

# LANGUAGE FUNDAMENTALS

## 1) Class

- a) Attributes or Members
- b) Methods or Functions
- c) Constructor
- d) Main method
- e) Object

## 2) Variables

- a) Variables
- b) Create Variables
- c) Change values of variables
- d) Rules for Naming Variables
- e) Types of Variables
  - i) Local Variables
  - ii) Instance Variables
  - iii) Static Variables

## 3) Literals

## 4) Identifiers

- a) Identifiers
- b) Rules for Identifiers

## 5) Java Keywords

## 6) Data Types

- a) Primitive
  - i) byte
  - ii) short
  - iii) int
  - iv) long
  - v) float
  - vi) double
  - vii) Char
  - viii) boolean
- b) Object

## **7) Operators**

- a) Arithmetic Operators
- b) Assignment Operators
- c) Relational Operators
- d) Logical Operators
- e) Unary Operators
- f) instanceof Operator
- g) Ternary Operator

## **8) Input & Output**

- a) Single-line comment
- b) Multi-line comment

## **9) Comments**

## **10) Flow Controls**

- a) if, if-else, if-else-if
- b) switch
- c) for loop
- d) foreach
- e) while
- f) do-while
- g) break
- h) continue

## **11) Arrays**

- a) Declaration
- b) Initialization
- c) Access Elements
- d) Looping Array Elements
  - i) for loop
  - ii) for each

## **12) Command line arguments**

## **13) Main Method**

## **14) Coding Standards**

# OOPS LEVEL - 1

## 1) Class & Objects

- a) Creating Class
- b) Creating Objects
- c) Accessing Members of Class
- d) Creating Objects in Different Class
- e) Creating Objects in the Same Class
- f) Methods

## 2) Methods

- a) Method Types
- b) User-defined Methods
- c) Standard Library Methods
- d) Advantages of Methods
- e) Constructor

## 3) Constructor

- a) Types of Constructors
- b) No-Arg Constructor
- c) Parameterized Constructor
- d) Default Constructor
- e) Constructor Overloading
- f) Importance of Constructors

## 4) this keyword

- a) Usage
- b) Getter & Setter
- c) this in Getter & Setter

## 5) Access Modifiers

- a) Types
- b) Default
- c) Private
- d) Protected
- e) Public

## **6) Strings**

- a) Creating Strings**
- b) String Operations (Additional Topics)**
- c) create String with new keyword**

# **OOPS LEVEL - 2**

## **1) Inheritance**

- a) Subclass**
- b) Superclass**
- c) Inheritance**
- d) is-a Relationship**
- e) super Keyword**
- f) Protected Members in Inheritance**
- g) Importance of Inheritance**
- h) Types of Inheritance**

## **2) Polymorphism**

- a) Polymorphism**
- b) Why Polymorphism?**
- c) Polymorphism Ways**
- d) Method Overloading**
- e) Method Overriding**

## **3) Super Keyword**

- a) Usage**
- b) Access Overridden Methods of the Superclass**
- c) Access Attributes of the Superclass**
- d) Use of super() to Access Superclass Constructor**

## **4) Abstract Class & Methods**

- a) Abstract Class**
- b) Abstract Method**
- c) Implementing Abstract Methods**
- d) Accessing Constructor of Abstract Classes**
- e) Java Abstraction**

## **5) Interfaces**

- a) **Interface**
- b) **Implementing an Interface**
- c) **Implementing Multiple Interfaces**
- d) **Extending an Interface**
- e) **Extending Multiple Interfaces**
- f) **Advantages of Interface**
- g) **Default Methods in Java Interfaces (Java 8 Feature)**
- h) **Private and Static Methods in Interface (Java 8 Feature)**

## **6) Encapsulation**

- a) **Encapsulation**
- b) **Why Encapsulation?**
- c) **Data Hiding**

# **COLLECTION FRAMEWORK**

## **1) Collection Framework**

- a) **Interfaces of Collections Framework**
- b) **Collection Interface**
- c) **Collections Framework vs. Collection Interface**
- d) **Subinterfaces of the Collection Interface**
- e) **Why the Collections Framework?**
- f) **Arrays vs Collections**

## **2) Java List**

- a) **Classes that Implement List**
- b) **How to use List?**
- c) **Methods of List**
- d) **Implementation of the List Interface**
  - i) **ArrayList**
    - (1) **Java ArrayList vs Array**
    - (2) **Creating an ArrayList**
    - (3) **Basic Operations on ArrayList**
    - (4) **Methods**
    - (5) **Iterate through an ArrayList**

### **(6) Basic Operations on ArrayList**

- (a) Add elements**
- (b) Access elements**
- (c) Change elements**
- (d) Remove elements**

