Java Inheritance - Inheritance allows one class (child or subclass) to inherit the properties and methods of another class (parent or superclass). This promotes code reuse.

Polymorphism - Polymorphism allows objects of different classes to be treated as objects of a common superclass. It's about using a single interface to represent different underlying forms (data types).

Method Overloading in Java - Method overloading occurs when multiple methods in the same class have the same name but different parameters (different type or number of parameters).

Example: A class MathOperations could have methods add(int a, int b) and add(int a, int b, int c) to add two or three integers, respectively.

Method Overriding in Java - Method overriding allows a subclass to provide a specific implementation for a method that is already defined in its superclass.

Example: If the Vehicle class has a method move(), the Car subclass can override this method to provide its own specific way of moving.

Java Abstraction - Abstraction is the concept of hiding complex implementation details and showing only the essential features of an object

Java Encapsulation - Encapsulation is the practice of keeping an object's data (attributes) private and providing public methods to access and modify that data. This helps to protect the integrity of the data and prevents unauthorized or invalid access.

Rules for Java Method Overriding

Same Name: The method in the subclass must have the same name as in the superclass.

Same Parameters: The method must have the same parameter list as in the superclass.

IS-A Relationship: The subclass should have an IS-A relationship with the superclass.

Access Modifier: The access level of the overriding method cannot be more restrictive than the overridden method.

Constructors in Java - Constructors are special methods used to initialize objects. They have the same name as the class and do not have a return type.

Types of Java constructors

Default constructor (no-arg constructor) - A constructor is called "Default Constructor" when it doesn't have any parameter. Initializes objects with default values

Parameterized constructor - A constructor which has a specific number of parameters is called a parameterized constructor

Constructor Overloading in Java - Constructor overloading in Java is a technique that allows a class to have multiple constructors with different parameter lists.

Interface in java

An interface in Java is a reference type, similar to a class, but it is a collection of abstract methods. A class implements an interface, thereby inheriting the abstract methods of the interface. Interfaces are used to achieve abstraction and multiple inheritance in Java.

Exception Handling

Exception handling in Java is a mechanism to handle runtime errors gracefully, preventing the program from crashing.

Hierarchy of Exception classes - The class at the top of the exception class hierarchy is the Throwable class, which is a direct subclass of the Object class. Throwable has two direct subclasses - Exception and Error.

Types of Exception

Checked Exception - The classes that directly inherit the Throwable class except Runtime Exception and Error are known as checked exceptions. Checked exceptions are checked at compile-time.

Unchecked Exception - The classes that inherit the Runtime Exception are known as unchecked exceptions. Unchecked exceptions are not checked at compile-time, but they are checked at runtime.

try-catch-finally - The try block is used to enclose the code that may throw an exception. The catch block catches the exception thrown in the try block, and multiple catch blocks can be used to handle different types of exceptions. The finally block, which is optional, always executes after try-catch blocks, whether an exception is thrown or not, and is typically used for cleanup operations like closing resources.

**throw and throws:** The throw keyword is used to explicitly throw an exception within a method or block of code. The throws keyword is used in the method signature to declare that the method might throw certain types of exceptions, indicating that the caller of the method needs to handle those exceptions.

Multiple catch blocks:Multiple catch blocks allow handling different types of exceptions separately by providing multiple catch blocks, each catching a specific type of exception.

Exception Handling with method Overriding - When a superclass method (overridden method) declares that it can throw an exception, then the subclass method (overriding method) must also declare that it can throw the same kind of exception or a subtype of that exception.

Java Collection Framework

Hierarchy of Collection Framework: Java Collection Framework is a set of classes and interfaces that provide an architecture to store and manipulate groups of objects. It offers various data structures and algorithms to handle collections of objects efficiently. At the top of the hierarchy is the Collection interface, which is the root interface for all collection classes in Java. Collections are further categorized into List, Set, and Queue interfaces. The Map interface represents a collection of key-value pairs.

Collection interface

The Collection interface is the foundation of the Java Collection Framework. It represents a group of objects known as elements. It provides basic operations such as add, remove, and iterate over its elements.

Iterator interface

The Iterator interface provides a way to traverse through the elements of a collection. It allows sequential access to the elements of a collection and provides methods like hasNext() and next().

Set - Represents a collection of unique elements.

List - Represents an ordered collection that allows duplicate elements.

Queue - Represents a collection designed for holding elements before processing.

Map - Represents a collection of key-value pairs, where each key is associated with a value.

Comparator - Provides a way to compare objects for ordering.

ArrayList - Implements the List interface using a dynamic array. It is resizable and allows fast random access.

Vector - Similar to ArrayList but is synchronized, making it thread-safe but slower.

LinkedList - Implements the List interface using a doubly-linked list. It provides efficient insertion and deletion operations.

PriorityQueue - Implements the Queue interface using a priority heap. Elements are ordered based on their natural ordering or a specified comparator.

HashSet - Implements the Set interface using a hash table. It does not allow duplicate elements and provides constant-time performance for basic operations.

LinkedHashSet - Maintains insertion order of elements while also implementing the Set interface.

TreeSet - Implements the Set interface using a self-balancing binary search tree. It stores elements in sorted order.

HashMap - Implements the Map interface using a hash table. It stores key-value pairs and provides constant-time performance for basic operations.

ConcurrentHashMap - is a thread-safe implementation of the Map interface in Java, designed for high-concurrency environments, dividing the map into segments for independent synchronization, ensuring safe concurrent access and modification.