GUJARAT TECHNOLOGICAL UNIVERSITY (GTU)

Competency-focused Outcome-based Green Curriculum-2021 (COGC-2021)
Semester-V

Course Title: Advanced Java Programming

(Course Code: 4XXXXXX)

Diploma programmer in which this course is offered	Semester in which offered
Information and Communication Technology	5th Semester

1. RATIONALE

This course is designed to teach advanced Java programming concepts and techniques, building upon the foundation established in the Java Programming course. The course covers a wide range of topics, including advanced language features, user interface development, data persistence and access, server-side development, and modern Java frameworks. It also introduces students to Java for IoT applications, enabling them to develop solutions for the growing field of connected devices.

The course aims to equip students with the skills and knowledge required to develop robust, scalable, and secure Java applications using industry-standard frameworks and best practices. By mastering these advanced concepts, students will be well-prepared to tackle complex real-world problems and develop enterprise-level applications.

The course emphasizes hands-on learning through practical exercises and mini-projects, allowing students to apply their knowledge and gain practical experience in developing advanced Java applications. Upon completion of this course, students will have a strong foundation in advanced Java programming, making them competitive in the job market and enabling them to pursue careers in software development, web development, and IoT domains.

2. COMPETENCY

The aim of this course is to help the students attain the following industry-identified competency through various teaching-learning experiences:

- **Develop advanced Java applications using modern frameworks and best practices.**

3. COURSE OUTCOMES (COs)

The practical exercises, the underpinning knowledge, and the relevant soft skills associated with this competency are to be developed in the student to display the following COs:

a) Utilize advanced Java language features and develop user interfaces using Java Swing and JavaFX frameworks.
b) Implement data persistence and access using JDBC, Hibernate, and JPA.
c) Create server-side applications using Java Servlets, JSP, and RESTful web services.
d) Apply modern Java frameworks such as Spring and Spring Boot for rapid application development.
e) Develop Java applications for IoT using Arduino, Raspberry Pi, and IoT protocols.

4. TEACHING AND EXAMINATION SCHEME

Teaching Scheme (In Hours)			Total Credits (L+T+P/2)	Examination Scheme				
				Theory Marks		Practical Marks		Total Marks
L	T	P	C	CA	ESE	CA	ESE	
3	0	2	4	30	70	25	25	150

(*): Out of 30 marks under the theory CA, 10 marks are for assessment of the micro-project to facilitate integration of COs and the remaining 20 marks is the average of 2 tests to be taken during the semester for the assessing the attainment of the cognitive domain UOs required for the attainment of the COs.

***Legends**: **L**-Lecture; **T** – Tutorial/Teacher Guided Theory Practice; **P** -Practical; **C** – Credit, **CA** - Continuous Assessment; **ESE** -End Semester Examination.*

5. SUGGESTED PRACTICAL EXERCISES:

The following practical outcomes (PrOs) are the subcomponents of the COs. These PrOs need to be attained to achieve the COs.

Sr. No.	Practical Outcomes (PrOs)	Unit No.	Approx. Hrs. required
1	Write a Java program using lambda expressions to sort a list of strings.	I	1
2	Implement a generic class for a stack data structure.	I	1
3	Use the Java Reflection API to dynamically invoke methods of a class.	I	1
4	Write a Java program using Streams API to filter and process a large dataset.	I	2
5	Create a Swing application with a custom layout and event handling.	II	2
6	Develop a JavaFX application using FXML and CSS styling.	II	3
7	Implement a JDBC program to perform CRUD operations on a database	III	2
8	Create a Hibernate application to map entities and relationships.	III	2
9	Develop a JPA application to perform database operations using JPQL	III	2
10	Optimize a database query using indexing and query plan analysis.	III	1

11	Implement a Java Servlet to handle form submissions.	IV	1
12	Create a JSP application with Expression Language (EL) and JSTL.	IV	2
13	Develop a RESTful web service using JAX-RS.	IV	2
14	Create a Spring MVC application with form validation and data binding.	V	2
15	Implement a Spring Boot application with RESTful endpoints.	V	2
16	Develop a Java application to interface with Arduino using serial communication.	V	1
17	Create a Java application to control Raspberry Pi GPIO pins.	V	1
Total			**28**

