JAVA CONTENT

Difference Between Core Java and Advanced Java

Parameter	Core Java	Advanced Java
Definition	A user won't be able to develop or understand any application using the JAVA EE (Java Enterprise Edition) in the absence of the fundamentals and basic concepts of Core Java.	From understanding all the database manipulations to the development of sophisticated web pages- Advanced Java basically derives from Core Java.
Areas or Topics	Core Java covers some topics like data types, OOP, operators, exception handling, swing, threading, and collections.	Advanced Java covers some topics like web services, database connectivity, JSP, Servlets, EJB, etc.
Uses	Core Java assists a user in building some general-purpose applications. These applications are generally independent of any platform.	Advanced Java caters to building some enterprise-level apps required for more sophisticated uses. For instance, it helps a user create mobile and web applications, and more.
Type of Application	This programming basically covers the socket, swings, AWT, classes, collection objects, and thread concepts.	Advanced java programming only works for enterprise and web-based applications.
Architecture	Core Java makes use of the single-tier architecture. Thus, it is known as *a stand-alone* application. In a single-tier application, there is zero third-party interaction. Core Java makes use of device storage or heap storage- localized to cater to the user's needs.	It makes use of the two-tier architecture, meaning the backend or server-side application and client-side architecture. It requires the use of more robust Client-server architecture.
Edition/Version of Java	The J2SE (Java Standard Edition) falls under Core Java.	The J2EE (Java Enterprise Edition) falls under Advanced Java.

Purpose	Core Java does not deal with socket programming, databases, etc.	Advanced Java deals with DOM, socket programming, and network applications.
Package	This programming language provides the java.lang.* package.	This programming language provides the java.servlet.* package.

Core Java and Advanced Java Topics

Introduction

- Programming language Types and Paradigms
- Computer Programming Hierarchy
- How Computer Architecture Affects a Language ?
- Why Java ?
- Flavors of Java,
- Java Designing Goal,
- Role of Java Programmer in Industry
- Features of Java Language, JVM –The heart of Java ,
- Java's Magic Bytecode.

The Java Environment

- Installing Java,
- Java Program Development ,
- Java Source File Structure
- Compilation, Executions

Basic Language Elements

- Lexical Tokens
- Identifiers, Keywords,
- Literals,
- Comments
- Primitive Datatypes,
- Operators Assignments.

Object Oriented Programming

- Object & Object reference
- Object Life time & Garbage Collection
- Creating and Operating Objects
- Constructor & initialization code block,

- Access Control, Modifiers,
- Inner Class &Anonymous Classes
- Abstract Class & Interfaces Defining Methods,
- Method Overloading
- Recursion
- Dealing with Static Members,
- Finalize() Method
- Native Method
- Use of “this “ reference
- Use of Modifiers with Classes & Methods
- Design of Accessors and Mutator Methods Cloning Objects
- shallow and deep cloning, Generic Class Types.
- Class Fundamentals

Extending Classes and Inheritance

- Use and Benefits of Inheritance in OOP
- Types of Inheritance in Java
- Inheriting Data members and Methods
- Role of Constructors in inheritance
- Overriding Super Class Methods
- Use of “super” , Polymorphism in inheritance
- Type Compatibility and Conversion Implementing interfaces.

Package

- Organizing Classes and Interfaces in Packages
- Package as Access Protection
- Defining Package
- CLASSPATH Setting for Packages
- Making JAR Files for Library Packages Import and Static Import
- Naming Convention For Packages.

Exception Handling

- The Idea behind Exception ,Exceptions & Errors
- Types of Exception
- Control Flow In Exceptions
- JVM reaction to Exceptions
- Use of try, catch, finally, throw, throws in Exception Handling
- In-built and User Defined Exceptions,
- Checked and Un-Checked Exceptions.

Array & String

- Defining an Array

- Initializing & Accessing Array
- Multi –Dimensional Array, Operation on String
- Mutable & Immutable String, Using Collection Bases Loop for String
- Tokenizing a String
- Creating Strings using StringBuffer .

Thread

- Understanding Threads
- Needs of Multi-Threaded Programming
- Thread Life-Cycle
- Thread Priorities
- Synchronizing Threads
- Inter Communication of Threads
- Critical Factor in Thread –DeadLock

A Collection of Useful Classes

- Utility Methods for Arrays
- Observable and Observer Objects
- Date & Times
- Using Scanner Regular Expression
- Input/Output Operation in Java(java.io Package)
- Streams and the new I/O Capabilities
- The Classes for Input and Output
- The Standard Streams
- Working with File Object
- File I/O Basics, Reading and Writing to Files,
- Buffer and Buffer Management
- Read/Write Operations with File Channel, Serializing Objects .

GUI Programming

- Designing Graphical User Interfaces in Java
- Components and Containers
- Basics of Components
- Using Containers
- Layout Managers
- AWT Components,
- Adding a Menu to Window
- Extending GUI Features Using Swing Components,
- Java Utilities (java.util Package)

The Collection Framework

- Collections of Objects

- Collection Types, Sets ,
- Sequence, Map
- Understanding Hashing
- Use of ArrayList & Vector.

Event Handling

- Event-Driven Programming in Java
- Event- Handling Process
- EventHandling Mechanism,
- The Delegation Model of Event Handling
- Event Classes, Event Sources, Event Listeners
- Adapter Classes as Helper Classes in Event Handling.

Database Programming using JDBC

- Introduction to JDBC
- JDBC Drivers & Architecture,
- CURD operation Using JDBC
- Connecting to non-conventional Databases.

Java Server Technologies Servlet

- Web Application Basics
- Architecture and challenges of Web Application
- Introduction to servlet
- Servlet life cycle, Developing and Deploying Servlets
- Exploring Deployment
- Descriptor (web.xml)
- Handling Request and Response.