Java Interview Questions

Java basics:

1. Explain How java becomes a platform independent language?
2. What is the difference between a Procedure Oriented Language and an Object-Oriented language?
3. What is Byte Code in java Programming language?
4. Explain the **difference between JVM, JRE, JDK and JIT Compiler** in java.
5. What is the difference between keywords and identifiers in java?
6. What are the different data types in java?
7. Explain the **difference between primitive and non-primitive data types** in java.
8. What is primitive type casting in java? Explain widening and narrowing in java.

9. What is a variable? What are the different types of variables in java?
10. Can a local variable have default value in java?
11. What is the use of break and continue keyword in java?
12. Why the main method of java should be always public?
13. Why the **main method of java should be always static?**
14. Why the return type of main method in java is always void?
15. How can we call a static method anywhere?
16. How can we call a non-static method anywhere?
17. What is the meaning of formal argument and actual argument of method?
18. What is **Compile Time Binding in java?**

Oops Concept:

Basic oops:

1. What is **class**?
2. What is **Object or instance** of class?
3. **Is java a 100% Object Oriented programming Language?**
4. What are the four pillars of an Object-Oriented Programming language?
5. What is “**new”** in java? Why it is used?
6. What is a **reference variable** in java? What is the default value for a reference variable in java?
7. What is the difference between local variable and instance variable?
8. Explain the important difference between **stack area and heap area.**
9. Explain **static** variable of java?

10. Explain **non-static or instance** variable in java.

11. Give example when to use static and when to use non-static variable.

12. What is the difference between a static variable and a non-static variable?

13. What is **variable Hiding** in java? How variable hiding is resolved in java?

14. Explain **this keyword** in java?

Constructor:

15. What is **constructor** in java?
16. What are the **differences between Method and Constructor**?
17. What are the different types of constructors?
18. What is default constructor in java? How it is created and what is its use?
19. Does every class in java has a constructor?
20. Explain **constructor Overloading** in java? What are the different ways to achieve it?
21. Explain this() statement in java and why it is required?
22. What is constructor chaining in java and how to achieve it in java?
23. Explain copy constructor.
24. Can a constructor be static in java?
25. What does a constructor return in java?

Method Overloading:

26. What is **method Overloading** in java? How to achieve it and what is its advantage?

27. Give some examples of method overloading in java.

28. **Can we overload main method in java?**

29. Can we overload both static as well as non-static methods?

30. **Why method overloading is also called compile time polymorphism or static polymorphism?**

31. What is method chaining in java and how to achieve it in java?

32. **Design a program to explain method chaining in java.**

Association:

33. What is Association/Relationship in java?

34. What are the different types of relationship in java?

35. What is **HAS-A and IS-A relationship** in java?

36. What is composition and aggregation in java?

37. **When an object is deleted or garbage collected from heap area** in java?

38. **What is the meaning of automatic garbage collection in java?**

39. Explain **System.out.println(**) statement in java?

40. **What is call by value and call by reference?**

41. **Does java support call by reference? Why?**

42. How many references can one object have in java?

Inheritance:

43. What is inheritance and how to achieve it in java?

44. What are the different types of inheritance in java? Explain them.

45. What is base class or super class?

46. What is derived class or sub class?

47. What is **super keyword** in java?

48. How many classes can be extended by one class?

49. **Why java doesn't support multiple inheritance?**

50. **What is diamond shape problem? How to resolve it in java?**

51. Explain **super()** statement in java.

52. What is the **difference between this and super keyword** in java?

53. Explain the **difference between this() and super()** statement.

54. **Why this() and super() statements can't be used in a constructor together**?

55. **Which type of members of parent class are not inherited by child class in java?**

56. **Does a constructor of parent class is inherited by child class? Why?**

57. What is non primitive type casting in java?

58. Explain Upcasting/Generalization and Downcasting/Specialization in java.

59. **What is the use of upcasting and downcasting?**

60. **Why we need to perform Upcasting?**

61. **Why we need to perform Downcasting?**

62. What is implicit and explicit type casting?

Method Overriding:

63. **What is method Overriding in java?**
64. Can we Override a static method in java?
65. When to go for method Overriding? Give some examples.
66. What is the benefit of method Overriding?
67. What are the requirements for method Overriding?
68. **Can we override the main method of java? Why?**
69. **Which method of parent class can't be Overridden by child class?**
70. Can we override constructor of a class in java?

Polymorphism:

71. **What is polymorphism?**
72. **What are the different types of polymorphism in java?**
73. What are the different ways to achieve Compile Time Polymorphism?