Note

i. *More **Practical Exercises** can be designed and offered by the respective course teacher to develop the industry relevant skills/outcomes to match the COs. The above table is only a suggestive list.*

ii. *The following are some **sample** 'Process' and 'Product' related skills (more may be added/deleted depending on the course) that occur in the above listed **Practical Exercises** of this course required which are embedded in the COs and ultimately the competency..*

Sr. No.	Performance Indicators for the PrOs	Weightage in %
1	Use of appropriate technology/software/tools.	10
2	Coding methodology.	30
3	Testing and Debugging of the program.	20
4	Correctness of Program.	20
5	Submission in time.	20
Total		**100**

6. MAJOR EQUIPMENT/ INSTRUMENTS REQUIRED

These major equipment/instruments and software required to develop PrOs are given below with broad specifications to facilitate procurement of them by the administrators/management of the institutes. This will ensure the conduction of practical in all institutions across the state in a proper way so that the desired skills are developed in students.

Sr. No.	Equipment Name with Broad Specifications	PrO. No.
1	Computer with the latest configuration with Windows/Linux/Unix Operating System.	All
2	JDK (Java Development Kit) Version 8 or above	All
3	Any IDE - Eclipse, IntelliJ IDEA, VS Code	All
4	Apache Tomcat Server	11, 12, 13

Sr. No.	Equipment Name with Broad Specifications	PrO. No.
5	MySQL or any other relational database	7, 8, 9, 10
6	Arduino UNO board and sensors	16
7	Raspberry Pi board	17

7. AFFECTIVE DOMAIN OUTCOMES

The following **sample** Affective Domain Outcomes (ADOs) are embedded in many of the above-mentioned COs and PrOs. More could be added to fulfill the development of this competency.

a) Follow best practices and coding standards while developing Java applications.
b) Work effectively in a team to develop complex Java applications.
c) Demonstrate a willingness to learn and adapt to new Java frameworks and technologies.
d) Develop an appreciation for the importance of testing and debugging in software development.
e) Recognize the significance of performance optimization in Java applications.

The ADOs are best developed through the laboratory/field-based exercises. Moreover, the level of achievement of the ADOs according to Krathwohl's 'Affective Domain Taxonomy' should gradually increase as planned below:

i. 'Valuing Level' in the 1st year
ii. 'Organization Level' in the 2nd year.
iii. 'Characterization Level' in the 3rd year.

8. UNDERPINNING THEORY:

The major underpinning theory is formulated as given below, and only higher-level UOs of Revised Bloom's taxonomy are mentioned for the development of the COs and competency in the students by the teachers. If required, more such higher-level UOs could be included by the course teacher to focus on the attainment of COs and competency.

Unit	Unit Outcomes (UOs)	Topics and Sub-topics
Unit I Advanced Java Language Features	1a. Apply advanced Java language features effectively. 1b. Utilize generics and type inference for type safety. 1c. Employ reflection	1.1 Lambda expressions and functional interfaces 1.1.1. Introduction to lambda expressions 1.1.2. Functional interfaces and their usage 1.1.3. Method references and their applications 1.2. Java Generics and Type Inference

	and annotations for runtime processing. 1d. Use Streams API for efficient data processing. 1e. Apply debugging, logging, and testing techniques.	1.2.1. Understanding generics and their benefits 1.2.2. Type inference and diamond operator 1.2.3. Bounded type parameters and wildcards 1.2.4. Generic methods and classes 1.3. Java Reflection API and Annotations 1.3.1. Reflection API basics and use cases 1.3.2. Annotations and their applications 1.3.3. Runtime annotation processing 1.3.4. Dynamic class loading and object creation 1.4. Streams API and parallel processing 1.4.1. Introduction to Streams API 1.4.2. Intermediate and terminal operations 1.4.3. Parallel processing with Streams 1.5. Java Debugging, Logging, and Testing 1.5.1. Debugging techniques and tools 1.5.2. Logging frameworks (e.g., Log4j, SLF4J) 1.5.3. Introduction to unit testing with JUnit 1.5.4. Test-driven development (TDD) principles
Unit II User Interface Development	2a. Compare and contrast Java UI technologies. 2b. Develop user interfaces using Java Swing. 2c. Create rich user interfaces with JavaFX and FXML. 2d. Implement event handling and UI design patterns.	2.1. Introduction to Java UI Technologies 2.1.1. Overview of JFC, Applet, AWT, and Swing 2.1.2. Limitations and alternatives 2.1.3. Differences between AWT and Swing 2.2. Java Swing 2.2.1. Swing components and containers 2.2.2. Layout managers and custom layouts 2.2.3. Swing event handling

