1. What is Object in Java and why we need object?
   1. An object is a combination of data and procedures working on the available data • An object has a state and behavior • The state of an object is stored in fields (variables), while methods (functions) display the object's behavior
   2. Objects are required in OOPs because they can be created to call a non-static function which are not present inside the Main Method but present inside the Class and provide the name to the space which is being used to store the data.
2. What is Inheritance and how many types of inheritance are supported by Java?
   1. INHERITANCE Is the ability to derive something specific from something generic.
   2. Single, Multiple, Multilevel, Hierarchy, Hybrid
3. What is the diamond problem in Java? And how can we resolve the problem?

The diamond problem occurs when two superclasses of a class have a common base class.

The solution to the diamond problem is default methods and interfaces.

1. What is Interface and what is abstract class? What are the differences between them?

* Like a class, an interface can have methods and variables, but the methods declared in interface are by default abstract (only method signature, no body)
* An abstract class is a class that contains at least one abstract method.

Abstract class vs Interface

Type of methods: Interface can have only abstract methods. An abstract class can have abstract and non-abstract methods.

Final Variables: Variables declared in a Java interface are by default final. An abstract class may contain non-final variables.

Type of variables: Abstract class can have final, non-final, static and non-static variables. The interface has only static and final variables.

Implementation: Abstract class can provide the implementation of the interface. Interface can’t provide the implementation of an abstract class.

Inheritance vs Abstraction: A Java interface can be implemented using the keyword “implements” and an abstract class can be extended using the keyword “extends”.

Multiple implementations: An interface can extend one or more Java interfaces, an abstract class can extend another Java class and implement multiple Java interfaces.

Accessibility of Data Members: Members of a Java interface are public by default. A Java abstract class can have class members like private, protected, etc.

1. What is Polymorphism? And how Java implements it?
   1. Polymorphism in java is a concept by which we can perform a single action by different ways.
   2. There are two types of polymorphism in java • Compile time polymorphism (overloading) • Runtime polymorphism (overriding)
2. What is the differences between overriding and overloading?
   1. When two or more methods in the same class have the same name but different parameters, it's called Overloading. When the method signature (name and parameters) are the same in the superclass and the child class, it's called Overriding.
3. What is Encapsulation? How Java implements it? And why we need encapsulation?
   1. Encapsulation is hiding information
   2. Making the fields in a class private and providing access to the fields via public methods
      * Flexibility — Internal logic changes won’t affect the caller of the method
      * Reusability — Encapsulated code can be used by different callers
      * Maintainability — Operations on encapsulated unit won’t affect others parts
4. What is the difference between abstraction and encapsulation?
   1. Encapsulation is hiding WHAT THE PHONE USES to achieve whatever it does Abstraction is hiding HOW IT DOES it

Encapsulation = Data Hiding + Abstraction

1. What is Final key word? (Filed, Method, Class)
   1. Java final keyword is a non-access specifier that is used to restrict a class, field, and method.

If we initialize a variable with the final keyword, then we cannot modify its value. If we declare a method as final, then it cannot be overridden by any subclasses. And, if we declare a class as final, we restrict the other classes to inherit or extend it.

1. What is toString() and why we need it?

The toString method is used to return a string representation of an object. If any object is printed, the toString() method is internally invoked by the java compiler. Else, the user implemented or overridden toString() method is called.

1. Can we use this keyword in constructor and why?
   1. **Yes**, we can call all the members of a class (methods, variables, and constructors) from instance methods or, constructors.
2. What is the new features of Java 8?
   1. 1. Functional Interface and Lambda Expressions 2. default and static methods in Interface 3. Java Stream API for Bulk Data Operations on Collections 4. forEach() method in Iterable interface
3. Where do we write a lambda expression? Try to give an example.
   1. lambda expressions implement the only abstract function and therefore implement

functional interfaces

example:

interface Square{ int area(int x) ;}

class Test {

public static void main(String[] args){

Square s= (int x)-> x\*x;

}

}

1. Why is the advantages brought by the Stream API?

make I/O operation fast.

A stream is a conceptually endless flow of data. We can either read from a stream or write to a stream. •

A stream is connected to a data source or a data destination

15. What is Exception and What is Error in Java? What are differences between them?

An exception is an unwanted or unexpected event, which occurs during the execution of a program i.e at run time, that disrupts the normal flow of the program’s instructions.

An Error indicates serious problem that a reasonable application should not try to catch

Exception indicates conditions that a reasonable application might try to catch.

16. What are types of Exception in Java? What is the differences? Why do we have two types

of Exception?

Checked exception

checked by the compiler at compile time

IOException, SQLException

Unchecked exception

checked by the JVM at run time

ArrayIndexOutOfBoundsException, NullPointerException

Checked Exception — Those who call a method must know about the exception so that they can handle properly. • Unchecked Exception — Runtime Exception may happen everywhere. Adding it to the method declaration will reduce the program clarity.

17. How can we handle an exception in Java?

try/catch block can be placed within any method that you feel can throw exceptions

18. What is custom Exception and why do we need to use custom Exception?

CUSTOM EXCEPTION is A class , which is sub class of Exception or RuntimeException.

Custom exceptions provide you the flexibility to add attributes and methods that are not part of a standard Java exception. These can store additional information, like an application-specific error code, or provide utility methods that can be used to handle or present the exception to a user.

19. What is the difference between final and finally key word?

The final keyword can be used with class method and variable. The finally keyword is used to create a block of code that follows a try block.

20. Is the following code legal? Why

Yes, The finally block is executed always after the try(-catch) block, if an exception is thrown or not.

21.

The code can run, but it will always be caught in exception a and never be caught in ArithmeticException a.

22.

1. 3 (compile error). The array is not initialized and will not compile.
2. 1 (error).
3. 4 (no exception). When you read a stream, you expect there to be an end of stream marker. You should use exceptions to catch unexpected behavior in your program.
4. 2 (checked exception).

23. What is Singleton Design Pattern and why do we use it? (lazy thread safe version)

Singleton pattern **restricts the instantiation of a class and ensures that only one instance of the class exists in the java virtual machine**. The singleton class must provide a global access point to get the instance of the class. Singleton pattern is used for logging, drivers objects, caching and thread pool.