

DEPARTMENT OF BACHELOR OF COMPUTER APPLICATIONS

COURSE MODULE: FUNDAMENTALS OF JAVA

Course Coordinator: Prof. Yeshashwini Bhandari K R				Academic Year: 2025-26	
Department: Bachelor of Computer Applications					
Course Code	Course Title	Core/Elective	Prerequisite	Contact Hours L: T: P:S	Total Hrs/ Sessions
BBCA301	FUNDAMENTALS OF JAVA	IPCC		2:2:0:0	28(TH) + 20(LAB)
Course Learning Objective:					
The course will enable the students to:					
<ul style="list-style-type: none"> • To learn primitive constructs JAVA programming language. • To understand Object Oriented Programming Features of JAVA. • To gain knowledge on: packages, exceptions and multithreaded programming. 					
Module-1					
<p>An Overview of Java: Object-Oriented Programming (Two Paradigms, Abstraction, The Three OOP Principles), Using Blocks of Code, Lexical Issues (Whitespace, Identifiers, Literals, Comments, Separators, The Java Keywords).</p> <p>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.</p> <p>Operators: Arithmetic Operators, Relational Operators, Boolean Logical Operators, The Assignment Operator, The ? Operator, Operator Precedence, Using Parentheses.</p> <p>Control Statements: Java's Selection Statements (if, The Traditional switch), Iteration Statements(while, do while, for, The For-Each Version of the for Loop, Nested Loops), Jump Statements (Using break, continue, return).</p>					
Module-2					
<p>Introducing Classes: Class Fundamentals, Declaring Objects, Assigning Object Reference Variables, Introducing Methods, Constructors, The this Keyword, Garbage Collection.</p> <p>Methods and Classes: Overloading Methods, Objects as Parameters, Argument Passing, Returning Objects, Recursion, Access Control, Understanding static, Introducing final.</p>					
Module-3					
<p>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.</p> <p>Interfaces: Interfaces, Default Interface Methods, Use static Methods in an Interface, Private Interface Methods.</p>					
Module-4					
<p>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.</p>					
Module-5					
<p>Packages: Packages, Packages and Member Access, Importing Packages.</p> <p>Multithreaded Programming: The Java Thread Model, The Main Thread, Creating a Thread, Creating Multiple Threads, Using isAlive() and join(), Thread Priorities, Synchronization, Interthread Communication, Suspending, Resuming, and Stopping Threads, Obtaining a Thread's State.</p>					

DEPARTMENT OF BACHELOR OF COMPUTER APPLICATIONS

PRACTICAL CONTENT FOR IPCC

SL No	EXPERIMENTS
01	Write a Java program that uses a for loop to calculate the sum of even numbers between 1 and 100. The program should use continue to skip odd numbers and print the final sum of even numbers after the loop finishes.
02	Write a Java program to input a number and determine if it is prime or not.
03	Write a Java program to create a class Student with members name and rollNumber. Define a constructor to initialize these members. Then, create a method displayDetails() to display the student's details.
04	Write a Java program to define a class Book with attributes title, author, and price. Use a constructor to initialize these attributes. Demonstrate the use of the this keyword to refer to instance variables and display the book details.
05	Write a Java program to demonstrate the use of the static keyword by creating a Counter class with a static variable count that tracks the number of instances of the class. Also, define a final variable to hold a constant value such as the maximum allowed instances.
06	Write a Java program that demonstrates method overriding and dynamic method dispatch. Create a base class Shape with a method area(). Override the area() method in derived classes Circle and Rectangle. Use dynamic method dispatch to call the area() method from the base class reference.
07	Write a Java program to create an interface Shape with methods area() and perimeter(). Implement this interface in two classes: Circle and Rectangle. In each class, implement the methods to calculate and display the area and perimeter. Instantiate the classes and display the results for both shapes.
08	Write a Java program that accepts an array of integers and attempts to access the elements using an invalid index. Use multiple catch clauses to handle ArrayIndexOutOfBoundsException and NumberFormatException (if the user inputs a non-integer value).
09	Write a Java program that creates a package named shape. In this package, define a class: Rectangle. The class should have methods to calculate the area and perimeter. In the main class, import the shape package and access the methods to calculate area and perimeter of shape.
10	Write a Java program to create two threads. The first thread should print numbers from 1 to 5, and the second thread should print the alphabet letters from A to E. Start both threads simultaneously and ensure that both print their respective outputs.

