

Main Features of Java

Object Oriented

Simple

Platform Independent

Secure

Robust

Portable

Multithreaded

Distributed

Principles of OOP

Abstraction Abstraction means using simple things to represent complexity using object, classes, and variables.

Encapsulation This is the practice of keeping fields within a class private, then providing access to them via public methods. It's a protective barrier that keeps the data and code safe within the class itself.

Inheritance It lets programmers create new classes that share some of the attributes of existing classes. This lets us build on previous work without reinventing the wheel.

Polymorphism Polymorphism refers to any entity whether it is a method or a constructor or an operator which takes many forms or can be used for multiple tasks.

Multithreaded Programming

Multithreaded programming is one of the key features of java which allows multiple threads to execute their task simultaneously.

There are two ways to create threads in Java

a) by extending the *Thread* class

b) by implementing the *Runnable* interface

Final, Finally, Finalize

Final Makes a variable, method, or class unchangeable.

Finally Exception handling after success or failure during a try catch block.

Finalize Called by the garbage collector on an object when garbage collection determines that there are no more references to the object.

Inheritance

Inheritance is one of the key principle of object oriented programming. Through inheritance, one class can inherit the properties of another class. The class from which properties are inherited is called super class and the class to which properties are inherited is called sub class.

Does Java support multiple inheritance?

Java supports multiple inheritance but only through interfaces. That means a class can implement more than one interfaces but can not extend more than one class.

Constructor & Method Overloading

A class can have any number of constructors. These constructors will have different list of arguments. It is called constructor overloading. Constructor overloading provides different ways to instantiate a class.

If a class has more than one method with same name but with different list of arguments, then it is called method overloading.

Overriding

If a super class method is modified in the sub class then it is called method overriding.

Abstract Classes

An abstract class is a class which cannot be instantiated. An abstract class is used by creating an inheriting subclass that can be instantiated. An abstract class does a few things for the inheriting subclass:

1. Define methods which can be used by the inheriting subclass.
2. Define abstract methods which the inheriting subclass must implement.
3. Provide a common interface which allows the subclass to be interchanged with all other subclasses.

Exception Handling

try The code to be monitored for exceptions will be kept in this block.

catch If any exceptions occurred in try block, those exceptions will be caught by this block.

finally This block will be always executed whether exception is raised or not and raised exceptions are caught or not.

Checked & Unchecked Exceptions

Checked	Known and checked at compile time.
Unchecked	Known at Runtime, unknown at compile time.

Static & Non-Static Methods

Static method is common to all instances of a class. *Static* methods are stored in the class memory. Where as *non-static* methods are stored in the object memory. Each instance of a class will have their own copy of non-static methods.

Synchronization

Synchronization is a way of controlling the access of a method or a block by multiple threads. Only one thread can enter into a method or a block which has been declared as synchronized. Synchronization is one of the way to achieve thread safety.

Garbage Collection

Removing unwanted objects or abandoned objects from the memory is called garbage collection. Garbage collection is done automatically in java. You need not to remove the unwanted objects explicitly. Garbage collector thread does this for you.

Cloning

Cloning is a process of creating an exact copy of an existing object in the memory. Cloning may be shallow or deep. In java, clone() method is used to create a clone of an object.

Interfaces

A Java interface is a bit like a class, except a Java interface can only contain method signatures and fields. An Java interface cannot contain an implementation of the methods, only the signature (name, parameters and exceptions) of the method. You can use interfaces in Java as a way to achieve polymorphism.

