

## **Topic wise Java Questions**

### **Classes & Objects**

1. What is a class write the class template?
2. What is an object?
3. What is byte code file? How many byte code files will be generated after compiling source file
4. What will be the name of byte code file?
5. Command used to compile & run java program
6. Can I declare local variable as static?
7. Does java allows global variable?
8. How many types of variable do java supports?
9. How to access static & non static members explain with an example.
10. When to declare data member as static & non static.

### **Blocks**

1. What is block and how many types of blocks available
2. Difference between block & method
3. When static block will execute?
4. When non-static block will execute?
5. How many times static block will execute?
6. How many times non-static block will execute?
7. Can I initialize static data member inside non-static block?
8. Can I initialize non-static member inside static block?

### **Constructors**

1. What is a constructor? & why is it used for?
2. When constructors will be executed? Explain
3. When to go for argument constructor? Explain with an example
4. WAP to demonstrate constructor overloading & what are its advantages.
5. Does the constructor allow return type? If we give what happens
6. Can the constructor be static?
7. Can the constructor be final?

8. What is the access modifiers allowed for constructor?
9. What happens when we create an object explain series of steps

### **Method overloading**

1. What is method overloading explain with an example?
2. Can we overload static methods? WAP to overload main method.
3. Explain how to pass arguments to main method explain.
4. WAP to overload non-static methods.
5. What are the advantages of method overloading?

### **Inheritance**

1. What is inheritance & what are its advantages
2. Explain types of inheritance.
3. Explain why multiple inheritance is not allowed in java?
4. What is this keyword? Why is it used for? Explain with an example
5. What is super keyword? Why is it used for? Explain with an example
6. Explain constructor calling with an example. What are its advantages?
7. Explain constructor chaining with an example? Why constructor chaining required?
8. Difference between this, super & this (), super ().
9. What is the first statement inside default constructor?
10. What happens if a class is declared as final? Example for final class
11. Explain diamond ring problem.
12. Is Object class a final? Justify
13. Explain what happens when we instantiate subclass.

### **Packages**

1. What is a package? & how to create it?
2. Where to write package creation statement?
3. Can one source file contain many package creation statements?
4. When to use import? Explain with an example

5. Can one source file contain many import statements?
6. In which order package & import keyword should exist in a source file?
7. Explain access modifiers provided by java
8. What is data hiding? & how to achieve it?
9. What is encapsulation?
10. What is jar file? Where is rt.jar present? & what it contains?
11. Does default member gets inherited across the package
12. Can I access public members outside the package if the class is default?
13. Can I declare a class as private? What access modifiers can I provide for a class
14. What is the name of byte code file of inner class

### **abstract method & abstract class**

1. What is abstract method?
2. What is abstract class?
3. Where we provide definition for the abstract methods & what if we failed to provide?
4. When to declare a method abstract?
5. When to go for abstract class?
6. Can abstract class be final? Why?
7. Can i declare static method as abstract? Why?
8. Can i change the access modifiers while providing implementation?

For the abstract methods.

9. Can the abstract class be instantiated? Why
10. Does abstract class allows constructor? If Yes, explain with an

Example it's usage

11. Does abstract class allows static members? If Yes, how to access them?

Write an example.

12. Can i declare private method as abstract? Why?

13. Advantages of abstract class

### **Interface**

1. What is an interface, write the template

2. Can an interface extend from another interface? Write an example

3. Can an interface extend from more than 1 interface? Write an example

4. What is marker interface? Examples.

5. What is functional interface? Examples

6. Does interface allows concrete methods?

7. Difference between abstract class & interface?

8. When to go for interface? & when to go for abstract class?

9. Can one interface have multiple implementation classes? Write an example

10. Advantages of interfaces.

11. Explain the importance of interface type reference var.

12. What is a factory class?

13. What is a factory method?

### **Typecasting**

1. What is type casting? Explain data typecasting with example.

2. What is narrowing? Why narrowing should be done by user explicitly.

3. What is widening? Why compiler will do it implicitly

4. Explain class typecasting with an example.

5. What is up casting? & what is down casting?

2. What is the advantage of class type casting, Explain with an example?

3. Explain **instanceof** operator with an example?

6. What is return type of **instanceof**?

7. What is generalization & specialization explain with an example.

### **Method Overriding**

1. What is overriding? Justify how overriding is runtime polymorphism
2. Difference between overloading & overriding
3. Which methods cannot be overridden?
4. Can we override final methods?
5. Can we override static methods?
6. Can we change the access modifier while overriding method?
7. Can we change the return type while overriding method?
8. Can we override abstract method?
9. Can we override concrete method as abstract? Justify its advantages.
10. What is polymorphism? Explain its types with example
11. When to go for overriding? Explain
12. Can we override constructor? Why?
13. While overriding, why co-variant type changes allowed & why not contra variant?
14. Difference between compile time & run time polymorphism

