

Code: 20CS6522

III B.Tech - I Semester - Regular Examinations - NOVEMBER 2024

**ADVANCED JAVA PROGRAMMING
(HONORS in COMPUTER SCIENCE & ENGINEERING)**

Duration: 3 hours

Max. Marks: 70

Note: 1. This paper contains questions from 5 units of Syllabus. Each unit carries 14 marks and have an internal choice of Questions.

2. All parts of Question must be answered in one place.

BL – Blooms Level

CO – Course Outcome

			BL	CO	Max. Marks
UNIT-I					
1	a)	Explain the structure and use of a generic class with two type parameters. Illustrate with a detailed example and describe how this setup differs from a non-generic approach.	L2	CO2	6 M
	b)	Write a comprehensive Java program that demonstrates the use of bounded types in generics. Explain how bounded types ensure type safety while allowing flexibility, providing examples of both upper and lower bounds in your code.	L3	CO2	8 M
OR					
2	a)	Describe the concept of wildcard arguments ('? Extends' and '? Super') in generics. Provide examples showing their usage and explain the scenarios where they are most effective.	L2	CO1 CO2	6 M
	b)	Develop a program that implements a generic method for sorting elements in an array. Discuss how the generic method adapts to different data types, and explain the use of generic type inference in simplifying the method call.	L3	CO2	8 M

UNIT-II					
3	a)	Analyze the benefits and limitations of using lambda expressions in Java, especially in the context of simplifying code and reducing verbosity. Provide examples comparing traditional anonymous classes with lambda expressions, highlighting the improvements lambda expressions bring.	L4	CO3	8 M
	b)	What is a functional interface, and how does it relate to lambda expressions? Provide examples to illustrate this relationship.	L1	CO1 CO3	6 M
OR					
4	a)	Explain how block lambda expressions differ from simple lambda expressions. Provide examples that demonstrate the use of block lambdas, especially when multiple statements or complex logic are involved.	L2	CO3	7 M
	b)	Differentiate between block lambda expressions and single-expression lambdas. Discuss when each should be used, focusing on readability, maintainability, and performance considerations.	L4	CO1 CO3	7 M
UNIT-III					
5	a)	Define the Java Collection Framework and explain its significance in managing groups of objects. Describe the core interfaces included in the framework and their primary functions.	L1	CO4	6 M
	b)	Critically assess the role of iterators and spliterators in traversing collections. Compare the benefits and limitations of each, and discuss situations where spliterators provide advantages over traditional iterators.	L5	CO1 CO4	8 M

OR					
6	a)	Explain the hierarchy of the Collection Framework, focusing on interfaces like ‘List’, ‘Set’, ‘Queue’, and their subinterfaces. How do these interfaces define the behavior of different collections?	L2	CO4	6 M
	b)	Analyze the usage of the ‘Queue’ and ‘Deque’ interfaces in Java. Discuss the differences between a traditional ‘Queue’ implementation and a ‘Deque’, providing examples where each is more suitable.	L4	CO4	8 M
UNIT-IV					
7	a)	Describe the role of comparators in sorting user-defined objects within collections. Provide examples demonstrating the difference between implementing <i>Comparable</i> and using <i>Comparator</i> .	L2	CO4	7 M
	b)	Discuss how maps are useful in real-world applications, focusing on scenarios such as lookup tables and caching.	L3	CO4	7 M
OR					
8	a)	Compare and contrast the use of built-in algorithms provided by the <i>Collections</i> class, such as <i>sort()</i> , <i>binarySearch()</i> , and <i>reverse()</i> .	L4	CO 4	7 M
	b)	List the key methods provided by the <i>Arrays</i> class for working with collections. How do these methods simplify operations on arrays and collections?	L1	CO1 CO4	7 M

UNIT-V

9	a)	Write a Java program that uses the <i>Calendar</i> and <i>GregorianCalendar</i> classes to manage and manipulate dates, such as setting specific dates, comparing dates, and calculating future or past dates. Discuss the benefits of using these classes over the older <i>Date</i> class.	L3	CO4 CO5	8 M
	b)	Describe the process of pattern matching using regular expressions in Java. Explain the basic syntax and rules of regular expressions and illustrate how to match different types of patterns using the <i>Pattern</i> and <i>Matcher</i> classes.	L2	CO5	6 M
OR					
10	a)	Compare and contrast the different pattern-matching options available in Java's regular expression library. Analyze the performance and flexibility of regular expressions for tasks like validation, search, and text extraction in large datasets.	L4	CO4 CO5	8 M
	b)	Develop a Java program that validates common input formats such as email addresses, phone numbers, and postal codes using regular expressions. Explain how you designed the patterns and how they address edge cases.	L3	CO5	6 M