Name:
Enrolment No:

End Semester Examination, December 2023

Program Name:	**BTech Big Data**	**Semester**	**: 5th**
Course Name:	**Advanced Functional Thinking**	**Time**	**: 3 hrs**
Course Code:	**CSBD3002**	**Max. Marks**	**: 100**

Section A
(5Qx4M=20Marks)

S. No.		Marks	CO
Q 1	Provide examples of how Scala's type inference property aids in writing concise functional code.	**4**	**CO1**
Q 2	Compare object oriented and functional programming paradigms.	**4**	**CO1**
Q 3	Differentiate between Pure functions and Closures.	**4**	**CO1**
Q 4	Explain the difference between map and filter functions in Scala. Provide use cases for each.	**4**	**CO2**
Q 5	Write a Scala program to read a weekday number and print weekday name using the match case.	**4**	**CO1**

Section B
(4Qx10M=40Marks)

Q 6	i. What is tail recursion optimization, and why is it important in functional programming? Write code for factorial using tail recursion. ii. Given a list of strings, use the map function to transform each string to uppercase.	**10**	**CO2**
Q 7	Explain the use of anonymous functions. Write a scala code to create anonymous functions for add, sub, and mul with _ operator.	**10**	**CO2**
Q 8	Define Singleton object and explain its properties using examples. Or Create a counter using a Singleton object.	**10**	**CO3**
Q 9	Compare strict and lazy evaluation. Write SCALA code for both to illustrate the execution difference between them.	**10**	**CO4**

Section C
(2Qx20M=40Marks)

Q 10	Explain the concept of proxy pattern in Scala and its features and significance in functional programming?	**20**	**CO3**

	Or Explain scala inhertance using traits. Define three traits: Readable (with a read method), Writable (with a write method), and Erasable (with an erase method). Create a class File that extends these traits and implements the required methods. Show the use of each method.		
Q 11	i. Explain the concept of functors and monads in the context of Scala and functional programming with suitable examples. ii. What will be the output of the codes: a. val s = "Scala programming is fun" val result = s.split(" ").map(_.reverse) .mkString(" ") println(result) b. val numbers = List(5, 10, 15, 20) val result = numbers.reduce((x, y) => x * y) println(result)	**20**	**CO4**