three words collection, Collection & Collections?
31. Among Set, List, Map type collections
    1) Which collection is unordered collection?
    2) Which collection is indexed ordered collection?
    3) Which collection is sorted collection?
    4) Which collection is unique collection?
    5) Which collection is duplicate collection?
    6) Which collection is homogeneous collection?
    7) Which collection is heterogeneous collection?

32. What are the 9 core interfaces of collection framework?
    1) What is the use of Collection interface, does it have any direct implementations?
    2) What is the use of Set interface, what are the implementations of Set interface?
    3) What is the use of List interface, what are the implementations of List interface?
    4) What is the use of Queue interface, what are the implementations of Queue?
    5) What is the use of Map interface, what are the implementations of Map interface?
    6) What is the use of SortedSet interface, what are the implementations of SortedSet?
    7) What is the use of SortedMap, what are the implementations of SortedMap?
    8) What is the use of NavigableSet interface, what is its implementation subclass?
    9) What is the use of NavigableMap interface, what is its implementation subclass?
33. What are the 15 important implemented classes of Collection & Map interfaces?
    1) When should we use ArrayList, LinkedList, Vector, and Stack?
    2) When should we use PriorityQueue?
    3) When should we use HashSet, LinkedHashSet, and TreeSet?
    4) When should we use HashMap, LinkedHashMap, TreeMap, and Hashtable?
    5) When should we use IdentityHashMap, WeakHashMap?
    6) When should we use Properties class?
34. What is the difference between Collection & Map?
35. What is the difference between Set, List, Queue and Map?

36. What is the difference between Set & SortedSet?
37. What is the difference between Map & SortedMap?
38. What is the difference between SortedSet & NavigableSet?
39. What is the difference between SortedMap & NavigableMap?
40. What is the difference between array & ArrayList?
41. What is the difference between ArrayList & Vector?
42. What is the difference between ArrayList & LinkedList?
43. What is the difference between HashSet & LinkedHashSet?
44. What is the difference between HashMap & LinkedHashMap?
45. What is the difference between HashSet & TreeSet?
46. What is the difference between TreeSet & TreeMap?
47. What is the difference between HashMap & IdentityHashMap?
48. What is the difference between HashMap & WeakHashMap?
49. What is the collection object that stores only unique objects in no order?
50. What is the collection object that stores only unique objects in insertion order?
51. What is the collection object that stores only unique objects in sorting order?
52. Which collection we must use to store duplicate objects?
53. Which collection we must use to store duplicate objects in multithreading environment?
54. How can we remove duplicate objects from ArrayList?
55. How can we find which object is repeated & how many times?
56. How can we store only unique objects in indexed order?
57. How can we get synchronous version of ArrayList object?
58. What are the activities done when we insert or remove an element in ArrayList?
59. What is the collection object we must use if more operations are inserting or removing objects at middle of the collection?
60. Why insertion and deletion operations are fast in LinkedList compared to ArrayList?

61. What is a cursor object?
    1) What is the difference between cursor object & container object?
    2) What are the three types of cursor objects exists in collection API?
    3) What is the difference between Enumeration & Iterator?
    4) What is the difference between Iterator & ListIterator?
    5) Why ListIterator can be a bidirectional cursor?
    6) Why Iterator does not have add() method?
    7) What are the various rules we must follow in using Iterator?
    8) Why we cannot call itr.remove() method before itr.next() method call?
    9) When will you get
        - NoSuchElementException
        - IllegalStateException &
        - ConcurrentModificationException?
    10) What is the difference between fail-fast & fail safe?
    11) How can we get Enumeration object on collection framework classes?
    12) How can make sure when we return collection from a method it is allowed only from reading operations?
    13) How can retrieve elements from Map using Iterator?
    14) What are the three views of Map object?
    15) What is Map.Entry?
    16) What is the difference between sequential access & random access?
    17) Is it possible to retrieve objects from collection randomly using cursor objects?
    18) How can List type collection provide random access?
    19) What is the difference between list.get(index) & itr.next()?
    20) What is the difference between litr.add() & list.add(index, obj) & list.add(obj)?

62. What are the different ways to retrieve objects from Set/List/Queue/Map collections?
63. What are the benefits added to collection programming with
    1) auto boxing and unboxing?
    2) Enhanced for-loop
    3) Generics
64. What is the difference between list.remove(5) & set.remove(5)?
65. **What are the operations performed by** addAll(), putAll(), containsAll(), removeAll(), retainAll() methods?
    1) What is the difference between add() & addAll()?
    2) What is the difference between put & putAll()?
    3) What is the difference between contains() & containsAll()?
    4) What is the difference between remove() & removeAll()?
    5) What is the difference between removeAll() & retainAll()?
    6) What is the difference between removeAll() & clear()?
66. Which method we must use to create collection of collections?
67. Which method we must use to create collection of maps?
68. What is the difference between capacity & size?
69. What is the default capacity of ArrayList, Vector, HashSet, HashMap, Hashtable?
70. When collection is filled, what will happen if we add new object?
71. How must vector capacity is incremented?
72. How much ArrayList capacity is incremented?
73. What is the meaning of load factor?
74. What is the load factor of set & map objects?
75. Which collection object does not allow null?
76. Which collection object allows only one null?
77. Which collection allows multiple nulls?

