



Shri Vile Parle Kelavani Mandal's

# Dwarkadas J. Sanghvi College of Engineering

*(Autonomous College Affiliated to the University of Mumbai)*

Scheme and Detailed Syllabus (DJS23)

of

Honors Degree Program

in

**DevOps (Development and Operations)**

*Revision: 2*

*With effect from the Academic Year: 2024-2025*

Sr.	Course Code	Course	Teaching Scheme (hrs.)				Continuous Assessment (A) (marks)			Semester End Assessment (B) (marks)					(A+B)	Total Credits
			Th	P	T	Credits	Th	T/W	Total CA (A)	Th	O	P	O & P	Total SEA (B)		
<b>Sem III</b>																
1	DJS23ICH1301	Development Frameworks	4	--	--	4	40	--	40	60	--	--	--	60	100	4
<b>Sem IV</b>																
2	DJS23ILH1401	Advanced Java Laboratory	--	4	--	2	--	25	25	--	--	--	25	--	50	2
<b>Sem V</b>																
3	DJS23ICH1501	DevOps	3	--	--	3	40	--	40	60	--	--	--	60	100	3
4	DJS23ILH1501	DevOps Lab	--	2	--	1	--	25	25	--	--	--	--	--	25	1
<b>Sem VI</b>																
5	DJS23ICH1601	DevSecOps	3	--	--	3	40	--	40	60	--	--	--	60	100	3
6	DJS23ILH1601	DevSecOps Lab	--	2	--	1	--	25	25	--	--	--	--	--	25	1
<b>Sem VII</b>																
7	DJ19ITHN1C4	MLOps and Cloud Engineering	4	--	--	4	40	--	40	60	--	--	--	60	100	4
		<b>Total</b>	<b>14</b>	<b>8</b>	<b>0</b>	<b>18</b>	<b>160</b>	<b>75</b>	<b>235</b>	<b>240</b>	<b>0</b>	<b>0</b>	<b>25</b>	<b>240</b>	<b>500</b>	<b>18</b>

**Course: Development Frameworks (DJS23ICH1301)**

**Pre-requisite:** Knowledge of any programming language and Database Management System

**Course Objectives:** The objective of this course is to familiarize learners with different development frameworks. The course also introduces students to the principles and process of software engineering.

**Course Outcomes:** On successful completion of this course, student should be able to:

1. Select appropriate frameworks for application development.
2. Apply software engineering principles for application development.
3. Work effectively as a member of team.

**Detailed Syllabus: (Unit wise)**

Unit	Description	Duration
1	<p><b>Introduction to Software Engineering and Process Model:</b> Introduction to Software Engineering, Process framework, Software Development Life Cycle (SDLC), Process Models: Sequential, Incremental and Evolutionary models, Software Requirements - Functional and Non-Functional requirements, Software Requirements Specification (SRS)</p> <p><b>Introduction to Frameworks:</b> Definition and characteristics of frameworks, Historical background and evolution of frameworks, Types of frameworks (e.g., web frameworks, application frameworks, testing frameworks).</p>	08
2	<p><b>Fundamentals of Agile Process:</b> Concept of agility, Need of Agile software development, Agile Manifesto and Principles, Stakeholders and Challenges, Overview of Agile Development Models: Scrum, Extreme Programming, Feature Driven Development, Crystal, Kanban, and Lean Software Development, Methods, Values, Roles, Artifacts, Stakeholders, and challenges. Business benefits of software agility, ASD, DSD.</p> <p><b>Introduction to Scrum:</b> Agile Scrum Framework, Scrum Artifacts, Meetings, Activities and Roles, Scrum Team Simulation, Scrum Planning Principles, Product and Release Planning, sprinting: Planning, Execution, Review and Retrospective; User story definition and Characteristics, Acceptance tests and Verifying stories, Burn down chart, Daily scrum, Scrum Case Study.</p>	12
3	<p><b>Introduction to Architectures:</b> Introduction to Model View Controller (MVC) Framework: History of MVC, Features of MVC, MVC Architecture, MVC Examples, Popular MVC Frameworks, Advantages and Drawbacks of MVC, 3-Tier Architecture Vs MVC Architecture.</p> <p><b>The Reactive Manifesto:</b> Introduction, Reactive Principles, Reactive Systems vs Reactive Programming</p> <p><b>Clean architecture:</b> Introduction, The Dependency Rule, A Typical Scenario.</p>	10
4	<p><b>SOLID Design principles:</b> Introduction, The Single Responsibility Principle, The Open- Closed Principle, The Liskov Substitution Principle, The Interface Segregation Principle, The Dependency Inversion Principle.</p> <p><b>Reactive architecture:</b> Introduction, Design Principles of Reactive Systems, commands and Events, Commands, Events, Messages, Commands Versus Events: An Example Destinations and Space Decoupling, Time Decoupling, The Role of Nonblocking Input/Output, Blocking Network I/O, Threads, and Concurrency, How</p>	10

