Java isa popular, high-level, object-oriented programming language and computing platformknown for its "Write Once, Run Anywhere" (WORA) capability, allowing code to run on any device with a Java Virtual Machine (JVM). It's widely used for building mobile apps, enterprise software, web applications, games, and big data solutions, offering a robust, secure, and platform-independent environment for developers

Java is a high-level, object-oriented programming language used to build web apps, mobile applications, and enterprise software systems.

Known for its Write Once, Run Anywhere capability, which means code written in Java can run on any device that supports the Java Virtual Machine (JVM).
Syntax and structure is similar to C-based languages like C++ and C#.

Basic Code Example
The following is a simple program that displays the message “Hello, World!” on the screen.

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

Why Learn Java?
Used to build Android apps, desktop and web apps, enterprise backend systems, and cloud-based software.
In high demand with many job opportunities in software development.
Has popular frameworks like Spring and Hibernate which makes it powerful for enterprise applications.
Supports object-oriented programming for clean, maintainable and reusable code.
Top companies like Amazon, Netflix, and LinkedIn use Java.
Basics
Java basics form the foundation of your programming journey, covering essential concepts like syntax, data types, variables, loops, and conditionals.

Introduction
Download and Install Java
JDK vs JRE vs JVM
Identifiers
Keywords
Quiz: Java Basics
Data Types
Variables
Operators
Quiz: Variables, Operator
Decision Making
Loops & Jump Statements
Quiz: Control Statements and Loops
Project: Number Guessing Game
Methods
Java methods are reusable blocks of code that perform specific tasks and help organize your program. They improve code readability, reduce repetition, and make debugging easier.

Introduction to Methods
Static Methods vs Instance Methods

Access Modifiers
Command Line Arguments
Variable Arguments (Varargs)
Quiz: Methods
Arrays
Java arrays are containers that store multiple values of the same data type in a single variable. They provide an efficient way to manage and access collections of data using index-based positions.

Introduction
Declare and Initialize Arrays
Multi-Dimensional Arrays
Quiz: Java Arrays
Jagged Arrays
Arrays Class
Final Arrays
Quiz: Array
Projects: Tic-Tac-Toe Game
Strings
Java Strings represent sequences of characters and are widely used in text processing. They are immutable, meaning once created, their values cannot be changed.

Introduction
Why Strings are Immutable
String Concatenation
String Methods
Quiz: String Basics
String Class
StringBuffer Class
StringBuilder Class
Quiz: String Classes
Strings vs StringBuffer vs StringBuilder
Regex
Java Regex (Regular Expressions) allows pattern matching and text manipulation using the java.util.regex package. It is powerful for validating, searching, and replacing strings based on specific patterns.

What is Java Regex?
Pattern Class
Matcher Class
Character Class
Quantifiers
Quiz: Regex Basics and Pattern Matching
OOP Concepts
Java follows the Object-Oriented Programming (OOP) paradigm, which organizes code into classes and objects. Core OOP principles like inheritance, encapsulation, polymorphism, and abstraction make Java modular and scalable.

What are OOP Concepts?
Classes and Objects
Quiz: Classes and Objects
Constructors
Quiz: Constructors
Object Class
Abstraction
Encapsulation
Inheritance
Quiz: Inheritance and Abstraction

Polymorphism
Packages
Quiz: Polymorphism and Packages
Project: Simple Banking Application
Interfaces
Java interfaces define a contract that classes must follow, specifying method signatures without implementations. They enable abstraction and support multiple inheritance in Java through a clean, structured approach.

Interfaces
Class vs Interface
Quiz: Interfaces
Functional Interface
Nested Interface
Marker Interface
Quiz: Interface types and Comparator
Project: Employee Management System
Exception Handling
Java Exception Handling is a mechanism to handle runtime errors, ensuring the program runs smoothly without crashing. It uses keywords like try, catch, throw, throws, and finally to manage exceptions.

Exceptions
Quiz: Java Exceptions
Final, Finally and Finalize
Throw and Throws
Customized Exception Handling
Chained Exceptions
Null Pointer Exceptions
Exception Handling with Method Overriding
Quiz: Exception Handling
Memory Allocation
Java Memory Allocation refers to how memory is assigned to variables, objects, and classes during program execution. It involves stack and heap memory, with the JVM managing allocation and garbage collection automatically.

Java Memory Management
How Java Objects Stored in Memory?
Quiz: Java Memory Allocation
Types of Memory Areas Allocated by JVM
Stack vs Heap Memory Allocation
Quiz: Heap vs Stack
Garbage Collection
Quiz: JVM Memory Management and Garbage Collection
Types of JVM Garbage Collectors
Memory Leaks
Collections
Java Collections provide a framework for storing and manipulating groups of objects efficiently. It includes interfaces like List, Set, and Map, along with classes like ArrayList, HashSet, and HashMap.

Collections Class
Collection Interface
Quiz: Collection
List Interface
ArrayList
LinkedList
Quiz: List, ArrayList, LinkedList

Set Interface
HashSet
TreeSet
Quiz: Set and HashSet
Queue Interface
Priority Queue
Deque Interface
Map Interface
HashMap
Quiz: Queue and Map Interface
Iterator
Comparator Interface
Comparable Interface
Quiz: Iterators, Comparator vs Comparable
Project: Face Detection System
Lambda Expressions and Streams
Java Streams and Lambda Expressions simplify data processing by enabling functional-style operations on collections. Lambdas provide concise syntax for anonymous functions, while Streams allow efficient filtering, mapping, and reduction of data.

Lambda Expressions
Method References
Java Streams
Quiz: Lambda Expressions and Streams
To know Java 8 in detail, refer to - Java 8 Features
Multithreading and Synchronization
Java Multithreading allows concurrent execution of two or more threads, enabling efficient CPU utilization and faster program performance. It is commonly used for tasks that required parallel processing and responsiveness from multiple ends.

Introduction
Threads
Thread.start() vs Thread.run() Method
Thread.sleep() Method
Runnable Interface
Quiz: Thread Basics and Lifecycle
Main Thread
Thread Priority in Multithreading
Daemon Thread
Quiz: Thread Methods and Daemon Threads
Java Synchronization
Quiz: Synchronization Basics
Thread Safety
Locks in Java
Lock vs Monitor in Concurrency
Lock Framework vs Thread Synchronization
Reentrant Lock
Deadlock in Multithreading
Thread Pools
Executor Framework
Quiz: Deadlocks and Synchronization
Project: Snake Game
File Handling
Java File Handling enables programs to create, read, write, and manipulate files stored on the system. It uses classes from the java.io and java.nio packages for efficient file operations.

Introduction to Java IO

Reader Class
Writer Class
File Handling
File Class
Quiz: File Handling
FileInputStream
FileOutputStream
FileReader Class
FileWriter Class
FileOutput Stream
BufferedReader Input Stream
BufferedReader Output stream
Fast I/O
Quiz: File Writing
FilePermission Class
FileDescriptor Class
Project: Text Editor
Networking
Java Networking enables communication between devices over a network using classes from the java.net package. It supports protocols like TCP and UDP for building client-server applications and data exchange.

Introduction to Java Networking
Quiz: Networking Basics and Protocols
Socket Programming
Server Socket Class
Quiz: Sockets and Server Communication
URL Class and Methods
Project: Chat Application
Java Database Connectivity(JDBC)
Introduction to Java JDBC
JDBC Driver
JDBC Connection
Types of Statements in JDBC