78. How many nulls we can store as (key & as value) in HashMap & TreeMap?
79. Is it possible to store null in Hashtable either as key or value?
80. Is it possible to store heterogeneous objects in HashSet & HashMap?
81. Is it possible to store heterogeneous objects in TreeSet & TreeMap?
82. Is it possible to store duplicate objects in HashSet & HashMap?
83. Explain how Set & Map collections will stop storing duplicate objects?
    1) What will be used in storing objects in Set & Map collections?
    2) Why Set & Map collection internally uses hashCode() & equals() methods?
    3) Will all subclasses of Set & Map use hashCode() & equals() methods?
    4) For finding duplicate object "==" operator & equals() methods are sufficient, then for what purpose hashCode() method is used?
    5) What is a bucket, what is the use of it, when a new bucket is created?
    6) How one group of objects are differentiated from another group of objects?
    7) Why we must override hashCode() & equals() methods in our class for storing its objects in Hash based collection classes such as HashSet, LinkedHashSet, HashMap, LinkedHashMap, Hashtable?
    8) What will be the problem if we do not override hashCode() & equals() methods in subclass?
    9) Why immutable object recommended to use as key in Map?
84. Explain how TreeSet can store objects in sorting order?
    1) What is the default sorting order of TreeSet?
    2) How can we passing sorting order of our objects to TreeSet class?

3) What are the differences between Comparable & Comparator interfaces?
4) What is the meaning of natural sorting order & custom sorting order?
5) What is the meaning of custom comparator?
6) What do you mean by non-comparable objects?
7) When will TreeSet use Comparable interface for sorting elements?
8) When will Tree Set use Comparator interface for sorting elements?
9) What are the steps we must follow to develop a class to add its objects to TreeSet or to TreeMap as key?
10) What types of objects are only allowed to store in TreeSet & TreeMap?
11) How can we store non-Comparable objects in TreeSet?
12) How can store a class objects in different sorting orders in TreeSet objects?
13) Why compareTo() method has one parameter & compare() method has two parameters?
14) compareTo() & equals() methods are meant from comparing objects of a class, then why compareTo() method return type is int & equals() method return type is boolean?

85. What are the three methods we must override in subclass to store its objects in all types of collection objects?
86. What is the use of **Collections** class?
    1) How can we find number of occurrences of a given object?
    2) How can we sort elements in ArrayList?
    3) How can we get synchronous version of ArrayList?
    4) How can we get immutable collection object?
    5) While passing a Collection as argument to a function, how can we make sure the function will not be able to modify it?

87. What is the use of Arrays class?
    1) How can sort objects in array?
    2) How can we search an object in array?
    3) How can you print all elements of array without using for loop explicitly?
88. What is the use of stack?
    1) Are both LIFO & FILO same?
    2) How should we add objects to Stack?
    3) What is the difference between pop() and peek() methods?
    4) What is the difference between insertion index and searching index in stack?
    5) Stack is a subclass of which collection?
    6) Can we use List interface methods on Stack object for adding, retrieving & removing its objects?
89. What is the use of Properties class?
90. What is the difference between Properties class & ResourceBundle class?
91. What is the use of StringTokenizer class?
92. What is the use of RandomAccess interface, is it a marker interface?
93. Explain in a project, when will we create collection of objects, collection of Collections & collection of Maps created?
94. Explain the meaning of collection can be used as *Data transferable object* & as *mini DB*?
95. What are the new features added in Java 8 for collections?

Enum Interview Questions

1. What is enum, what is the use of enum?
2. How can we create enum in Java?
3. What is the rule based on enum keyword?
4. What are the members we can place inside enum?
5. What is the importance of ; in enum?
6. When ";" is mandatory in enum?

7. What are the code changes done by compiler in enum .class file?
8. What is a named constant, what is the type of named constant in enum?
9. What is printed if we print named constant?
10. Will enum have constructor?
11. What is the default accessibility modifier of enum constructor?
12. What is the difference between default constructor in class and enum?
13. Can we create object of enum like normal class?
14. Can you draw enum object memory structure?
15. Can we create variables, methods, blocks and constructors in enum?
16. What is the syntax to access static & non-static members of an enum?
17. What is the superclass of enum?
18. Can we create enum from another enum?
19. What is the separator of named constants, is it "," or ";"?
20. Is it possible to assign values to named constant using "=" operator?
21. What is the procedure to assign values to named constants?
22. What are the rules we must satisfy to assign value to enum?
23. Can we create inner enum inside a class?
24. Can we create enum inside a method?