- **Basics of Java**
  - Java - What, Where and Why?
  - History and Features of Java
  - Internals of Java Program (bytecode)
  - Difference between JDK, JRE and JVM
  - The evolution of Java
- **An overview of Java**
  - Object-oriented programming
    - The OOP principles
  - First java program (practical)
    - The main method in java
    - Block of code
    - Example of java file and class file (bytecode)
- **Data Types and variables** (practical)
  - Primitive types
    - Integers - byte, short, int, long
    - Floating-point types - float, double
    - Characters
    - Booleans
  - Variables
    - Declaring a variable
    - Dynamic initialization
    - The scope and lifetime of a variable
  - Type conversion and casting
    - Automatic type conversion
    - Converting incompatible types
- **Arrays** (practical)
  - One-dimensional array
  - Multi-dimensional array
  - Different array declaration syntax
- **Operators** (practical)
  - Arithmetic operators
    - Basic arithmetic operators
    - Modulus operator
    - Increment and decrement operator
  - Relational operator
  - Assignment operator
  - The bitwise operator
    - Bitwise logical operator
    - Left shift
    - Right shift
    - Unsigned right shift
  - Boolean logical operator
    - Short-circuit logical operator
  - The ternary operator
  - Operator precedence
  - Using parentheses

- **Control statements** (practical)
  - Java's selection statement
    - If
    - Switch
  - Iteration statements
    - While
    - Do-while
    - For
    - Nested loops
  - Jump statements
    - Using break
    - Using continue
  - Advance iteration statements
    - Advance for loop (for each version)
    - Java 8 stream
- **Classes and Methods** (practical)
  - Class fundamentals
  - Declaring objects
    - New variable deep dive
  - Assigning object reference to a variable
  - Methods
    - Defining one method
    - Returning a value
    - Take parameter to that method
  - Constructors
    - Parameterized constructors
    - Private constructors
  - this keyword
  - Garbage collection
    - The finalize() method
- **More on Classes and methods** (practical)
  - Overloading methods
    - Overloading constructors
  - Using object as parameter
  - Returning objects
  - Recursion
  - Access control
    - Private, public, protected
  - static keyword
  - final keyword
  - Nested and inner classes
  - Using command line arguments
  - Using variable-length arguments
- **Inheritance** (practical)
  - Basic of inheritance
    - Member access
    - Super and subclass concept

    - super keyword
    - Creating multilevel hierarchy
    - When constructors are called/constructor chaining
    - Method overriding
    - Dynamic method dispatch
    - Abstract class
    - final revisited with inheritance
    - The object class
- **Packages and interface**s (practical)
    - Package
    - Access protection
    - Importing packages
    - Interfaces
        - Defining an interface
        - Implementing an interface
        - Nested interface
        - Interface can be extended
- **Exception handling**
    - Exception fundamentals
    - Types of exceptions
    - Handle exception (practical)
        - Using try-catch
            - Multiple catch clauses
            - Nested try statements
        - throw, throws and finally keyword
    - Java's built in exception (practical)
    - Custom exception (practical)
    - Chained exceptions (practical)
- **Multithreading** (practical)
    - Java thread model
        - Thread priorities
        - Synchronization
        - Messaging
        - The thread class and Runnable interface
    - The main thread
    - Creating a thread
        - Implementing runnable
        - Extending thread class
    - Using isAlive() isJoin() method
    - Suspending, resuming and stopping threads
- **String class** (practical)
    - Example of string methods
    - StringBuilder and StringBuffer
- **Enumerations** (practical)
- **Autoboxing** (practical)
    - Autoboxing fundamentals
    - Autoboxing in expressions

- **Generics** (practical)
  - Basics with example
  - Bounded types
  - Using wildcard arguments
  - Creating a generic method
    - Generic constructor
  - Generic class hierarchy
    - Generic super class
    - Generic subclass
    - Casting
  - Restrictions
- **Wrapper classes** (practical)
- **I/O basics** (practical)
  - Basic
  - Stream - byte and character streams
  - Reading console input
  - Writing console output
  - Reading and writing files
  - Using Scanner class
- **Collection framework** (practical)
  - Interfaces
    - Collection
    - List
    - Set
    - SortedSet
    - Map
  - Classes
    - ArrayList class
    - LinkedList class
    - HashSet
    - LinkedHashSet
    - TreeSet
    - HashMap
  - Accessing collection via iterator
  - Accessing map with NavigableMap interface
  - Comparators
- **Java 8**
  - Revisit the code with Java 8 features
    - Stream api
    - Thread model
      - Locking
      - Semaphore
      - Concurrency handler
  - New Date API
  - StringJoiner
  - Optional
- **Java with Database, JDBC**

    - JDBC Drivers
    - Steps to connect to the database
    - Connectivity with MySQL
    - DriverManager
    - Connection interface
    - Statement interface
    - ResultSet interface
    - PreparedStatement
    - ResultSetMetaData
    - DatabaseMetaData
    - Stored procedures and functions
    - Transaction Management
    - Batch Processing
- **Reflection API**
- **Mini project**