74. What are the different ways to achieve Run Time Polymorphism?

75. **Explain the difference between Compile Time Polymorphism/Static Polymorphism and Run Time Polymorphism/Dynamic Polymorphism.**

76. Explain Static Method Dispatch and Dynamic Method Dispatch.

77. **Why operator overloading is not allowed in java?**

78. Explain method shadowing and variable shadowing.

Encapsulation:

79. **Define encapsulation? How encapsulation is achieved in java?**

80. **What are getters() and setters() method which are used in encapsulation.**

81. **What is data hiding in java? Why it is used?**

Abstraction:

82. **What is abstraction in java?**

83. **What are the different ways to achieve abstraction?**

84. What is an abstract method?

85. Can a static method be abstract?

86. Can we make an abstract method private?

87. Can we create object of an abstract class?

88. **Are constructors allowed inside an abstract class? Why?**

89. Is it mandatory to have abstract method inside an abstract class?

Final Keyword:

90. What is the **purpose of final keyword** in java?

91. Can we make an instance variable final? What is its use and how it should be initialized?

92. **Can we override a final method?**

93. **Can we declare an abstract method as final?**

94. Can a class be declared final? Can a final class have a child class?

95. When we need to make a class as final?

Interface:

96. **What is an interface in java? How can we create an interface?**
97. **For what purpose interface is used?**
98. **Explain all the different types of members which are allowed inside an interface.**
99. **Can we have instance variables inside an interface?**
100. **Can we create Object of an interface?**
101. **Are constructors allowed inside an interface? Why?**
102. What are the different types of members which were allowed inside interface in java8 version?
103. Which type member is allowed inside interface in java9 version?
104. Explain the default method of interface.
105. **How multiple inheritance is supported by interface? Explain.**
106. Can an interface define non-static variables?

107. How many interfaces can be implemented by one class?

108. Can an interface extend a class?

109. Can we make an interface as final?

110. Explain Functional and Marker interface.

111. **What is the difference between a class and an abstract class?**

112. **What is the difference between a class and an interface?**

113. **What is the difference between an abstract class and an interface?**

114. **Give some real time examples of abstraction/interface.**

115. **Explain briefly all the four pillars of java.**

Package:

116. **What is package? What are the benefits of creating package?**

117. Name some important packages in java?

118. **How user defined packages are created in java?**

Access Modifier:

119. What are access modifiers? Why access modifiers are used?

120. **Explain public, protected, default and private access modifiers of java.**

121. **What is the scope of public access modifier?**

122. **What is the scope of protected access modifier?**

123. **What is the scope of default access modifier?**

124. **What is the scope of private access modifier?**

125. Which access modifiers are not allowed with class or interface?

126. **Can a constructor be private? What is its use?**

127. What is the access of default constructor?

128. **Can we decrease the visibility of method in child class while overriding a method of parent class?**

129. Can we make an abstract method as private?

130. **Can we override a method if it is private in parent class?**
131. What are non-access modifiers in java?
132. Mention some non-access modifiers in java with its use.

Other:

133. Can we use static and abstract keyword together?
134. What is the difference between static block and non-static block?
135. **What is implicit and explicit constructor call in java?**
136. Describe Outer and Inner class.
137. Explain about static class.
138. What is the use of instanceof keyword?
139. What is the purpose of native keyword?
140. What is the purpose of strictfp keyword?

Object Class:

141. Describe Object class.
142. Describe toString(), hashCode() and equals() method of Object class.

143. **Why we need to Override toString() method of java?**

144. What is hashCode() method in java?

145. **Why we need to Override equals() method of Object class?**

146. What is the purpose of finalize () method of Object class?

Other Important Questions:

147. What is Wrapper class?

148. Explain Boxing and Unboxing in java.

149. **What is the meaning of String[] args of main method? Why it is used?**

150. Describe Command Line Arguments in java.

151. What is varargs method? Explain how it is used?

152. Explain static import statement in java.

153. **What is a java Bean class? What is its use?**

154. **What is immutable class? What is its use and how to create it?**

155. **What is singleton class? How to create it and what is its use?**

156. **Mention important features of java8 version.**

157. **Explain lambda expression introduced in java8 update.**

String:

1. What is String in java?
2. **Why String class is final?**
3. **What is the meaning of Immutable in terms of String?**
4. **Why String Object is immutable in java and what is its advantage?**
5. What are the different ways to create a String Object?
6. **What is String Constant Pool(SCP) in java?**
7. What is the difference between two String Objects provided below:
   String s1=”abcde”;
   String s2=new String(“abcde”);
8. What is StringBuilder and StringBuffer in java?

