

Branch Name:	Bachelor of Computer Application (Graphics Animation and VFX)
Program Code:	CS102
Course Title:	Java Programming
Course Code:	24-CAM207-2C
Pre-requisite Course:	Basic Knowledge of Object-Oriented Programming (OOP).

Course Objective (CO):

This course is intended to make the students:

- To understand object-oriented concepts using Java
- To learn Java fundamentals (syntax, JVM, data types, control structures)
- To develop programs using classes, inheritance, interfaces, and packages
- To handle runtime errors using exception handling

Teaching and Examination Scheme:

Teaching Scheme (Hours per week)				Evaluation Scheme (Marks)								
Lecture (L)	Tutorial (T)	Practical (P)	Credit	Theory (Marks)				Practical (Marks)				Total (Marks)
				University Assessment		Continuous Assessment		University Assessment		Continuous Assessment		
				Min	Max	Min	Max	Min	Max	Min	Max	
3	--	4	5	18	50	09	25	09	25	09	25	100

Course Contents:

Sr. No	Topic	Total Hours	Weightage (%)
1.	Introduction to OOP • Basic concepts of OOP, Benefits and applications Introduction to Java • History and features of Java, JDK, JRE, JVM, Bytecode and platform independence, Java program structure, Compiling and executing Java programs Basics of Java Programming • Tokens: Keywords, Identifiers, Literals, Operators, Data Types and Variables, Scope of variables, Constants (final), Type Conversion, Garbage Collection, Command Line Arguments, Arrays Control Statements • Selection: if, switch • Iteration: for, while, do-while • Jump statements: break, continue, return	12	25
2.	Classes, Objects & Constructors Classes and Objects • Defining classes and methods, Creating objects, Accessing class members, Static members Methods • Method Overloading, Nesting of methods, Use of this keyword, Passing object as argument, Returning object Constructors • Introduction to constructors, Default constructor, Parameterized constructor, Constructor overloading	12	25

3.	Inheritance, Interfaces & Packages Inheritance - Introduction and types of inheritance, Method overriding, final method and class, Use of super keyword, Abstract class and abstract methods Interfaces (Multiple Inheritance) - Defining interfaces, Implementing interfaces, Extending interfaces, Packages Introduction to Java API packages: - Java.lang, java.util, java.io, java.net, java.awt - Creating packages, Adding classes to packages, CLASSPATH, Importing packages Access Control - Access modifiers: public, private, protected, default Important Classes java.lang package : Math, String, Wrapper classes java.util package : Vector, Stack, Date	14	30
4.	Exception Handling Error & Exception Concepts - Errors and types of errors, Exception and types of exceptions Exception Handling Mechanism - try, catch, finally, Multiple catch blocks, throw keyword, throws clause User-defined Exceptions - Creating and handling custom exceptions	10	20

Course Learning Outcomes (CLO):

At the end of this course, the student would be able

- CLO1:** Apply object-oriented programming concepts such as class, object, encapsulation, inheritance, and polymorphism to solve real-world problems.
- CLO2:** Understand and implement the fundamentals of Java programming, including syntax, data types, operators, control structures, and arrays.
- CLO3:** Demonstrate knowledge of Java environment and core concepts such as JVM, JDK, program structure, constructors, and memory management.
- CLO4:** Develop Java applications using classes, methods, constructors, and packages, and perform program execution, compilation, and debugging.
- CLO5:** Build and deploy enterprise web applications.

List of Text Book:

- Programming with Java — by E. Balagurusamy, Tata McGraw Hill.
- Core Java: An Integrated Approach — by R. Nageswara Rao, Dreamtech.
- Let Us Java, — by Yashwant Kanetkar, BPB Publications.

List of Reference Book:

- Java: The Complete Reference — by Herbert Schildt, McGraw-Hill.
- Core Java Volume I - Fundamentals.— by Cay S. Horstmann & Gary Cornell
- Java: How to Program – by Paul Deitel & Harvey Deitel

E-Resource / Web Links:

- Oracle Java Tutorials – <https://docs.oracle.com/javase/tutorial/>
- TutorialsPoint Java – <https://www.tutorialspoint.com/java/>
- GeeksforGeeks Java – <https://www.geeksforgeeks.org/java/>
- W3Schools Java – <https://www.w3schools.com/java/>
- JavaTpoint – <https://www.javatpoint.com/java-tutorial>

6. BeginnersBook – <https://beginnersbook.com/java-tutorial-for-beginners-with-examples/>
7. Guru99 Java – <https://www.guru99.com/java-tutorial.html>

List of Experiments:

Note: The experiment list provided beneath is for reference only. The course teacher may change/formulate it as per his/her methodology and requirement.

Practical List

Section–A: Basic Programs & Control Statements

1. Write a program to display the message "My First Java Program".
2. Write a program to find the area of a circle.
3. Write a program to accept command-line arguments and display them.
4. Write a program to find the largest number among two numbers.
5. Write a program to find the largest number among three numbers.
6. Write a program to implement the switch statement.
7. Write a program to find the sum of digits of a number.
8. Write a program to display the factorial of a number.
9. Write a program to demonstrate break and continue statements.

Section–B: Classes, Objects & Methods

10. Create a Student class and calculate Total, Percentage, and Grade.
11. Create an Employee class and generate Salary Slip.
12. Write a program to demonstrate static members.
13. Write a program to demonstrate method overloading.
14. Write a program to demonstrate nesting of methods.
15. Write a program to pass an object as a method argument.
16. Write a program to return an object from a method.

Section–C: Constructors

17. Write a program to implement default constructor.
18. Write a program to implement parameterized constructor.
19. Write a program to implement constructor overloading.

Section–D: Inheritance & Polymorphism

20. Write a program to implement single inheritance.
21. Write a program to implement multilevel inheritance.
22. Write a program to implement hierarchical inheritance.
23. Write a program to implement method overriding.
24. Write a program to demonstrate final method.
25. Write a program to demonstrate final class.
26. Write a program to demonstrate use of super keyword.
27. Write a program to implement abstract class and abstract method.

Section–E: Interfaces

28. Write a program to implement an interface.
29. Write a program to implement multiple inheritance using interfaces.

Section–F: Packages & Built-in Classes

30. Write a program to access classes within the same package.
31. Write a program to import classes from another package.
32. Write a program to demonstrate access modifiers (public, private, protected, default) using packages.
33. Write a program to implement methods of Math class.
34. Write a program to implement methods of String class.

35. Write a program to implement methods of Vector class.

36. Write a program to implement methods of Stack class.

Section–G: Exception Handling

37. Write a program using try–catch for exception handling.

38. Write a program using finally block.

39. Write a program using multiple catch blocks.

40. Write a program to demonstrate throw statement.

41. Write a program to demonstrate throws clause.

42. Write a program to create and handle user-defined exception.

Mapping of CLO with PO and PSO

Course Learning Outcomes	Program Outcomes (POs)												Program Specific Outcomes (PSOs)		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CLO1	H	H	M	M	L				M				H	M	
CLO2	H	H	H	M	M	M			L	M			H	M	M
CLO3	M	H				M							M		
CLO4	H		H			M				M			H	M	
CLO5		H	H	H	M	M	M		H	M	M		H	M	H
CLO6		H	H	H	M	M	M		H	M	M		H	M	H

H: High, M: Medium, L: Low