

Government Polytechnic, Balangir							
Department of Computer Science Engineering							
LESSON PLAN 2026-27(WINTER)							
NAME OF THE TEACHER : DEEPAK KUMAR BARDA, LECT.CSE(STAGE-II)							
Subject: PROGRAMMING WITH JAVA (Course Code: CSEPE 301A/TH3(A))							
Program: Diploma in Computer Science and Engineering							
Semester: 5th							
Total Contact Hours: 45							
Total Marks: 100							
Assessment: Internal Assessment – 30, End Term – 70							
To keep the table concise, the Course Outcomes are referenced by number: CO1: Explain the history, evolution, and core concepts of Java. CO2: Use datatypes, variables, arrays, operators, and control statements effectively. CO3: Implement classes, methods, inheritance, packages, and interfaces. CO4: Implement exception handling to ensure robust and error-free programs. CO5: Create and manage multiple threads for concurrent programming.							
Lesson	Unit	Topic PDF	Learning Objective	Activity	Course Outcome	Learning Methodology	Homework
1	Unit I: Intro to Java	History and features of Java	Understand Java's origins and key features.	Feature matching quiz.	CO1	Lecture & Discussion	Read Java history.
2	Unit I: Intro to Java	Setting up JDK & JVM	Install JDK and understand JVM architecture.	System configuration check.	CO1	Live Demonstration	Install JDK on personal PC.
3	Unit I: Intro to Java	Basic Java syntax: data types, variables	Declare and initialize variables using proper syntax.	Code tracing exercise.	CO2	Interactive Lecture	Write variable declarations.
4	Unit I: Intro to Java	Operators, control flow (if-else, switch, loops)	Apply control structures to solve basic logic problems.	Whiteboard logic mapping.	CO2	Lecture & Practice	Create a basic calculator script.
5	Unit I: Intro to Java	Input/Output (Scanner and System.out)	Read user input and display formatted output.	Pair programming I/O.	CO2	Live Coding	Build an interactive console form.
6	Unit I: Intro to Java	Writing, compiling, executing; Create Packaged class	Write and compile a complete packaged Java program.	First program execution.	CO1, CO2	Hands-on Lab	Write a "Hello World" packaged class.
7	Unit II: Intro to OOP	Basic concepts: objects, classes, encapsulation, inheritance, polymorphism	Define core OOP paradigms.	Real-world object modeling.	CO3	Lecture & Brainstorming	Identify 5 real-world objects & attributes.
8	Unit II: Intro to OOP	Fundamentals of Class & Object, New Keyword, reference variable	Instantiate objects using the 'new' keyword.	Object creation drill.	CO3	Lecture & Live Demo	Write a basic Student class.
9	Unit II: Intro to OOP	Members methods of a class, Access modifiers	Define methods and apply public, private, protected, default modifiers.	Access modifier debugging.	CO3	Interactive Lecture	Create methods with different modifiers.

10	Unit II: Intro to OOP	Encapsulation: Hiding data through access modifiers, Access Control	Protect class data using getter and setter methods.	Encapsulation refactoring.	CO3	Case Study	Secure variables in the Student class.
11	Unit II: Intro to OOP	Static Methods, Static Variables	Differentiate between instance and static members.	Memory state visualization.	CO3	Lecture & Demonstration	Create a class counter using static variables.
12	Unit II: Intro to OOP	Static Block	Initialize static variables using static blocks.	Code output prediction.	CO3	Live Coding	Write a class with a static initialization block.
13	Unit II: Intro to OOP	Constructor	Initialize objects effectively upon creation.	Constructor creation drill.	CO3	Hands-on Practice	Add a constructor to the Student class.
14	Unit II: Intro to OOP	Overloading methods, Overloading constructors	Implement compile-time polymorphism via overloading.	Overloading challenge.	CO3	Lecture & Lab	Write overloaded methods for area calculation.
15	Unit II: Intro to OOP	Using final keyword, Finalize method	Apply final to constants and understand garbage collection basics.	Keyword application quiz.	CO3	Interactive Lecture	Implement a final constant in a program.
16	Unit III: Inheritance	Basics of Inheritance, members accessibility	Establish parent-child class relationships.	Hierarchy mapping.	CO3	Lecture & Discussion	Design a vehicle inheritance tree.
17	Unit III: Inheritance	Types: Single, multilevel, hierarchical	Implement different structural types of inheritance.	Multi-level coding.	CO3	Guided Practice	Code a multi-level inheritance program.
18	Unit III: Inheritance	Super keyword, Method overriding	Access parent members and achieve runtime polymorphism.	Overriding exercise.	CO3	Live Coding	Override a method from a parent class.
19	Unit III: Inheritance	Final keyword (for classes, methods, and variables)	Prevent modification or extension using final.	Code restriction lab.	CO3	Interactive Lecture	Apply final to prevent overriding.
20	Unit III: Inheritance	Dynamic Method Dispatch, Abstract class	Utilize abstract classes for template design.	Abstract class modeling.	CO3	Lecture & Demonstration	Create an abstract Shape class.
21	Unit III: Inheritance	Preventing Overriding, Preventing Inheritance	Secure classes against unwanted inheritance.	Security code review.	CO3	Case Study	Review and secure a legacy class.
22	Unit IV: Interfaces	Purpose of interface, Defining and implementing interfaces	Establish contracts using Java interfaces.	Interface contract design.	CO3	Lecture & Practice	Define a Playable interface.
23	Unit IV: Interfaces	Interface reference variables	Use interface references to hold implementing objects.	Reference swapping drill.	CO3	Live Demonstration	Instantiate objects via interface references.

