GUJARAT UNIVERSITY
5 Year integrated M. Sc. (Computer Science)
Semester: V
Java Programming
Practical Assignment - 5

Date : 12-11-2025
Submission Date : 25-11-2025

---

1. Thread Creation using `Thread` class
   Write a Java program to create two threads:
   - Thread 1 prints numbers from 1 to 10
   - Thread 2 prints alphabets from A to J
     Run both threads simultaneously.
2. Thread Creation using `Runnable` interface
   Create a program where one thread displays even numbers and another thread displays odd numbers up to 20 using the `Runnable` interface.
3. Thread Priority Example
   Create three threads with different priorities (`MIN_PRIORITY`, `NORM_PRIORITY`, and `MAX_PRIORITY`) and observe the order of execution.
4. Thread Sleep and Join Example
   Create a thread that counts from 1 to 5 with a delay of 1 second between counts using `Thread.sleep()`.
   Use `join()` in the main thread to wait until the counting thread finishes.
5. Synchronization Example – Banking System
   Write a Java program where multiple threads try to withdraw money from a shared bank account object.
   Use synchronization to prevent race conditions.
6. Producer-Consumer Problem
   Implement the classic producer-consumer problem using `wait()` and `notify()`.
   - Producer adds items to a shared buffer
   - Consumer removes items from the buffer
     Ensure proper thread synchronization.
7. Thread Communication Example
   Create two threads:
   - One thread prints numbers from 1 to 10
   - Another thread prints “Halfway there” when count reaches 5
     Use inter-thread communication for coordination.
8. Multithreading with File Handling
   Write a Java program that reads data from one file using one thread and writes the data to another file using another thread simultaneously.
9. Matrix Multiplication using Threads
   Perform matrix multiplication using multiple threads — one thread per row or per cell.
10. Thread Pool Example (Executor Framework)
    Create a fixed thread pool of 3 threads using `Executors.newFixedThreadPool()`.
    Submit 5 tasks that print the thread name and task number.
11. Daemon Thread Example
    Create a daemon thread that continuously prints “Background task running...” while the main thread executes a separate task.
12. Deadlock Simulation
    Write a Java program that demonstrates a deadlock situation between two threads, and then modify it to resolve the deadlock using proper synchronization order.
13. Multithreading with Anonymous Class / Lambda Expression
    Use anonymous inner classes or lambda expressions to create and run threads that perform simple calculations (like factorial or sum of array).

Q14. Write a program to implement a stack using arrays with the following operations:

- push() – to insert an element into the stack

- pop() – to remove the top element
- peek() – to view the top element
- display() – to display all elements of the stack

Q15. Implement a stack using Linked List and perform push and pop operations.
Q16. Write a Java program to check whether a given string is a palindrome using a stack.
Q17. Write a program to implement a queue using arrays with operations:

- enqueue() – insert element
- dequeue() – remove element
- display() – show all elements

Q18. Implement a circular queue using arrays. Perform insertion, deletion, and display operations.
Q19. Write a Java program to implement a priority queue where higher numbers have higher priority.
Q20. Write a program to implement a singly linked list with the following operations:

- Insert at beginning
- Insert at end
- Delete from beginning
- Delete from end
- Display the list

Q21. Implement a doubly linked list and perform insertion and deletion operations at both ends.
Q22. Write a program to implement a circular linked list and perform insertion and deletion.
Q23. Write a Java program to reverse a linked list.
Q24. Write a program to implement Bubble Sort to sort an array of integers in ascending order.
Q25. Write a program to implement Selection Sort to sort an array of numbers in descending order.
Q26. Write a program to implement Insertion Sort and display the number of passes required for sorting.
Q27. Write a program to implement Merge Sort for sorting an array in ascending order.
Q28. Write a program to implement Quick Sort and display the pivot element for each pass.
Q29. Write a program to evaluate a postfix expression using stack.
Q30. Write a program to convert infix expression to postfix using stack.
Q31. Write a program to count the total number of nodes in a linked list.
Q32. Write a program to merge two sorted linked lists into one sorted list.
Q33. Write a program to find the middle element of a linked list without using length.