Shirpur Education Society's

R. C. Patel Institute of Technology, Shirpur

(An Autonomous Institute)

Course Structure and Syllabus

Honors Program
in
DevOps (Development and Operations)

Information Technology

With effect from Year 2025-26

Shahada Road, Near Nimzari Naka, Shirpur, Maharashtra 425405
Ph: 02563 259 802, Web: www.rcpit.ac.in

Honors Program in DevOps (Development and Operations) offered by Information Technology (w.e.f. 2025-26)

Sr. No.	Course Category	Course Code	Course Title	Teaching Scheme			Evaluation Scheme					Total	Credit	
							Continuous Assessment (CA)							
							TA	Term Test 1 (TT1)	Term Test 2 (TT2)	Average of (TT1 & TT2)	ESE			
				L	T	P	[A]			[B]	[C]	[A+B+C]		
Sem-III														
1	H1	RCP23IH1201	Development Frameworks	4			25	15	15	15	60	100	4	4
Sem-IV														
2	H1	RCP23IH1251L	Advanced Java Laboratory			4	25				25	50	2	2

Prepared by:

Ms. J. S. Sonawane

Checked by:

Ms. P. D. Saraf

Prof. Dr. D. R. Patil

BOS Chairman

Prof. S. P. Shukla

C.O.E.

Prof. Dr. P. J. Deore

Dean Academics & Dy. Director

Prof. Dr. J. B. Patil

Director

INFORMATION TECHNOLOGY DEPARTMENT
Office Seal
SHIRPUR

Semester - III

Program: Information Technology	S.Y. B.Tech.	Semester: III
Development Frameworks (RCP23IH1201)		

Prerequisite: : Knowledge of any programming language and Database Management System.

Course Objective(s): 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:

COs	Course Outcomes	Blooms Level	Blooms Description
CO1	Select appropriate frameworks for application development.	L2	Understand
CO2	Apply software engineering principles for application development.	L3	Apply
CO3	Work effectively as a member of team.	L3	Apply

Development Frameworks (RCP23IH1201) Course Contents

Unit-I Introduction to Software Engineering and Process Model: 08 Hrs.

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) Introduction to Frameworks: Definition and characteristics of frameworks, Historical background and evolution of frameworks, Types of frameworks (e.g., web frameworks, application frameworks, testing frameworks).

Unit-II Fundamentals of Agile Process: 12 Hrs.

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.

Introduction to Scrum: 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.

Unit-III Introduction to Architectures: 10 Hrs.

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.

The Reactive Manifesto: Introduction, Reactive Principles, Reactive Systems vs Reactive Programming

Clean architecture: Introduction, The Dependency Rule, A Typical Scenario.

Unit-IV SOLID Design principles: 10 Hrs.

Introduction, The Single Responsibility Principle, The Open- Closed Principle, The Liskov Substitution Principle, The Interface Segregation Principle, The Dependency Inversion Principle.

Reactive architecture: 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 Does Nonblocking I/O Work? Reactor Pattern and Event Loop, Anatomy of Reactive Applications.

Unit-V Core Technologies of Spring Framework: 06 Hrs.

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.

Unit-VI Spring Event Handling and Aspect Oriented Programming (AOP): 06 Hrs.

Event Handling in Spring, Custom Events in Spring, AOP Concepts, Types of AOP, AOP in Spring, AOP Spring Architecture, Framework Services for AOP, Using @AspectJStyle Annotations, AspectJ Integration, Spring - Transaction Management, Spring Web MVC Framework, Spring - Logging with Log4J. Spring Boot: Introduction to spring boot, spring boot Build systems, spring boot Code structure, Springs and dependency injection, spring boot Runners, Spring Boot – Application Properties

Text Books:

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

Reference Books:

1. Ashish Sarin J. Sharma, "Getting Started with Spring Framework", 2nd Edition, CreateSpace, 2012.
2. Rod Johnson et. al., "Professional Java Development with the Spring Framework", John Wiley Sons, 2005.

Program: Information Technology	S.Y. B.Tech.	Semester: IV
Advanced Java Laboratory (RCP23IH1251L)		

Prerequisite: Structured programming using C, Object Oriented Programming using Java.

Course Objective(s): 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:

COs	Course Outcomes	Blooms Level	Blooms Description
CO1	Develop reusable codes using generics.	L3	Apply
CO2	Use various APIs in Java for efficient application development.	L3	Apply
CO3	Use appropriate design patterns.	L3	Apply

Advanced Java Laboratory (RCP23IH1251L)
Course Contents

Unit-I Java Collections — 08 Hrs.

Collections in Java, basic data structures, arrays and lists, stacks, and queues, sets and maps.

Generics: Basic generics, bounded type parameters, type inference, wildcards, type erasure.

Unit-II Java Reflection API — 08 Hrs.

Modifiers and Security, Accessing Fields, Accessing Methods, Accessing Constructors, What About Arrays? Accessing Generic Type Information, Accessing Annotation Data, Dynamic Interface Adapters.

Unit-III Lambda Expression — 12 Hrs.

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.

Unit-IV The Stream API — 06 Hrs.

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.

Unit-V Annotations — 12 Hrs.

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, GregorianCalender, TimeZone, SimpleTimeZone, Locale.

Unit-VI Introduction to Design Patterns — 06 Hrs.

Creational: Singleton Pattern, Structural: Adapter Pattern, Behavioural: Observer Pattern.

List of Laboratory Experiments(At Least 10)

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.

Text Books:

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

Reference Books:

1. Herbert Schildt, "Java: The Complete Reference", 9th edition, Aug 2014, 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" 9th Edition, Prentice Hall, PTR.