

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING - DATA SCIENCE

COURSE MODULE OF THE SUBJECT TAUGHT FOR THE SESSION 2026-27

(ODD SEM)

Course Syllabus with CO's

Faculty Name: Mrs. Madhu Nagaraj				Academic Year: 2026 - 2027			
Department: CSE (Data Science)							
Course Code	CourseTitle	Core/Elective	Prerequisite	Contact Hours			Total Hrs/ Sessions
				L	T	P	
1BCS302	Object Oriented Programming With Java	Core	Basic programming logic	3	0	2	42:0:28
Module 1							Contact Hours
An Overview of Java: Object-Oriented Programming (Two Paradigms, Abstraction, The Three OOP Principles), Using Blocks of Code, Lexical Issues (White-space, Identifiers, Literals, Comments, Separators, The Java Keywords). Data Types, Variables, and Arrays: The Primitive Types (Integers, Floating-Point Types, Characters, Booleans), Variables, Type Conversion and Casting, Automatic Type Promotion in Expressions, Arrays. Operators: Arithmetic Operators, Relational Operators, Boolean Logical Operators, The Assignment Operator, The ? Operator, Operator Precedence, Using Parentheses. Control Statements: Java’s Selection Statements (if, The Traditional switch), Iteration Statements (while, do-while, for, The For-Each Version of the for Loop, Local Variable Type Inference in a for Loop,Nested Loops), Jump Statements (Using break, Using continue, return). Textbook: Chapters 2, 3, 4, 5							9
Module 2							
Introducing Classes: Class Fundamentals, Declaring Objects, Assigning Object Reference Variables, Introducing Methods, Constructors, the 'this' Keyword, and Garbage Collection. Methods and Classes: Overloading Methods, Using Objects as Parameters, Argument Passing, Returning Objects, Recursion, Introducing Access Control, Understanding static, Introducing final, Introducing Nested and Inner Classes. Textbook: Chapters 6, 7							8
Module-3							
Inheritance: Inheritance Basics, Using super, Creating a Multilevel Hierarchy, When Constructors Are Executed, Method Overriding, Dynamic Method Dispatch, Using Abstract Classes, Using 'final' with Inheritance, Local Variable Type Inference and Inheritance, The Object Class. Packages: Packages, Packages and Member Access, Importing Packages. Interfaces: Interfaces, Default Interface Methods, Use static Methods in an Interface, Private Interface Methods. Textbook: Chapters 8, 9							9
Module - 4							
Exceptions: Exception-Handling Fundamentals, Exception Types, Uncaught Exceptions, Using try and catch, Multiple catch Clauses, Nested try Statements, throw, throws, finally, Java’s Built-in Exceptions, Creating Your Own Exception Subclasses, Chained Exceptions. Multithreaded Programming: The Java Thread Model, The Main Thread, Creating a Thread, Creating Multiple Threads, Using isAlive() and join(), Thread Priorities,Obtaining a Thread’s State. Textbook: Chapters 10, 11							8

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING - DATA SCIENCE

Module - 5	
Enumerations, Type Wrappers and Autoboxing: Enumerations (Enumeration Fundamentals, The values() and valueOf() Methods), Type Wrappers (Character, Boolean, The Numeric Type Wrappers), Autoboxing (Autoboxing and Methods, Autoboxing/Unboxing Occurs in Expressions, Autoboxing/Unboxing Boolean and Character Values). Generics: A Simple Generics Example, A Generic Class with Two Type Parameters, The General Form of a Generic Class. Textbook: Chapters 12, 14	8
PRACTICAL COMPONENT OF IPCC	
1. Develop a Java program to add TWO matrices of suitable order N. 2. Develop a stack class to hold a maximum of 10 integers with suitable methods. Develop a Java main method to illustrate Stack operations. 3. Create a Java class called Student with the USN, Name, Branch, Phone details as variables within it. Write a Java program to create N Student objects and print the USN, Name, Branch, and Phone of these objects with suitable headings. 4. Develop a Java program to create a class named Shape. Create three sub classes, namely: circle, triangle and square. Each class has two member functions named draw and erase. Demonstrate the concepts of polymorphism by developing appropriate methods, defining member data, and implementing the main program. 5. Develop a Java program to create an interface Resizable with methods resizeWidth(int width) and resizeHeight(int height) that allow an object to be resized. Create a class Rectangle that implements the Re-sizable interface and implements the resize methods. 6. Develop a Java program to create a package named mypack and import & implement it in a suitable class. 7. Develop a Java program to raise a custom exception (user-defined exception) for DivisionByZero using try, catch, throw and finally. 8. Develop a program to create a class MyThread with a class constructor, call the base class constructor, using super and start the thread. The run method of the class starts after this. It can be observed that both main thread and created child thread are executed concurrently. 9. Develop a Java program using an enumeration called Department containing CSE, ISE, ECE, and MECH. Accept a department name as input and display the selected department using valueOf(). 10. Develop a Java program to read the marks of three subjects as strings, convert them into wrapper objects, calculate the total and average using Autoboxing/unboxing and display the result. 11. Micro Project: A team of two to three students shall develop a project and submit a brief report for a specific task assigned by the course instructor based on the concepts covered in the syllabus. Note: Each experiment (1 to 10) will be evaluated for 30 marks (total 300), and the micro-project for 120 marks. Total marks (420) shall be scaled down to 15. Make use of the most recent version of the IDE tool for program development.	

