

CORE JAVA:

Day1

- Introduction
- Installation(JDK/Eclipse)
- Features
- Byte code
- JVM/JRE/JDK
- Platform Independent
- First program (Hello World). Compile and Execute in notepad
- Explain Class
- What is mainmethod?
- First program(Hello World) in Eclipse

Day2

- Types of comments
 1. Single line comment
 2. Multi line comments
 3. Document
- Keywords and Reserved words in Java
- Variables
- Data Types in Java
 - Primitive data types
 - Non-primitive data types
- Operators
 - Arithmetic Operators
 - RelationalOperators
 - Assignment Operators
 - Bitwise Operators
 - Logical Operators
 - Ternary Operator

Day 3

- Decision Control Statements
 - Selection/Conditional Statements
 1. Simple if Statement
 2. if...else Statement
 3. Nested if statement
 4. [if...else if...else statement](#)
 5. [Switch statement](#)
 - Looping Statements in Java
 1. While
 2. Do...while
 3. [For](#)
 - Branching Statements in Java
 1. Break
 2. Continue
 3. Return

Day 4

- Class (Re-Visited)
- Object
- Methods
- Types Variables
 - local variable
 - instance variable
 - static variable

Day 5

- Constructors

- Instance block
- this Keyword
- static method and static block
- main method Explanation(Re-Visited)

Day 6

- Inheritance(IS-A)
 - Single Inheritance
 - Multilevel Inheritance
 - Hierarchical Inheritance
 - Multiple inheritance
 - Hybrid Inheritance
- super keyword
- super() method
- Aggregation(HAS-A)

Day 7

- Polymorphism
 - Static Polymorphism
 - Dynamic Polymorphism

Day 8

- Abstraction
 - Interface
 - abstract
- Encapsulation

Day 9

- Arrays
- For each
- Command line Arguments

Day 10

- Packages
- Access Modifiers
 - default
 - private
 - protected
 - public
- Non - Access Modifiers

Day 11

- String
- String Buffer
- String Builder

Day 12

- Wrapper Classes
 - Boolean
 - Character
 - Byte
 - Short
 - Integer
 - Long
 - Float
 - Double
- Autoboxing and Unboxing

- Inner classes
 - Nested Inner class
 - Local inner classes
 - Anonymous inner classes
 - Static nested classes

Day 13, 14 and 15

- Collections
 - List
 1. ArrayList
 2. Linked List
 3. Vector
 4. Stack
 - Set
 1. HashSet
 2. LinkedHashSet
 3. TreeSet
 - Map
 1. HashMap
 2. LinkedHashMap
 3. TreeMap
- Iterators
 - Iterator
 - List Iterator
 - Enumeration
- fail-fast and fail-safe
- Comparable and Comparator interface
- HashMap inside implementation

- Hashing and Bucket
- Contract Between equals() and hashCode()

Day 16

- Generics
 - Need for Generics
 - Generic Classes
 - Generic Methods
 - Wildcards

Day 17

- Exceptions and Errors
 - Types of Exceptions
 1. Checked Exceptions
 2. Un-Checked Exceptions
 - Handling Exceptions
 - API hierarchy for Exceptions
 - Errors and Handling
 - Keywords in Exception API
 1. try,
 2. catch,
 3. finally,
 4. throw,
 5. throws
 - User Define Exception

Day 18

- Files and Input/output
 - Streams
 1. Byte Input Stream
 2. Byte Output Stream
 3. Character Input Stream
 4. Character Output Stream
 - Directories

Day 19

- Serialization
- Deserialization
- Serializable
- Transient
- Transient vs. Static

Day 20

- Threads

Day 21

- Garbage Collection

Day 22

- Annotations
- Enum

Day 23

- Object Cloning
 - Shallow Copy
 - Deep Copy

Day 24

- Garbage Collection

Day 25

- Utility Classes
 - Java Date Class (Java 8)
 - Calendar In
 - Java Random Class
 - StringTokenizer
 - Math Class

Day 26

- Reflection API

Day 27

- Java 5 Features
 - For-each loop (Java 5)
 - Varargs (Java 5)
 - Static Import (Java 5)
 - Autoboxing and Unboxing (Java 5)
 - Enum (Java 5)
 - Covariant Return Type (Java 5)
 - Annotation (Java 5)
 - Generics (Java 5)
- Java 7 Features

- String in switch statement (Java 7)
- Binary Literals (Java 7)
- The try-with-resources (Java 7)
- Caching Multiple Exceptions by single catch (Java 7)
- Underscores in Numeric Literals (Java 7)

Day 28

- Java8 Features
 - Lambda Expressions (Java 8)
 - Method References (Java 8)
 - Functional Interfaces (Java 8)
 - Stream (Java 8)
 - Base64 Encode Decode (Java 8)
 - Default Methods (Java 8)
 - forEach Method(Java 8)
 - Collectors(Java 8)
 - Optional class (Java 8)