#### **ii) LinkedList**

- (1) Creating a LinkedList**
- (2) Working of a LinkedList**
- (3) LinkedList Methods**
- (4) Other Methods**

#### **e) Vector**

- i) Creating a Vector**
- ii) Methods of Vector**
- iii) Other Methods**

#### **f) Stack**

- i) Creating a Stack**
- ii) Stack Methods**

### **3) Set Interface**

- a) Classes that Implement Set**
- b) Interfaces that Extend Set**
- c) How to use Set?**
- d) Methods of Set**
- e) Set Operations**
- f) Implementation of the Set Interface**
- g) HashSet Class**
- h) Creating a HashSet**
- i) Methods of HashSet**
- j) Other Methods**
- k) Why HashSet?**
- l) LinkedHashSet Class**
- m) LinkedHashSet**
- n) Create a LinkedHashSet**
- o) Methods of LinkedHashSet**
- p) Other Methods**

- q) **LinkedHashSet Vs. HashSet**
- r) **TreeSet**
- s) **Creating a TreeSet**
- t) **Methods of TreeSet**
- u) **Other Methods**
- v) **TreeSet Vs. HashSet**
- w) **Iterator Interface**
- x) **ListIterator Interface**
- 4) **Map Interface**
  - a) **Working of Map**
  - b) **Use of Map**
  - c) **Methods of Map**
  - d) **Implementations**
  - e) **HashMap**
  - f) **LinkedHashMap**
  - g) **WeakHashMap**
  - h) **SortedMap**
  - i) **NavigableMap**
  - j) **TreeMap**
  - k) **ConcurrentMap**
  - l) **ConcurrentHashMap**

## EXCEPTION HANDLING

- 1) **What are Exceptions**
- 2) **Exception hierarchy**
- 3) **Errors**
- 4) **Exceptions**
- 5) **Java Exception Types**
  - a) **RuntimeException (Unchecked)**
  - b) **IOException (Checked)**
- 6) **Exception Handling**
  - a) **Java try...catch block**
  - b) **Java finally block**
  - c) **throw and throws keyword**

- d) try...catch block
  - e) try...finally block
  - f) try...catch...finally block
  - g) Multiple Catch blocks
  - h) Catching Multiple Exceptions
- 7) throw and throws
  - 8) Try-with-resources

## MULTI THREADING

- 1) Introduction
- 2) Ways to Instantiate Thread
- 3) Getting & Setting Names for Threads
- 4) Thread Priorities
- 5) Thread Prevention Methods
- 6) Synchronization
- 7) Deadlock

## JAVA LANG PACKAGE

- 1) Object Class Methods
- 2) String
- 3) StringBuffer
- 4) StringBuilder
- 5) Wrapper Classes
- 6) AutoBoxing & AutoUnboxing
- 7) Cloning
- 8) Type Casting



# SERIALIZATION

- 1) Introduction
- 2) Serialization
- 3) Deserialization
- 4) Transient Keyword
- 5) Static vs Transient
- 6) Transient vs Final
- 7) Marker Interface

# IO PACKAGE

- 1) File Class
- 2) FileWriter
- 3) FileReader
- 4) BufferedWriter
- 5) BufferedReader
- 6) PrintWriter
- 7) Module 10: Reflection
- 8) What is Reflection
- 9) Reflection On
- 10) Fields
- 11) Methods
- 12) Constructor

# GENERICS

- 1) Create Generics Class
- 2) Generics Method
- 3) Advantages

# ENUMS

- 1) **Java Enum**
- 2) **Enum Class**
- 3) **Methods of Enum**
- 4) **ordinal()**
- 5) **compareTo()**
- 6) **toString()**
- 7) **name()**
- 8) **valueOf()**
- 9) **values()**
- 10) **Why Enums?**

# GARBAGE COLLECTION

- 1) **Introduction**
- 2) **Ways to Make Object Eligible for GC**
- 3) **Methods to Run GC**
- 4) **finalize()**

# JAVA 8 STREAM API

- 1) **Introduction**
- 2) **Lambda Expressions**
- 3) **Default & Static Methods**
- 4) **Stream API**
- 5) **Date and Time API**
- 6) **Optional Class**
- 7) **CompletableFuture**

## JSP & SERVLETS

- 1) Introduction
- 2) JSP
- 3) Servlets

## JDBC

- 1) Introduction
- 2) Sample Example
- 3) CRDU Application

## ADDITIONAL FREE TOPICS

- 1) Git
- 2) Maven
- 3) Debugging
- 4) Design patterns
- 5) Data Structures
- 6) Agile
- 7) Jira
- 8) Mail & Teams Communication
- 9) Coding Programs
- 10) Mock Interviews
- 11) Unit Testing
- 12) Logging
- 13) CRDU Application