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING - DATA SCIENCE

TextBooks:

1. Herbert Schildt and Danny Coward, Java: The Complete Reference, 13th Edition, McGraw-Hill, 2024.

Reference Books

1. Cay S. Horstmann, Core Java Volume I – Fundamentals, 14th Edition, Oracle Press, 2026.
2. E. Balagurusamy, Programming with Java, 7th Edition, McGraw-Hill Education, 2024.

List of URL's

1. Official Java documentation (Oracle): <https://docs.oracle.com/javase/tutorial/>
2. Modern Java learning resources (Dev Java): <https://dev.java/learn/>
3. Java Tutorial: <https://www.w3schools.com/java/>
4. Programming examples and practice (GeeksforGeeks): <https://www.geeksforgeeks.org/java/java/>
5. Infosys Springboard Course on Java Programming Fundamentals
<https://infyspringboard.onwingspan.com/web/en/login>
6. HackerRank platform: <https://www.hackerrank.com/domains/java>
7. Structured video lectures (NPTEL): (<https://nptel.ac.in/courses/106105191/>)

Course outcomes: The students should be able to:

- CO1. Demonstrate proficiency in writing simple programs involving branching and looping structures.
- CO2. Construct object-oriented solutions using classes, objects, and methods.
- CO3. Develop programs using inheritance, packages, and interfaces.
- CO4. Make use of exception handling and multithreading in solving complex problems
- CO5. Apply concepts of autoboxing, enumerations, and generics in program development.
- CO6. Choose a suitable IDE tool to design, develop, and debug Java programs.

Assessment Details (both CIE and SEE)

The assessment for each course is equally divided between Continuous Internal Evaluation (CIE) and the Semester End Examination (SEE), with each component carrying 50% weightage (i.e., 50 marks each).

The CIE Theory component will be 25 marks and CIE Practical component will be 25 marks.

To qualify and become eligible to appear for SEE, in the CIE theory component, a student must score at least 40% of 25 marks, i.e., 10 marks.

To qualify and become eligible to appear for SEE, in the CIE Practical component, a student must secure a minimum of 40% of 25 marks, i.e., 10 marks.

To pass the SEE, a student must secure a minimum of 35% of 50 marks, i.e., 18 marks.

A student is deemed to have successfully completed the course if the combined total of CIE and SEE is at least 40 out of 100 marks.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING - DATA SCIENCE

The Correlation of Course Outcomes (CO's) and Program Outcomes (PO's)

Subject Code	1BCS302			Title: Object Oriented Programming with Java								
List of Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Total
CO-1	3	2	2	—	2	—	—	—	—	—	—	09
CO-2	3	3	3	—	2	—	—	—	2	—	—	13
CO-3	3	3	3	—	3	—	—	—	2	—	—	14
CO-4	3	3	3	2	3	—	—	—	2	—	—	16
CO-5	3	3	3	2	3	—	—	—	—	—	—	14
CO-6	3	2	3	2	3	—	—	—	2	2	—	17

Note: 3 = Strong Contribution 2 = Average Contribution 1 = Weak Contribution - = No Contribution

The Correlation of Course Outcomes (CO's) and Program Specific Outcomes (PSO's)

Subject Code	1BCS302	Title: Object Oriented Programming with Java	
List of Course Outcome's	PSO1	PSO2	Total
CO-1	3	2	5
CO-2	3	2	5
CO-3	3	2	5
CO-4	3	2	5
CO-5	3	2	5
CO-6	3	2	5

Note: 3 = Strong Contribution 2 = Average Contribution 1 = Weak Contribution - = No Contribution