	Does Nonblocking I/O Work? Reactor Pattern and Event Loop, Anatomy of Reactive Applications.	
5	<b>Core Technologies of Spring Framework:</b> Introduction to Object oriented programming concept, Spring–Environment Setup, Spring beans and its scopes, Spring bean lifecycle, how to create a bean using Factory Bean? How to create a bean using static Factory Bean? Best Practices of spring Framework, Spring Dependency Injection and Inversion of Controls, Spring Java Configuration vs XML configuration.	06
6	<b>Spring Event Handling and Aspect Oriented Programming (AOP):</b> Event Handling in Spring, Custom Events in Spring, AOP Concepts, Types of AOP, AOP in Spring, AOP Spring Architecture, Framework Services for AOP, Using @AspectJ-Style Annotations, AspectJ Integration, Spring - Transaction Management, Spring Web MVC Framework, Spring - Logging with Log4J. <b>Spring Boot:</b> Introduction to spring boot, spring boot Build systems, spring boot Code structure, Springs and dependency injection, spring boot Runners, Spring Boot – Application Properties	06

**Books Recommended:**

*Textbooks:*

1. Iuliana Cosmina Rob Harrop Chris Schaefer Clarence Ho,” An In-Depth Guide to the Spring Framework and Its Tools”, 5<sup>th</sup> Edition, Apress, 2017.
2. Roger S Pressman, “Software Engineering: A Practitioner’s Approach”, 8<sup>th</sup> Edition, McGraw-Hill, 2015.
3. Ian Sommerville, “Software Engineering”, 9<sup>th</sup> Edition, Pearson Education, 2011.
4. Clement Escoffier , Ken Finnigan, “Reactive Systems in Java: Resilient, Event-Driven Architecture with Quarkus, 1<sup>st</sup> Edition, O’Reilly Media, 2021
5. Craig Walls. “Spring Boot in Action” 6<sup>th</sup> Edition, Manning, 2016.

*Reference Books:*

1. Ashish Sarin J Sharma, “Getting Started with Spring Framework”, 2<sup>nd</sup> Edition, CreateSpace, 2012
2. Rod Johnson et al,” Professional Java Development with the Spring Framework”, John Wiley & Sons 2005.

Prepared by

Checked by

Head of the Department

Principal

**Course: Advanced Java Laboratory (DJS23ILH1401)**

**Pre-requisite: Structured programming using C, Object Oriented Programming using Java.**

**Course Objectives:** The objective of the course is to introduce and familiarize students with advanced concepts that go beyond Core Java – most importantly the APIs defined in Java 8. Through this course, students will delve into Java Collections, exploring the intricacies of managing and manipulating data structures efficiently. Understand how to apply design patterns to solve common software design problems and improve code quality, reusability, and scalability.

**Course Outcomes:** On successful completion of this course, student should be able to:

1. Develop reusable codes using generics.
2. Use various APIs in Java for efficient application development.
3. Use appropriate design patterns.

