

Multithreading and Reactive Programming

Subject: CSW2(CSE 3141)

Section: All

Session: Feb 2024 to April 2024

Branch: CSE&CSIT

Q1. Write a Java program to create a simple calculator that performs arithmetic operations (addition, subtraction, multiplication, division) using **multiple threads**. Each arithmetic operation should be handled by a separate thread.

Q2. Write a Java program that uses reactive programming to read a file asynchronously. Use RxJava or another reactive library to handle the **file reading** and processing.

Q3. Write a Java program to **multiply two matrices** using multithreading. Divide the task of multiplying rows of the matrices among multiple threads to improve performance.

Q4. Implement a program where two **threads communicate** with each other using wait() and notify() methods. One thread should print even numbers, and the other should print odd numbers in sequence.

Q5. Implement a program that demonstrates the **use of locks** (e.g., ReentrantLock) for thread synchronization. Create a scenario where multiple threads access a shared resource, and use locks to ensure that only one thread can access the resource at a time.

Q6. Write a Java program that **generates prime numbers** up to a given limit using multiple threads. Each thread should generate a subset of the prime numbers.

Q7. Implement a Java program that demonstrates the use of **CompletableFuture** to perform asynchronous tasks and handle their results. Create **CompletableFuture** instances to perform tasks such as fetching data from a remote server or processing a large file asynchronously. Use **thenApply()** or **thenCombine()** to combine the results.

Q8. Develop a Java program that **analyzes real-time weather data** using reactive programming. The program should fetch weather data from a weather API asynchronously and perform analysis (e.g., temperature trends, rainfall predictions). Use a reactive approach to handle the asynchronous nature of weather data updates. Use reactive operators (e.g., map, filter) to process and analyze the weather data stream.

Q9. Write a Java program that **processes a large text file** using multiple threads. Each thread should read a portion of the file, process it (e.g., count words, find specific patterns), and then merge the results to get the final output. The program should use a thread-safe data structure (e.g., ConcurrentHashMap) to store intermediate results from each thread. Use synchronization mechanisms to ensure that each thread processes a unique portion of the file.

Q10. Write a Java program that **processes a batch of images concurrently** using multiple threads. Each thread should apply a specific image processing operation (e.g., resizing, filtering) to its assigned image. Use a thread pool to manage the execution of image processing tasks. Ensure that each image is processed only once by a single thread to avoid duplication or conflicts.
