Coursework 3 Task Sheet

CPT204 AY2425

CW3 Introduction

Coursework 3 is the final coursework component of the course this semester, which accounts for **40% of your final mark**. You will form **a team of two** with your classmates and apply object-oriented principles and advanced data structures, as you have learned throughout the semester, to solve the tasks specified in this document. You will be tasked with writing a report and creating a video presentation to demonstrate your knowledge of 1. object-oriented concepts, 2. problem-solving techniques, 3. AI-assisted team cooperation, 4. equality, diversity, and inclusion principles, and 5. planning and self-learning in object-oriented programming.

You are required to submit the following files:

- Java code in a ZIP file;
- A Word report;
- An MP4 video;
- A PowerPoint (PPT) presentation used in the video.

Phase	Date
Task Sheet Release	Week 9, Monday, 14th of April, 9:00 am (UTC+8).
Submission	Week 12, Sunday, 11th of May, 11:59 pm (UTC+8).
Late Submission	Week 13, Monday-Friday, 12th-16th of May. 5% lateness penalty per day, max 5 days (Monday-Friday), no submissions are accepted after 11:59 pm (UTC+8), 16th of May.

Tasks Overview and Requirements

Overview

For **Tasks A and B**, you are required to design and implement an application for planning road trips in the USA. Assume the user will input a starting city/origin (e.g., New York) and an ending city/destination (e.g., Chicago). In addition, the user will input zero, one, or more places of interest (e.g., NASA Space Center, Hollywood Sign, and so on. Once implemented, your solution must find a route from the starting city to the ending city in a way that visits each of the places of interest while keeping the number of miles driven (the total distance) to a minimum.

With this objective, you are supplied two files in CSV format, as described here:

1. **attractions.csv** --- A file with names of places of interest. The user may choose to stop at zero, one or more of the places listed in this file. The file is in CSV format with two columns, in order:
   a. Names of the places of interest (e.g., Statue of Liberty)
   b. Location (city and state; e.g., New York is the name of the city, NY is the abbreviation of the state)

Place of Interest	Location
Statue of Liberty	New York NY
Hollywood Sign	Los Angeles CA
Millennium Park	Chicago IL
NASA Space Center	Houston TX
Desert Botanical Garden	Phoenix AZ
Liberty Bell	Philadelphia PA
The Alamo	San Antonio TX
Balboa Park	San Diego CA
The Sixth Floor Museum	Dallas TX
Winchester Mystery House	San Jose CA
Texas State Capitol	Austin TX
Jacksonville Zoo and Gardens	Jacksonville FL
Fort Worth Stockyards	Fort Worth TX
COSI	Columbus OH
NASCAR Hall of Fame	Charlotte NC

Fig.1 Preview of the file attraction.csv

2. **roads.csv** --- A file with names of cities and the distances between them. The user must choose one starting city and one ending city available in this file. The file is in CSV format with three columns, in order:
   a. CityA (city, state; point/node A)
   b. CityB (city, state; point/node B)
   c. Distance (between A and B; in miles)

CityA	CityB	Distance
New York NY	Los Angeles CA	2800
New York NY	Chicago IL	790
New York NY	Houston TX	1627
New York NY	Phoenix AZ	2400
New York NY	Philadelphia PA	95
New York NY	San Antonio TX	1800
New York NY	San Diego CA	2800
New York NY	Dallas TX	1550
New York NY	San Jose CA	2900
New York NY	Austin TX	1700
New York NY	Jacksonville FL	1100
New York NY	Fort Worth TX	1600
New York NY	Columbus OH	500
New York NY	Charlotte NC	640
Los Angeles CA	Chicago IL	2015
Los Angeles CA	Houston TX	1540
Los Angeles CA	Phoenix AZ	370
Los Angeles CA	Philadelphia PA	2700
Los Angeles CA	San Antonio TX	1370
Los Angeles CA	San Diego CA	120

Fig.2 Preview of the file roads.csv - for instance, row 1 indicates that from New York NY to Los Angeles CA is 2800 miles

Since the road network forms a graph, your program must leverage graph-based shortest path algorithms. In other words, once the user has input a starting city, an ending city, and a set of places of interest, your solution should return the shortest route in the shortest possible algorithmic running time.

In **Task C**, you will be given four extra datasets of different sizes and orders, which may serve as the final database for this project in the future. Your task is to evaluate and analyze the performance of the three algorithms we have learned (i.e., Insertion, Quick, and Merge) on these datasets and report your findings.