### **Java Bean Class**

1. What is java bean class? Write an example
2. What are getters & setters? Write the syntax of it.

### **Private Constructor & Singleton Class**

3. What is the default access modifier of constructor?
4. Can the constructor be private? Then, how to create an object. Explain with an example
5. What is singleton class? Write an example
6. Explain how to achieve constructor chaining in singleton class.

## 7. Advantages of singleton class

### **Object class & its methods**

1. What is object class? Where is it present?
2. Is Object class final? Justify
3. Mention the methods Object class
4. Which methods of Object class cannot be overridden
5. Explain the behavior of **toString()**, **hashCode()** & **equals(Object obj)** .
6. WAP to compare whether two mobiles are same or not w.r.t name, price & ram. Program should also display states of an object
7. WAP to compare whether two cars are same or not w.r.t name, price & color. Program should also display states of an object
8. WAP to compare whether two watches are same or not. Program should also display states of an object
9. What is the difference between class & Class?
10. How to identify fully qualified name of the class when we have the object.

### **String class & its methods**

1. What is String class? Where is it present?
2. What is the relationship between String & Object class?
3. What methods of Object class overridden in String class & how?
4. Explain constructors of String class.
5. How to convert array of characters to String?
6. What is immutable? & why String is immutable?

7. Justify how String is immutable.
8. What are the advantages & disadvantages of immutable
9. What is difference between **String**, **StringBuilder** & **StringBuffer**?
10. Explain when the objects will be created in SCP & heap memory?
11. Which methods Object class overridden in StringBuffer & StringBuilder?
12. What is the difference between **final** & **immutable**?
13. Why String is final?
14. Why do you think there are two ways of creating object of String class
15. How to create our own immutable class

### **Exception Handling**

1. What is exception? & why to handle it?
2. If user don't handle exception, then who will handle?
3. What's benefit of user handling exception?
4. Why we shouldn't allow default exception handler to handle exception
5. Explain exception hierarchy in java
6. How many handlers does java provides to handle exception
7. Explain try catch block with an example
8. What statements we should write in the try block?
9. Explain what happens when exception occurs in the try block
10. Does catch block execute always?
11. Can one try block have multiple catch blocks? If yes which catch block will execute? What happens if the corresponding catch block not found?
12. Can we write generalized catch block & specialized catch block together?

13. How many exceptions occur in single try block? How to handle more than one exception?
14. Explain multiple try catch blocks.
15. What is the use of throw keyword
16. Explain the difference between JVM creating an exception object & user creating an exception object?
17. Explain exception propagation? & when exception object will propagate?
18. What is checked exception & unchecked exception?
19. When to use throws? Explain with an example.
20. Difference between **checked exception** & **unchecked exception**?
21. Difference between **throw** & **throws**
22. Difference between **throw** & **Throwable**
23. What is the importance of finally block? Explain with an example.
24. Difference between **final**, **finally** & **finalize()**
25. WAP to demonstrate custom checked exception
26. WAP to demonstrate custom unchecked exception

### **Collection Framework Library**

1. What is collection? & why to use collection?
2. Difference between array & collection
3. What is collection framework library & where is it present
4. Explain classes & interfaces available in collection framework library
5. Explain Collection(I) methods.
6. What is Collection & Collections



7. Explain the properties of List(I)
8. Explain the implementation classes of List(I)
9. Explain how the objects will be stored in ArrayList collection
10. Difference between ArrayList & LinkedList
11. When ArrayList is best choice
12. When LinkedList is best choice
13. Difference between List(I) & Set(I)
14. Explain the properties of Set(I)
15. Difference between HashSet & LinkedHashSet
16. Explain the cursors of java collection framework library
17. Explain the working behavior of Iterator(I) & ListIterator(I)
18. Difference between Enumeration(I), Iterator(I) & ListIterator(I)
19. Explain the properties SortedSet(I)
20. Explain the properties of TreeSet.
21. Explain Map(I) methods
22. What is an entry?
23. Difference between HashMap & HashTable