9. **Why SCP in not eligible for automatic garbage collection in java?**
10. How many interfaces are implemented by String class?
11. **What is the difference between = = (double equal operator) and equals() method in Java?**
12. What is a mutable string in Java?
13. Why StringBuffer objects in Java are mutable?
14. Why do we need StringBuilder in Java?
15. Which class is more preferable: StringBuffer or StringBuilder? Why?
16. When should you use StringBuilder?
17. **Mention the difference between String, StringBuilder and StringBuffer class.**

Exception Handling:

1. What is exception in java?
2. What is exception handling in java?
3. **Explain the types of exception in java?**
4. **Explain the difference between checked exception and unchecked exception.**

5. Give some examples of checked and unchecked exception.
6. Describe exception hierarchy.
7. Which is the super most class of all exceptions and errors?
8. Which is the super most class of all exceptions?
9. Why an exception occurs in the program?
10. **What is the difference between exception and error?**
11. What is the advantage of using exception handling in Java?
12. What is NullPointerException? When it occurs?
13. What is ClassCastException? When it occurs?
14. What happens when an exception occurs in the program?
15. **What are the different ways to handle exception in java?**
16. What does JVM do when an exception occurs in a program?
17. **Explain try catch block in java?**

18. How many exceptions can be thrown from a try block at a time?

19. How many catch blocks can be used with one try block?

20. How many try catch blocks can be used inside a program?

21. What is nested try catch block?

22. Explain finally block? What is the main purpose of using finally block?

23. Can we use try block without catch block?

24. Explain the use of throw keyword.

25. What is the purpose of throws keyword in java?

26. Explain the difference between throw and throws keyword.

27. What is the use of printStackTrace() method in exception handling?

28. Mention some scenarios when finally block will not execute.

29. What is custom exception? How to create and use custom exception in java?

30. What will happen to exception object after exception handling is done?

31. Define unreachable catch block error.

Collection Framework:

1. What is Collection Framework?
2. What are the different hierarchy of collection framework?
3. What are the differences between collection and array?
4. What is List? Explain its features.
5. **Explain the features of ArrayList?**
6. **For which requirements ArrayList is best recommended to use?**
7. **What is the difference between Array and ArrayList?**
8. How ArrayList grows?
9. What is the internal implementation of data structure used by ArrayList?
10. **Explain LinkedList and its features. What is the internal implementation of data structure used by LinkedList?**
11. **For which requirements LinkedList is best recommended to use?**

12. What are the different interfaces implemented by LinkedList?

13. What are the differences between ArrayList and LinkedList?

14. Explain Vector and its features.

15. What are the differences between ArrayList and Vector?

16. Explain Stack and its features with some of its methods.

17. Explain set and its features.

18. What are the differences between List and Set?

19. Explain HashSet implementations in details along with its features.

20. What is Hashing?

21. Explain LinkedHashSet implementations in details along with its features.

22. Explain TreeSet implementations in details along with its features.

23. Describe the differences between HashSet, LinkedHashSet and TreeSet.

24. What is Queue? Explain its features.

25. Explain PriorityQueue implementations in detail.

26. What is Iterator? How to use it?

27. What is ListIterator? How to use it?

28. What is the use of Collections class?

29. How Collections class helps to achieve sorting for Objects?

30. What is Comparable interface and how it is used to achieve Sorting for custom objects?

31. What is Comparator interface and how it is used to achieve Sorting for custom objects?

32. Describe the differences between Comparable and Comparator interface.

33. What is Map? How Objects are stored in Map?

34. What is HashMap and explain its implementations along with features.

35. What is LinkedHashMap and explain its implementations along with features.

36. What is TreeMap and explain its implementations along with features.

37. How to run loop on Map?

Multithreading:

1. What is a Process?
2. What is a Thread?
3. Explain Thread Properties.
4. **What is multithreading?**
5. **What are the advantages of multithreading?**
6. **What are the different ways of creating a thread?**
7. How a thread starts in java?
8. **What are the different stages of a thread lifecycle?**
9. What is thread priority? What are the range of thread priority and what is the default priority?
10. Which is the first thread created by JVM when a program starts its execution?
11. What is the use of Runnable interface?
12. Explain sleep() method of Thread class.
13. Explain join() method of Thread class.
14. What is thread safe environment? How it is achieved?

15. **What is Object lock? When it gets created?**

16. **Explain class lock? When it gets created?**

17. **What is thread deadlock?**

18. **Explain how to overcome thread deadlock?**

19. What is interthread communication? How to achieve it?

20. Explain wait() method.

21. Explain notify() and notifyAll() method.

22. **What is thread Synchronization? How it is achieved?**

23. **What is a daemon thread?**

24. **What are the differences between user thread and daemon thread?**

25. How to pause execution of current thread?

26. What is the use of volatile keyword?

File Handling:

1. **What is serialization and de-serialization process?**

2. **What is the use of transient keyword?**

Other:

1. **What are the differences between final, finally and finalize() in java?**
2. **Explain the use of garbage collector?**
3. Can we call garbage collector explicitly?
4. **Why pointers are not used in Java?**
5. **Explain java8 features.**