24	Unit IV: Interfaces	Interface with variables	Utilize static final variables within interfaces.	Constant utilization.	CO3	Interactive Lecture	Define and use interface constants.
25	Unit IV: Interfaces	Implementing interfaces (multiple)	Achieve multiple inheritance-like behavior.	Multi-interface coding.	CO3	Guided Lab	Implement two interfaces in one class.
26	Unit IV: Interfaces	Extending interfaces	Inherit interfaces from other interfaces.	Interface hierarchy build.	CO3	Hands-on Practice	Extend a base interface and implement it.
27	Unit V: Arrays & Strings	Arrays: declaring, initializing, accessing	Store and traverse single-dimensional data sets.	Array manipulation lab.	CO2	Lecture & Live Coding	Create and print an array of integers.
28	Unit V: Arrays & Strings	Multidimensional arrays	Work with matrix-style data structures.	Matrix addition exercise.	CO2	Guided Practice	Write a 2D array matrix printer.
29	Unit V: Arrays & Strings	String manipulation: creating strings	Instantiate strings using literals and objects.	Memory pool discussion.	CO2	Interactive Lecture	Compare String creation methods.
30	Unit V: Arrays & Strings	String methods (length, concat, substring, etc.)	Process and manipulate text data effectively.	Text parsing drill.	CO2	Hands-on Lab	Extract substrings from a sentence.
31	Unit V: Arrays & Strings	Wrapper classes: Integer, Float, etc.	Convert primitives to objects for collection use.	Autoboxing exercise.	CO2	Lecture & Demonstration	Write a program using Integer wrappers.
32	Unit V: Arrays & Strings	Wrapper classes: Double, parsing basics	Utilize Double and perform primitive parsing.	Parsing user input.	CO2	Guided Practice	Parse command-line args to Double.
33	Unit VI: Exceptions & Files	Exceptions: types of exceptions	Distinguish between checked and unchecked exceptions.	Error identification quiz.	CO4	Lecture & Discussion	Read documentation on common exceptions.
34	Unit VI: Exceptions & Files	try-catch blocks	Trap and handle runtime errors safely.	Break-it-and-fix-it lab.	CO4	Live Coding	Handle an ArithmeticException.
35	Unit VI: Exceptions & Files	finally block, Throwing exceptions	Ensure resource cleanup and manually throw errors.	Exception propagation.	CO4	Interactive Lecture	Implement a finally block.
36	Unit VI: Exceptions & Files	Custom exceptions	Define user-specific exception classes.	Custom error creation.	CO4	Hands-on Practice	Create an InvalidAgeException.
37	Unit VI: Exceptions & Files	File input/output: reading/writing files	Understand Java file I/O operations conceptually.	I/O flow mapping.	CO4	Lecture & Diagramming	Outline standard file I/O steps.
38	Unit VI: Exceptions & Files	Using File Input Stream, File Output Stream	Perform byte-stream file operations.	Byte copying exercise.	CO4	Live Demonstration	Write a program to copy a text file.
39	Unit VI: Exceptions & Files	Buffered Reader, Buffered Writer	Perform efficient character-stream operations.	Text logging lab.	CO4	Guided Practice	Write user inputs to a log file.

40	Unit VII: Multithreading	Basics of threads, Java threaded model	Understand concurrency and the thread lifecycle.	Lifecycle state diagram.	CO5	Lecture & Discussion	Review thread lifecycle states.
41	Unit VII: Multithreading	Defining threads using Runnable interface	Create threads by implementing Runnable.	Runnable creation lab.	CO5	Live Coding	Implement Runnable for a task.
42	Unit VII: Multithreading	Defining threads using Thread superclass	Create threads by extending the Thread class.	Thread extension lab.	CO5	Guided Practice	Extend Thread and start execution.
43	Unit VII: Multithreading	Multiple threads	Execute multiple tasks concurrently.	Race condition simulation.	CO5	Hands-on Practice	Run 3 separate threads simultaneously.
44	Unit VII: Multithreading	Thread Priority values	Assign and manage execution priorities.	Priority testing.	CO5	Interactive Lecture	Set MAX and MIN priorities to threads.
45	Unit VII: Multithreading	Thread Synchronization Using synchronized blocks	Prevent data corruption in multithreaded apps.	Thread safety exercise.	CO5	Live Demonstration & Lab	Synchronize a shared counter resource.

Deepak K M
25.06.2026
Signature of Teacher

Deepak K M
25.06.2026
Signature of HOD