Task D requires you to provide a general summary of the entire project, focusing on how you utilized AI-assisted tools for project management, how you implemented the principles of equality, diversity, and inclusion during development, and what you have gained from this project as a developer.

Task E requires you to create a PPT and use it to deliver an oral report of your project in the form of a video. Both PPT and the video (.mp4) are required to be submitted.

Task A - Program Design

Recall the scenario. Since the users will input the starting city, ending city, and the places of interest, the solution must have one function to determine the route with a signature similar to the following:

List<?>route(String startingCity, String endingCity, List<String> attractions)

This function could take the user-input starting city, ending city and list of attractions. (The list of attractions does not have a particular order.) This function returns a list representing the route the user should take.

In the report, you must think and include a description of the type or class contained by the returned list (hence the "?" character in the code above). You should also describe and justify your design by considering the questions from the following aspects:

- **Data Structure**
  - What data structure(s) will you use to represent the data in the file “attractions.csv”?
  - What data structure(s) will you use to represent the data in the file “roads.csv”?
- **Classes and Functions** (Feel free to use UML/code screenshot)
  - What classes did you use?
  - What public or private functions would the classes have?
  - How are these functions/classes implemented, and how do they collaborate/interact in your solution?
- **OOP Principles** (Feel free to use UML/code screenshot)
  - What object-oriented principles (e.g., encapsulation, inheritance, polymorphism, abstraction) have been applied in your overall Java program solution?
  - How are these principles applied specifically? For example, how did you encapsulate user input data to avoid directly manipulating raw data structures?
  - Why are these OOP principles important to your program?

Task B - Algorithm Evaluation: Graph

The implementation of your program must accept one or more queries from the user. The implementation must respond with a path (the optimal route) based on a graph algorithm. An example of the output is as follows (user’s inputs in green):

```
Enter starting city (e.g., New York NY): New York NY
Enter destination city (e.g., Miami FL): Miami FL
Enter attractions (comma-separated, e.g., Hollywood Sign) Hollywood Sign

Result:
Start: New York NY
Destination: Miami FL
Attractions: [Hollywood Sign]
Optimal Route: [New York NY, Philadelphia PA, Columbus OH, Los Angeles CA]
Total Distance: 2445.0 miles
```

Fig 3. Example Output

In the report, you need to provide the output of the following cases with the **total route distance calculated and displayed in the console (see Fig 3)**:

- *Houston TX* to *Philadelphia PA* via ***no*** *place of interest*
- *Philadelphia PA* to *San Antonio TX* via *Hollywood Sign*
- *San Jose CA* to *Phoenix AZ* via *Liberty Bell* and *Millennium Park*

After that, you should also evaluate the implementation of your graph algorithm by considering the following questions:

- What graph algorithm do you use and why?
- How is the algorithm implemented in the program?
- What is the time and space complexity of your chosen algorithm (in Big-O notation)?
- What is the optimality/sub-optimality of your algorithm in terms of its efficiency?
- (In the case where the algorithm is sub-optimal) What can be a better alternative and why?

Task C - Algorithm Evaluation: Sorting

In this task, you need to critically evaluate the performance of different sorting algorithms for potential usability requirements. You are therefore given the following test datasets to evaluate the performance of three sorting algorithms taught in class, namely **Insertion**, **Quick,** and **Merge.**

- 1000places_sorted.csv - contains 1000 alphabetical names of places that are already sorted (A-Z), one per line
- 1000places_random.csv - contains 1000 random alphabetical names of places, one per line
- 10000places_sorted.csv - contains 10000 alphabetical names of places that are already sorted (A-Z), one per line
- 10000places_random.csv - contains 10000 random alphabetical names of places, one per line

In the report, you need to provide the following table summarizing the test results. You can use ***System.nanoTime()*** or ***System.currentTimeMillis()*** to calculate the time.

Datasets	Insertion (ns/ms)	Quick (ns/ms)	Merge (ns/ms)
1000places_sorted			
1000places_random.csv			
...			

