

## Assignment-7 for Lambdas and Functional Programming

Subject: CSW2 (CSE 3141)

Session: Jan to May 2025

Branch: CSE

Section: All

Course Outcomes: CO3

Learning Levels: Remembering (L1), Understanding (L2), Application (L3), Analysis (L4),  
Evaluation(L5), Creation (L6)

Q no.	Questions	Learning Levels
Q1.	You are required to create a Java program that uses a functional interface to process strings. Define a functional interface named <b>StringProcessor</b> with a method that takes a string as input and returns an integer. Within the <b>main</b> method of a class, use a lambda expression to implement this method such that it returns the length of the given string.	L1, L2
Q2.	Create a functional interface <b>Calculator</b> with methods for addition, subtraction, multiplication, and division. Implement these methods using lambda expressions. Define the <b>Calculator</b> functional interface with methods for arithmetic operations. Implement the interface methods using lambda expressions for basic arithmetic operations.	L2, L3
Q3.	Write a program that sorts a list of strings based on their lengths in descending order. Define a custom comparator using a lambda expression that compares strings based on their lengths. Use the custom comparator to sort the list of strings in descending order of length.	L3, L4
Q4.	Create a functional interface <b>Shape</b> with a method double <b>area()</b> and a default method void <b>printArea()</b> . Implement the interface using lambda expressions for different shapes. Define the Shape functional interface with an area method. Implement the interface for shapes like circle, square, and rectangle using lambda expressions. Use the default method to print the area of each shape.	L2, L3
Q5.	Write a Java program that demonstrates the concept of a function returning another function using Java's <b>Function</b> interface. Define a method that returns a lambda function that takes an integer as input and returns its square. In the <b>main</b> method, retrieve the returned function and use it to calculate and display the squares of different numbers.	L2, L3
Q6.	Write a Java program that demonstrates the use of a lambda expression with the <b>Function</b> functional interface to calculate the factorial of a number. Define a method that returns a lambda expression that takes an integer as input and computes its	L3, L4

	factorial using an iterative approach. In the <b>main</b> method, use this lambda expression to calculate and display the factorial of a given number.	
<b>Date and Time API</b>		
Q7.	Using the Java Date and Time API, write a Java program to perform the following tasks: a) Retrieve and display the current date and time. b) Calculate the date that is 2 weeks from today. c) Format and display the current date in the "MM/dd/yyyy" format.	L1, L2
Q8.	Write a Java program that demonstrates how to format a <b>LocalDateTime</b> object using the <b>DateTimeFormatter</b> class. Create a <b>LocalDateTime</b> instance representing a specific date and time. Then, use a custom format pattern "yyyy-MM-dd HH:mm:ss" to convert this date-time into a formatted string. Finally, print the formatted result.	L2, L3
Q9.	Write a Java program to calculate the simple interest on a current account using the Java Date and Time API. The program should prompt the user to enter the start date and the initial amount. Assume the rate of interest is <b>8%</b> per annum. Calculate the number of days between the start date and the current date, then compute the interest using the simple interest formula. Finally, display the total amount after adding the interest to the initial amount.	L3, L4
Q10.	Write a Java program using the Java Date and Time API to calculate the number of days between two dates. The program should prompt the user to enter a start date and an end date (in <b>YYYY-MM-DD</b> format), and then display the total number of days between the two dates using the <b>ChronoUnit.DAYS</b> method.	L3, L4
<b>-END-</b>		