Course Outcomes:

At the end of the course, the student will be able to:

- CO1.** Demonstrate proficiency in writing simple programs involving branching and looping structure.
- CO2.** Design a class involving data members and methods for the given scenario.
- CO3.** Apply the concepts of inheritance and interfaces in solving real world problems.
- CO4.** Use the concept exception handling in solving complex problems.
- CO5.** Apply concepts of packages and multithreading in program development

DEPARTMENT OF BACHELOR OF COMPUTER APPLICATIONS

Assessment Details (both CIE and SEE) of Integrated Course:

To satisfy academic requirements and earn credits for each subject/course, students must meet the following criteria:

- Secure at least 40% (20/50 marks) in Continuous Internal Evaluation (CIE)
- Obtain a minimum of 35% (18/50 marks) in the Semester End Exam (SEE)
- Achieve a combined total of at least 40% (40/100 marks) in both CIE and SEE

CIE Marks Breakdown

- **Theory Component:** 25 marks
 - **Internal Assessment Tests:** 15 marks (2 tests, each 15 marks, 1-hour duration)
 - **Test 1:** After covering 40-50% of the syllabus
 - **Test 2:** After covering 85-90% of the syllabus
 - **Other Assessment Methods:** 10 marks
- Practical Component: 25 marks

CIE Marks Calculation

- The sum of marks from the two tests and other assessment methods will be scaled down to determine the CIE marks for the theory component (out of 25 marks).

Qualifying Criteria

- To qualify in the CIE of the theory component, students must secure at least 40% of 25 marks, i.e., 10 marks.

CIE for Practical Component of IPCC

Evaluation Criteria:

- Laboratory Record and Experimentation: 15 marks
 - Evaluation of laboratory record and experimentation (including viva-voce) after each experiment/program
 - Marks awarded on the same day
- Laboratory Test: 10 marks
 - Test conducted after completion of all laboratory sessions (duration: 2-3 hours)
 - Originally marked out of 50, scaled down to 10 marks

CIE Marks Calculation:

- Write-up Evaluations: Marks awarded for each experiment report (out of 10) added and scaled down to 15 marks
- Total CIE Marks: Scaled-down marks of write-up evaluations and laboratory test added to give CIE marks for the practical component (out of 25)

Qualifying Criteria:

- To qualify in the CIE of the practical component, students must secure at least 40% of 25 marks, i.e., 10 marks

Semester End Examination (SEE) for IPCC

Theory SEE:

- Conducted by the University as per the scheduled timetable
- Common question papers for the course (duration: 3 hours)

Question Paper Structure:

- 10 questions, each worth 20 marks
- 2 questions from each module, with a mix of topics under each module (max. 3 subquestions per question)

DEPARTMENT OF BACHELOR OF COMPUTER APPLICATIONS

- Students must answer 5 full questions, selecting one from each module

Marking Scheme:

- Marks scored will be proportionally scaled down to 50 marks

IPCC Examination Structure:

- Theory portion: Both CIE and SEE components
- Practical portion: CIE component only (no SEE)
- Note: SEE questions may include topics from the practical component

List of Textbooks

- Java: The Complete Reference, Twelfth Edition, by Herbert Schildt, November 2021, McGraw-Hill, ISBN: 9781260463422
- Programming with Java, 6th Edition, by E Balagurusamy, Mar-2019, McGraw Hill Education, ISBN: 9789353162337.
- Thinking in Java, Fourth Edition, by Bruce Eckel, Prentice Hall, 2006

Web links and Video Lectures (e-Resources):

- https://onlinecourses.nptel.ac.in/noc22_cs47/preview
- https://www.youtube.com/watch?v=3a1FXBR6QXY&list=PLZ2ps_7DhBaXTORd_OKXZRXCmH3dGnBZ&index=1

Skill Development Activities Suggested:

The students, with the help of the course teacher, can take up Java-related technical activities such as coding exercises, mini projects, and problem-solving tasks, or interact with industry to identify problems for study in the form of research, testing, or projects. Creative and innovative solutions should be attempted, and the prepared report shall be evaluated for CIE marks.

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

SUBJECT CODE: BBKA301		TITLE: FUNDAMENTALS OF JAVA					FACULTY: YESHASHWINI BHANDARI K R	
List of Course Outcomes	Program Outcomes							
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO-1								
CO-2								
CO-3								
CO-4								
CO-5								
Total								

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