Table 1. Template table for sorting algorithm performance

In addition, you need to finish this task by considering the following questions:

- Based on your test data, which of these sorting algorithms does the order of input (i.e., sorted versus random) have an impact on? Why?
- Based on your test data, which of these sorting algorithms does the size of input (i.e., 1000 versus 10000) have an impact on? Why?
- If you were to sort a dataset containing duplicate values, which sorting algorithm would you choose among **Insertion, Quick, and Merge** sort? Why?
- If you were to implement sorting in a system with limited memory (e.g., embedded system), which of these algorithms would you choose? Why?

Task D - Project Reflection

Now that you have coded the project, you are required to provide a critical project reflection section in the report, considering the following aspects:

- **AI-assisted planning and collaboration**
  - How do you use AI tools (such as JIRA and Trello as we mentioned in the class) to help with the planning and collaboration (e.g., task allocation) in the project?
  - What do you think are the advantages and disadvantages of using the AI-empowered software management tools?
- **• Recognition of equality, diversity, and inclusion**
  - Illustrate your understanding of the definition and benefits of applying equality, diversity, and inclusion principles in the current project.
  - Based on these principles, in what areas can the current project be optimized in the future? (e.g., based on the equality principle, one may think of providing features like text-to-speech for visually impaired users and so on).
  - What challenges might arise when applying the improvement above, and how can these challenges be addressed?
- **Conclusion**
  - How did completing this project improve your skills and knowledge as a programmer and software developer?

Task E - Project Presentation

Finally, you are required to create a video explanation with PPT slides to illustrate your project in a succinct manner with the following requirements.

- Briefly introduce the project (e.g., project purpose, key objectives).
- Explain how OOP principles are applied in the project (Encapsulation, Inheritance, etc.).

- Explain how the graph algorithm works to calculate the shortest path using **another test case** that is different from the ones required in the report.
- Considering the size and order differences, explain which types of datasets that **Insertion Sort, Quick Sort, and Merge Sort** are best suited for respectively?
- Conclude your presentation with a proper reflection regarding planning, collaboration, application of equality, diversity, and inclusion principles, and future plans.

Report Requirements

Structure

The purpose of your report is to show how you solved Tasks A, B, C, and D in a well-detailed manner. You need to organize your report in the following format.

- **Coursework submission cover page**
  - (This page is **compulsory**. You can download the template from the CPT204 LMO homepage.)
- **Chapter 1 – Program Design (Task A)**
  - (You may add sub-chapters)
- **Chapter 2 – Graph Algorithm (Task B)**
  - (You may add sub-chapters)
- **Chapter 3 – Sorting Algorithm (Task C)**
  - (You may add sub-chapters)
- **Chapter 4 – Project Reflection (Task D)**
  - (You may add sub-chapters)
- **Chapter 5 – Program Code**
  - Include all your code in Chapter 5 as text, copy and paste each source file content into the report. **The use of screenshots in Chapter 5 will result in an automatic total mark of 0 for the entire coursework task.**
- **Chapter 6 - Contribution Form**
  - Use the following table to indicate your contribution:

Student ID	Contribution (%)

  - **Note:** *The total percentage is 100%. That is, if you and your teammate contribute equally to this project, both of you should fill 50% in the table.*
- **Chapter 7 – Appendix**

- You may use this section to include any supplementary sources (e.g., output of the sorting tests).

Formatting

Write your report using Word with the following settings:

- Font Calibri, Font Size 12, Line Spacing 1.5, Normal Margins.
- The page limit is a maximum of **20** pages, excluding the cover page and Chapters 5-7.
- Consider using images and diagrams to improve the readability of your report.

Coding

- You must only use libraries that are covered in CPT204 (including in Liang textbook). **Violating this by using third-party libraries that are not covered in CPT204 (e.g., JGraphT) will result in an automatic total mark of 0.**
- Plagiarism, copying, collusion, using or consulting unauthorized materials (e.g., copying code from sharing forums and/or generative AI tools, including, but not limited to, ChatGPT/DeepSeek) will not be tolerated and will be dealt with in accordance with the University Code of Practice on Academic Integrity

Project Presentation Requirements

Create a PPT presentation and video explanation (**Task E**) satisfying the following requirements:

- Clearly demonstrate your understanding of the project in terms of program design, algorithm implementation, analysis, and project reflection in a succinct manner.
- You can use any template for your PPT, not limited to the XJLTU standard theme.
- The video must not exceed 8 minutes in length. You must use your own voice for narration and appear on camera (i.e., show your face in the video). **Exceeding the time limit, using English audio translation software for narration, or failing to show your face will result in a score of 0 for Task E.**
- Submit to the LearningMall:
  - A video file in **MP4** format,
  - The PPT file presented in the video.

