Department of ICT
Faculty of Technology
University of Colombo
IC4302 – Programming III

Group Assignment: Java Application for Sorting Algorithm Performance Evaluation

Objective:

This group assignment requires the development of a Java application that allows the user to upload a dataset in .csv format, select a column for sorting, apply various sorting algorithms, and evaluate their performance based on execution time. Each group can consist of up to 3 members. The assignment also includes a recorded presentation where each group member explains their contribution and demonstrates the application.

Group Guidelines:

1. **Group Size**
   - This is a group assignment, with a **maximum of 3 members per group**.
   - You can complete this assignment with your already allocated research group.
2. **Submission Platform**
   - The final submission must be uploaded to **Moodle**.
3. **Presentation**
   - The group must **record a presentation** using Zoom (or any other suitable tool).
   - The presentation must include a **detailed code explanation** and an **application demonstration**.
   - Each group member must explain the portion of the code they contributed to, ensuring individual accountability.
   - **Self-introduction** before each member's explanation is mandatory (e.g., "I'm Shanaka, my student number is T085XX").
4. **Deadline**
   - The assignment must be submitted **on or before the given deadline**.
   - Late submissions may result in reduced marks.

Application Requirements:

1. **Dataset Upload**
   - The application must allow users to upload a `.csv` file and display the available columns.
     - Note: Sample CSVs can download using Kaggle - https://www.kaggle.com/datasets

    - After uploading, the user must be able to select a specific column to sort.
        - Note: Always need to select numeric columns to sort.
2. **Sorting Algorithms**: Implement the following sorting algorithms:
    - Insertion Sort
    - Shell Sort
    - Merge Sort
    - Quick Sort
    - Heap Sort
3. **Performance Evaluation**:
    - Measure the execution time of each sorting algorithm and display it.
    - Identify and display the best-performing algorithm based on the shortest execution time.
4. **User Interface**:
    - The application should be GUI-based, and the interface should be user-friendly and allow for easy column selection and sorting execution.

Submission Deliverables

1. **Code Submission**
    - Upload the code to Moodle.
2. **Recorded Presentation**:
    - Each member must explain their part of the code and demonstrate the functionality they worked on.
    - The presentation should include:
        1. **Introduction**: Each member introduces themselves with their name and student number.
        2. **Code Explanation**: Explanation of the portion of the code they contributed to.
        3. **Application Demonstration**: Demonstrating the application functionality, including file upload, column selection, sorting, and performance evaluation.
    - Upload the recorded presentation to Moodle along with the code.

Presentation Guidelines

- Each member must clearly explain their contribution.
- The presentation should include:
    1. **File Upload Functionality**: Demonstration of uploading the `.csv` file and showing the available columns.
    2. **Column Selection and Sorting**: Demonstration of selecting a column and applying the sorting algorithms.
    3. **Performance Evaluation**: Showing the execution times for each algorithm and identifying the best one.
- Each member should **introduce themselves** before presenting their part. Example: "I'm Shanaka, my student number is T085XX."

Evaluation Criteria

The assignment will be evaluated based on the following criteria:

1. **Clarity of the Code (20%)**:
    - Code structure and organization.
    - Clear comments and documentation.
    - Proper use of classes, methods, and algorithms.
2. **Correctness of the Algorithms (30%)**:
    - Proper implementation of each sorting algorithm (Insertion Sort, Shell Sort, Merge Sort, Quick Sort, Heap Sort).
    - Correct sorting based on the selected column.
3. **Individual Contribution (20%)**:
    - Each group member must demonstrate and explain their individual contribution during the presentation.
4. **Overall Completeness of the Application (30%)**:
    - The application should meet all functional requirements.
    - The sorting functionality, performance evaluation, and user interface should be fully operational.
    - Proper handling of edge cases (e.g., invalid file uploads, empty columns).

Suggested Workflow:

1. **Group Coordination**
    - Decide the individual responsibilities among group members. Suggested roles:
        - Member 1: File upload functionality and dataset handling.
        - Member 2: Implementation of sorting algorithms.
        - Member 3: Performance evaluation and user interface (GUI or command-line).
2. **Code Development**
    - Develop the application in parts, and integrate the code in the shared repository.
    - Ensure regular communication and collaboration via the shared repository.
3. **Testing**
    - Test the application thoroughly with different `.csv` files.
    - Test the correctness of sorting algorithms and the performance evaluation.
4. **Presentation**
    - Record the Zoom presentation, with each member introducing themselves and explaining their contribution.
    - Include a full demonstration of the application.

Timeline:

- **Week 1**: Group formation and initial planning.
- **Week 2**: Implementation of file upload functionality and dataset handling.
- **Week 3**: Implementation of sorting algorithms.
- **Week 4**: Performance evaluation and user interface development.
- **Week 5**: Final integration, testing, and presentation recording.

- **Submission Deadline**: Upload the assignment (code and recorded presentation) on or before the deadline to Moodle.

Notes:

- Proper time management and task delegation among group members is crucial for successful completion.
- Make sure to handle file upload errors and invalid input scenarios gracefully within the application.
- The presentation is an important component of the assignment, as it demonstrates individual contributions and the overall functionality of the application.

****