

## CORE JAVA BATCH:

- 1) what is jvm, jdk, jre ?
- 2) what is JIT compiler?
- 3) is java interpreted or compiled language ?
- 4) Data types in JAVA.
- 5) Primitive vs. Non-Primitive.
- 6) Difference between double and float.
- 7) What is type casting and types in JAVA.
- 8) what is narrowing and what is widening ?
- 9) what are array in JAVA?
- 10) different types of ARRAYS in java ?
- 11) W.A.P to check if the number is prime?
- 12) W.A.P. to swap two numbers without the third variable.
- 13) What is String in JAVA ?
- 14) What is String constant pool in JAVA ?
- 15) String vs. StringBuffer . vs. StringBuilder in JAVA.

16) Explain the concept of String immutability in JAVA.

17) What are constructors in JAVA?

18) Explain types of constructors in java?

19) what is default constructor?

20) Param vs. Non-param constructors in JAVA?

21) Explain the role of this keyword in java?

22) Relation between this keyword and instance variables in JAVA.

23) what is CI in java?

24) How do you make the compulsion to users to at least provide some values at the time of the object creation?

25) use of encapsulation in data hiding and security.

26) bypassing the privates with the help of setters.

27) why you should not apply the CI to private instance variables.

28) explain difference between this call and this();

29)what is constructor jumping.

30) how do you track the no. of instances of the given class.

31)explain the super call in the context of constructor , method and variable.

32)types of inheritances in java.

33)why multiple inheritance is not supported. With code of diamond/ambiguity problem.

34)compile time vs run time polymorphism.

35)inner mechanism of the compulsion to keep the return type and the signature of the method same in the child class.

36)W.A.P to demonstrate over-riding vs. runtime polymorphism.

37) what are access modifiers in JAVA.

38 ) types and order of access modifiers from less to more secure. Prove it with the codes.

39)Protected vs Default.

40) Prove it you can't change the return type in child overridden method if it's primitive / void /String.

41)what is the relation between access modifiers and Overriding.

42)what is abstraction ?

43)prove that abstract class's object is created but indirectly.

44)why it is a compulsion to override the abstract methods in the child class.

45) create the reference of the abstract class with the help of anonymous class.

46) Difference between Abstract class and Interface.

47) Solve the diamond problem in java with the help of the Interfaces (default methods).

48) what is backward compatibility in java 8?

49)default and static methods in java 8 i/f.

50)over-riding in I/F with static and default methods.

51)explain the method hiding with the help of static methods in java 8.

52)What is functional i/f.

53) how do you invoke a method of i/f with anonymous class.

54) writing of lambda expressions.

55)Object class.

56) ToString() and getClass()

57)hierarchy of exception handling.

58)is it compulsory to write the catch block with try ? explain.

59)can we write consecutive try/ finally blocks ?

60)how can we write multiple catch blocks?

61)will finally always get executed ?

62)why it's compulsory to handle checked(Compile) time exceptions at compile time? But optional to handle unchecked(Runtime) exceptions at compile time?

63) difference between compile and runtime exceptions and their working.

64) when to use the throw keyword ?

65) can you throw the multiple exceptions from the single try block a) with overlapping logic. b) separate logic.

66) Can we throw an Exception from constructor ?

67) what is static block ? and its use ?

68) custom compile time exceptions vs custom runtime exceptions.

69) What is .hashCode() and its significance?

70) what is .intern() method and String interning in java ?

71) how can we override the .hashCode()?

72) String class and .hashCode() method.

73) Throw vs. Throws.

74) pipe operator and its constraint in java ?

75) instanceof operator.

76) Propagation of exception in java to JVM?

77) what are the cases when finally does not execute ?

78) why should not we handle the errors?

79) try with resource.

80) what is .hashCode() and its significance ?

81) default behaviour of the .equals() in java ?

82) why does .equals() behaves differently in String class?

83) what is the contract between .equals() and .hashCode() in java ?

84) define :

- a) reflexivity
- b) consistency
- c) not-nullity
- d) Symmetry
- e) transitivity

contracts of `.equals()` in java with example.

85) why to override `.equals()` and `.hashCode()` together in java ?

86) what are the best practices followed when overriding the `.equals()` and `.hashCode()` in Java?

87) what is the HAS-A relationship in Java ? Can you equate the HAS-A with Dependency injection in java ?

88) what is the difference between reference and Object in Java ?

89) Composition vs Aggregation.



90) loose coupling vs. Tight coupling.