<b>Detailed Syllabus: (unit wise)</b>		
<b>Unit</b>	<b>Description</b>	<b>Duration</b>
<b>1</b>	<b>Java Collections:</b> Collections in Java, basic data structures, arrays and lists, stacks, and queues, sets and maps. <b>Generics:</b> Basic generics, bounded type parameters, type inference, wildcards, type erasure	<b>08</b>
<b>2</b>	<b>Java Reflection API:</b> Modifiers and Security, Accessing Fields, Accessing Methods, Accessing Constructors, What About Arrays? Accessing Generic Type Information, Accessing Annotation Data, Dynamic Interface Adapters	<b>08</b>
<b>3</b>	<b>Lambda Expression:</b> Lambda expression fundamentals, Functional Interfaces, examples on Lambda Expressions, Block Lambda Expressions, Generic Functional Interfaces, Passing Lambda Expression as Arguments, Lambda Expression and Exceptions, Lambda Expression and variable Capture, Method References to static methods, Method references to Instance methods, Method references with Generics, Constructor References, Predefined Functional Interfaces, Comparing method references with lambda expressions.	<b>12</b>
<b>4</b>	<b>The Stream API:</b> Stream Basics, Stream Interfaces, How to Obtain a Stream, A Simple Stream Example, Reduction Operations, Using Parallel Streams, Mapping, Collecting, Iterators and Streams, Use an Iterator with a Stream, Use Spliterator	<b>06</b>
<b>5</b>	<b>Annotations:</b> Annotations Basics, specifying a retention policy, Obtaining Annotations at run time by use of Reflections, The AnnotatedElement Interface, using default values, Marker Annotations, Single member Annotations, The Built in Annotations, Type Annotations, Repeating Annotations, some Restrictions.  Comparable and Comparator, Optional Class: Date/Time API: Date, Calendar, GregorianCalendar, TimeZone, SimpleTimeZone, Locale	<b>12</b>
<b>6</b>	<b>Introduction to Design Patterns</b> Creational: Singleton Pattern, Structural: Adapter Pattern, Behavioural: Observer Pattern	<b>06</b>

**List of Practical's:**

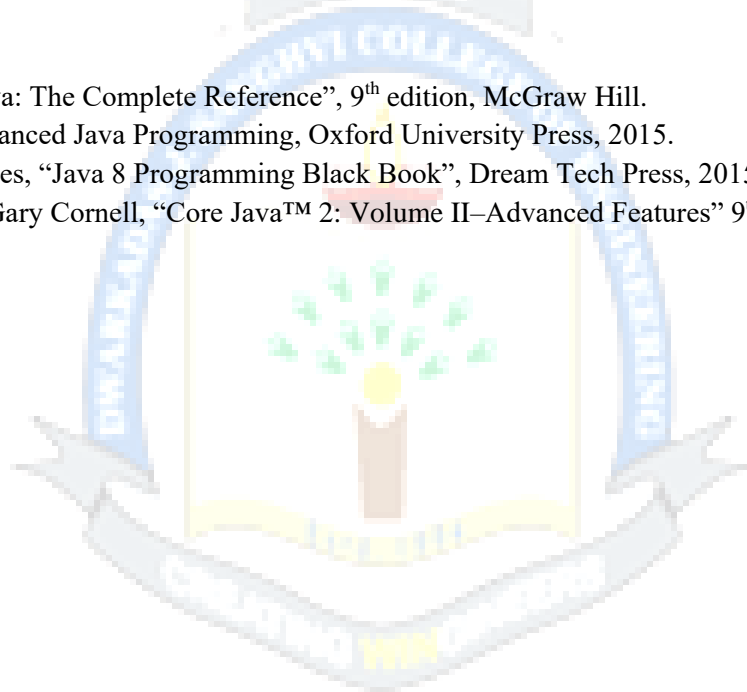
1. Creating JDBC application
2. Implementation of different collection types (stacks, queues, vectors etc)
3. Creation of generic classes, methods
4. Use reflection API to examine or modify the behaviour of methods, classes, and interfaces at runtime.
5. Using streams API to implement program logic by composing functions and executing them in a data flow.
6. Demonstration of lambda expressions.
7. Implementation of Functional Interfaces, Comparable and Comparator.
8. Implementation of Optional Class, Date/Time API.
9. Implementation of Annotations.
10. Implementation of Singleton Design Patterns.
11. Implementation of Structural Design Patterns.
12. Implementation of Behavioral Design Patterns.

**Books Recommended:***Textbooks:*

1. Anita Seth, B.L. Juneja, "JAVA: ONE STEP AHEAD", 1<sup>st</sup> Edition, Oxford University Press; (20 May 2017)
2. Patrick Niemeyer, Daniel Leuck, "Learning Java", 4<sup>th</sup> Edition, O'Reilly Media, Inc, June 2013

*Reference Books:*

1. Herbert Schildt, "Java: The Complete Reference", 9<sup>th</sup> edition, McGraw Hill.
2. Uttam K. Roy, "Advanced Java Programming, Oxford University Press, 2015.
3. D.T. Editorial Services, "Java 8 Programming Black Book", Dream Tech Press, 2015.
4. Cay S. Horstmann, Gary Cornell, "Core Java™ 2: Volume II–Advanced Features" 9<sup>th</sup> Edition, Prentice Hall PTR.



Prepared by

Checked by

Head of the Department

Principal