		2.2.4. Swing threading and concurrency 2.3. JavaFX and FXML 2.3.1. Introduction to JavaFX architecture 2.3.2. Creating user interfaces with FXML 2.3.3. Styling with CSS 2.3.4. JavaFX charts, graphs, 3D, and animations 2.3.5. Using Scene Builder for visual design 2.4. Event Handling and UI Design Patterns 2.4.1. Event-driven programming concepts 2.4.2. UI design patterns (e.g., MVC, MVP, MVVM) 2.4.3. Implementing design patterns in JavaFX
Unit III Data Persistence and Access	3a. Design and implement database applications using JDBC. 3b. Utilize ORM with Hibernate for data persistence. 3c. Apply JPA for object-relational mapping. 3d. Optimize database queries for performance.	3.1. Database Design and JDBC 3.1.1. Two-Tier and Three-Tier Database Design 3.1.2. JDBC basics and database connectivity 3.1.3. Connection pooling concepts and implementation 3.1.4. Prepared statements and result set handling 3.1.5. Transaction management with JDBC 3.2. Object-Relational Mapping (ORM) with Hibernate 3.2.1. Introduction to ORM and Hibernate 3.2.2. Mapping entities and relationships 3.2.3. Hibernate query language (HQL) and criteria API 3.2.4. Caching and performance optimization with Hibernate 3.3. Java Persistence API (JPA) 3.3.1. JPA concepts and entity lifecycle 3.3.2. Entity relationships and inheritance

		3.3.3. JPA query language (JPQL) and native queries 3.3.4. JPA caching and performance tuning 3.4. Query Optimization and Performance Tuning 3.4.1. Optimization techniques for JDBC, Hibernate, and JPA 3.4.2. Identifying and resolving performance bottlenecks 3.4.3. Indexing and query plan analysis 3.4.4. Caching strategies and cache invalidation
Unit IV Server-Side Development	4a. Develop server-side applications using Java Servlets. 4b. Create dynamic web pages using JavaServer Pages (JSP). 4c. Integrate servlets and JSP for web application development. 4d. Implement RESTful web services using Java.	4.1. Java Servlets and Filters 4.1.1. Servlet lifecycle and API 4.1.2. Creating and configuring servlets 4.1.3. Implementing filters for request processing 4.1.4. Servlet security and authentication 4.2. JavaServer Pages (JSP) 4.2.1. JSP syntax and directives 4.2.2. Expression Language (EL) and JSTL 4.2.3. MVC pattern with JSP and servlets 4.2.4. JSP best practices and performance optimization 4.3. Integration of Servlets and JSP 4.3.1. Combining servlets and JSP for dynamic web pages 4.3.2. Handling form submissions and validations 4.3.3. Session management and state maintenance 4.4. RESTful Web Services 4.4.1. Introduction to REST principles and architecture 4.4.2. Designing RESTful APIs 4.4.3. Implementing RESTful endpoints with JAX-RS

		4.4.4. Consuming RESTful services with Java clients
Unit V Modern Java Frameworks & Java for IoT	5a. Apply Spring Framework for developing Java applications. 5b. Utilize Spring Boot for rapid application development. 5c. Develop Java applications for IoT devices.	5.1. Spring Framework (Core, MVC, Security, Data) 5.1.1. Spring Core concepts (DI, IoC, AOP, POJO) 5.1.2. Spring MVC for web application development 5.1.3. Spring Security for authentication and authorization 5.1.4. Spring Data for data access and persistence 5.2. Spring Boot for Rapid Application Development 5.2.1. Introduction to Spring Boot 5.2.2. Auto-configuration and starter dependencies 5.2.3. Creating RESTful APIs with Spring Boot 5.2.4. Spring Boot Actuator and monitoring 5.3. Java for IoT (2 hours lecture, 2 hours lab) 5.3.1. Overview of IoT concepts and protocols 5.3.2. Interfacing with Arduino and NodeMCU using Java 5.3.3. Raspberry Pi programming with Pi4J library 5.3.4. IoT protocols (MQTT, CoAP) with Java 5.3.5. Security considerations for IoT applications