Submission Requirement

Submit the following files in accordance with the requirements above. Four separate submission portals (links) will be set up for each file on LMO:

- A **ZIP file** containing **ALL** your code files;
- A **Word file** for the report;
- An **MP4 file** for the video recording;
- A **PPT file** that is used in the video recording.

Marking Metrics

Metrics for Project Report (Tasks A-D, 80/100)

Task	Weight	Criteria
Chapter 1 - Task A	20	**Clarity and correctness of the system design**: The report should explain the overall system architecture and its components. **Selection and justification of data structures**: Clearly describe the data structures used to store attractions.csv and roads.csv (e.g., HashMaps, Trees, Graphs) and justify why they are appropriate. **Class structure and relationships**: Explain the designed classes and their interactions using UML diagrams or code snippets. **Application of OOP principles**: Demonstrate what and how OOP principles are applied in the program and why they improve the program.
Chapter 2 - Task B	20	**Correctness and functionality**: The program should accurately compute the shortest path between cities, considering user-specified attractions. **Demonstration of results**: Provide clear console outputs for at least the three test cases specified in the assignment. **Algorithm selection and justification**: Explain why your particular shortest path algorithm(s) were chosen over others. **Efficiency analysis**: Evaluate the algorithm's time complexity and compare it to alternative approaches.

		Optimality evaluation: Discuss whether the algorithm guarantees an optimal solution, or if a more optimal alternative exists. If suboptimal, propose an improvement.
Chapter 3 - Task C	15	**Performance benchmarking**: Conduct and report the sorting experiments on datasets of different sizes and initial orders (i.e., the four dataset provided). Use proper methods such as System.nanoTime() or System.currentTimeMillis() to accurately capture sorting times. **Analysis and justification:** Analyze the performance of the sorting algorithms in different conditions as specified in the questions in the task specification. Justifications are required for the last two questions.
Chapter 4 - Task D	15	**Planning and cooperation**: Provide a clear description and evaluation of the team's planning and cooperation with the help of AI-empowered tools. **Understanding of equality, diversity, and inclusion principles**: Show clear understanding of the definition and benefit of these principles. Discuss how these principle can improve the project as well as the potential challenges involved. **Personal Learning and Future Enhancements**: Thoughtful discussion on personal growth, challenges, and future improvements for the project.
Chapter 5 - Code	10	**Code clarity, organization, and readability:** Includes appropriate variable naming, comments, documentation, and code structure.

Metrics for Project Presentation (Task E, 20/100)

Task	Weight	Criteria
PPT	10	**Logical flow of presentation**: The slides and spoken content should follow a structured progression, covering all major topics smoothly.

		Engagement: The PPT should be engaging, making effective use of visual aids to maintain attention
Video	10	**Fluency & Clarity**: Presenters should speak clearly, avoiding any potential misunderstanding of the slides.

Collaboration and Academic Integrity

This module recognises that interactions with classmates and others can facilitate mastery of the module's materials. However, there remains a line between enlisting the help of another and submitting the work of another. This statement characterises both sides of that line.

The essence of all work submitted to this module for credit must be **entirely your group's own work**. Collaboration on completing coursework with any non-group member is prohibited except to the extent that you may ask classmates and others for some general hints, as opposed to sharing/receiving specific solutions. **All group members are equally and collectively responsible for the entire submission**. **All group members are held jointly accountable for the integrity of the entire submission**. Each group member acts as a guarantor of the group's academic integrity and must actively monitor contributions and violations of academic integrity. All members must confirm their submission and sign the submission cover sheet. If any member violates the university academic integrity policy without the group's immediate reporting, **all members share the responsibilities and will face penalties**.

Plagiarism, copying, collusion, or dishonest use of data will be penalised. Disciplinary actions and mark penalties ranging from capped scores to an award of zero can be applied. Please refer to the University's Academic Integrity Policy on ebridge for guidance. You may also contact module leaders or examination officers if you have any confusion relating to academic integrity.

Please note: Students who would like to submit the same or similar work from previous years to the current module or other modules must receive written permission from all instructors involved in advance of the assignment due date. A student who fails to receive written permission may be penalised according to the University's Academic Integrity Policy.

--- End of CW3 Task Sheet ---