Note: *The UOs need to be formulated at the 'Application Level' and above of Revised Bloom's Taxonomy' to accelerate the attainment of the COs and the competency.*

9. SUGGESTED SPECIFICATION TABLE FOR QUESTIONPAPER DESIGN:

Unit No.	**Unit Title**	**Teaching hours**	**Distribution of theory marks**			
			R	**U**	**A**	**Total**
I	Advanced Java Language Features	8	3	4	5	12
II	User Interface Development	9	3	5	8	16
III	Data Persistence and Access	8	3	5	8	16
IV	Server-Side Development	9	3	5	8	16
V	Modern Java Frameworks & Java for IoT	8	2	3	5	10
Total		**42**	**14**	**22**	**34**	**70**

***Legends**: **R**=Remember, **U**=Understand, **A**=Apply and above (Revised Bloom's Taxonomy)*

Note: *This specification table provides general guidelines to assist students in their learning and teachers in teaching and question paper designers/setters to formulate test items/questions to assess the attainment of the UOs. The actual distribution of marks at different taxonomy levels (of R, U, and A) in the question paper may vary from the above table.*

10. SUGGESTED STUDENT ACTIVITIES

Other than the classroom and laboratory learning, the following are the suggested student-related co-curricular activities which can be undertaken to accelerate the attainment of the various outcomes in this course: Students should conduct the following activities in groups and prepare small reports of 1 to 5 pages for each activity. For the micro-project report, students should follow the suggested format. For other activities, students and teachers together can decide on the format of the report. Students should also collect/record physical evidence such as photographs/videos of the activities for their (student's) portfolio, which will be useful for their placement interviews:

- Undertake micro-projects in teams.
- Develop a complete web application using Spring MVC and Hibernate.
- Create a mobile application using Java and Android SDK.
- Participate in coding competitions and hackathons related to Java programming.
- Contribute to open-source Java projects on GitHub.
- Attend seminars, workshops, and tech talks on advanced Java topics.
- Prepare a comparative study of different Java web frameworks.
- Develop a IoT project using Java and Raspberry Pi.

11. SUGGESTED SPECIAL INSTRUCTIONAL STRATEGIES (if any)

These are sample strategies, which the teacher can use to accelerate the attainment of the various outcomes in this course:

- Massive open online courses (MOOCs) may be used to teach various topics/sub-topics.
- Guide student(s) in undertaking micro-projects.
- Encourage students to contribute to open-source Java projects.
- Ask students to study research papers on advanced Java topics and present their findings.
- Invite industry experts to deliver guest lectures on latest trends in Java development.
- Organize coding competitions and hackathons to foster a competitive learning environment.
- Encourage students to participate in Java user groups and online forums.
- Use flipped classroom approach for selected topics to promote self-learning.

12. SUGGESTED MICRO-PROJECTS

Only one micro-project is planned to be undertaken by a student that needs to be assigned to him/her in the beginning of the semester. In the first four semesters, the micro-project are group-based (group of 3 to 5). However, in the fifth and sixth semesters, the number of students in the group should not exceed three.

The micro-project could be industry application based, internet-based, workshop-based, laboratory-based or field-based. Each micro-project should encompass two or more COs which are in fact, an integration of PrOs, UOs and ADOs. Each student will have to maintain dated work diary consisting of individual contribution in the project work and give a seminar presentation of it before submission. The total work load on each student due to the micro-project should be about 16 (sixteen) student engagement hours (i.e., about one hour per week) during the course. The students ought to submit micro-project by the end of the semester (so that they develop the industry-oriented COs).

A suggestive list of micro-projects is given here. This should relate highly with competency of the course and the COs. Similar micro-projects could be added by the concerned course teacher:

- Develop a web-based employee management system using Spring MVC, Hibernate, and MySQL.
- Create an e-commerce application with shopping cart functionality using Java Servlets, JSP, and JDBC.
- Implement a chat application using Java Sockets and Swing for the user interface.
- Develop a weather monitoring system using Java, Raspberry Pi, and sensors.
- Create a task management application using JavaFX and JPA for data persistence.
- Develop Degree/Diploma Admission web application using java web application technologies to get student registration data, process and filter data.
- Develop MVC based web application for college placement record keeping.
- Student Record Keeping System
- Library management system
- Electricity billing system
- Quiz management system
- Online Voting System

13. SUGGESTED LEARNING RESOURCES

Sr. No.	Title of Book	Author	Publication with place, year and ISBN
1	Effective Java	Joshua Bloch	Addison-Wesley, 2018, ISBN: 9780134685991
2	Spring in Action	Craig Walls	Manning Publications, 2018, ISBN: 9781617294945
3	Hibernate in Action	Christian Bauer, Gavin King	Manning Publications, 2015,

			ISBN: 9781932394887
4	Java Persistence with Hibernate	Christian Bauer, Gavin King	Manning Publications,2016, ISBN: 9781617292286
5	JavaFX in Action	Simon Ritter, Johan Vos	Manning Publications, 2022, ISBN: 9781617299131

14. SOFTWARE/LEARNING WEBSITES

- https://docs.oracle.com/javase/tutorial/ (Java Documentation)
- https://www.baeldung.com/ (Java Tutorials and Articles)
- https://www.javatpoint.com/advanced-java-tutorial (Advanced Java Tutorial)
- https://www.tutorialspoint.com/spring_boot/index.htm (Spring Boot Tutorial)
- https://www.javatpoint.com/hibernate-tutorial (Hibernate Tutorial)
- https://www.javacodegeeks.com/ (Java Code Geeks)
- https://www.journaldev.com/java-tutorial-java-ee-tutorials (Java and Java EE Tutorials)
- https://www.mkyong.com/ (Java Tutorials and Examples)
- https://www.javaguides.net/ (Java Guides)
- https://www.udemy.com/topic/java/ (Java Courses on Udemy)
- https://www.javatpoint.com/java-ee
- https://www.geeksforgeeks.org/introduction-to-java-swing/
- https://docs.jboss.org/hibernate/orm/3.6/reference/en-US/pdf/hibernate_reference.pdf
- https://www.javatpoint.com/what-is-advance-java
- https://docs.spring.io/spring-framework/docs/6.0.7/reference/pdf/spring-framework.pdf
- https://docs.oracle.com/cd/E13222_01/wls/docs81/jsp/intro.html
- https://www.javatpoint.com/servlet-tutorial
- https://docs.oracle.com/javase/tutorial/jdbc/
- https://www.oracle.com/java/technologies/jspt.html
- https://www.youtube.com/playlist?list=PLbRMhDVUMngcx5xHChJ-f7ofxZI4JzuQR

15. PO-COMPETENCY-CO MAPPING:

Semester V	**Advanced Java Programming (Course Code: 4350304)**						
	POs						
Competency & Course Outcomes	PO 1 Basic & Discipline specific knowledge	PO 2 Problem Analysis	PO 3 Design/ development of solutions	PO 4 Engineering Tools, Experimentation &Testing	PO 5 Engineering practices for society, sustainability & environment	PO 6 Project Management	PO 7 Life-long learning
Competency	**Develop advanced Java applications using modern frameworks**						

	and best practices.						
Course Outcomes CO1: Utilize advanced Java language features and develop user interfaces using Java Swing and JavaFX frameworks.	3	2	3	2	-	2	2
CO2: Implement data persistence and access using JDBC, Hibernate, and JPA.	3	3	3	2	-	2	2
CO3: Create server-side applications using Java Servlets, JSP, and RESTful web services.	3	3	3	2	-	2	2
CO4: Apply modern Java frameworks such as Spring and Spring Boot for rapid application development.	3	3	3	2	-	2	3
CO5: Develop Java applications for IoT using Arduino, Raspberry Pi, and IoT protocols.	3	3	3	3	1	2	3

*Legend: '**3**' for high, '**2**' for medium, **'1'** for low and **'-'** for no correlation of each CO with PO.*

16. COURSE CURRICULUM DEVELOPMENT COMMITTEE

GTU Resource Persons

Sr. No.	**Name and Designation**	**Institute**	**Contact No.**	**Email**
1	Shri M. J. Dabgar - Lecturer	Government Polytechnic, Palanpur	8128576285	milav.dabgar@gujgov.edu.in