



SCHOOL OF INFOCOMM TECHNOLOGY

Diploma in Information Technology
Diploma in Cybersecurity & Digital Forensic

DATA STRUCTURES & ALGORITHMS

Year 2024/25 - Semester 4

ASSIGNMENT

Duration	: 3 weeks (20 Jan 2025 to 9 Feb 2025) Week 15 to 17		
Weightage	: 40% of total coursework		
Individual/Team	: Team of 2		
Format	: Programming (60%) Program – Basic Features (42%) Program – Advanced Features (18%) Report (20%) Presentation (20%)		

Penalty for late submission: 10 marks per day

NO late submission shall be entertained after Week 18.

There is a total of 7 pages (including this page) in this specification.

WARNING

If a student is found to have submitted work not done by him/her, he/she will not be awarded any marks for this assignment. Disciplinary action will also be taken. Similar action will be taken for the student who allows other student(s) to copy his/her work.

1. OBJECTIVE

This assignment is to test your understanding of the key concepts in the Data Structures and Algorithms module and to evaluate your ability in the selection, design and implementation of appropriate data structures and algorithms in a C++ application.

2. INTRODUCTION

A community of movie enthusiasts wants to create an application about actors, movies and the movies the actors act in. The application should allow an actor with his name and year of birth to be added, a movie with its title, plot and year of release to be added and the actors acting in that movie. The actors and movies information can also be updated by the application administrator when members of the group report any errors.

You are provided with 3 datasets (.csv files) containing information of actors, movies and the movies they act in.

You are to form a team of TWO (2) and implement this movie application using C++.

The basic features of the application should

- (i) Allow the administrator to:
 - a. Add new actor
 - b. Add new movie
 - c. Add an actor to a movie
 - d. Update actor/movie details.
- (ii) Allow the user to:
 - e. Display (in ascending order of age) the actors with age between x and y (inclusive) where x and y are integer values to be entered by the user
 - f. Display movies made within the past 3 years (in ascending order of year)
 - g. Display all movies an actor starred in (in alphabetical order)
 - h. Display all the actors in a particular movie (in alphabetical order)
 - i. Display a list of all actors that a particular actor knows.

For example, if actor A1 starred in movie M1 and actor A2 also starred in movie M1, then actor A1 knows actor A2. This can extend for one more level and no further i.e., if actor A2 also starred in movie M2 and actor A3 acted in movie M2, then actor A1 also knows A3.

A text driven user interface is adequate for the application for now. However, it must be user friendly.

Note 1: You are **NOT** to make use of the Standard Template Libraries (STL) from the C++ standard library for the implementation of your data structures and operations.

3. BASIC REQUIREMENTS

The application must:

- Make use of at least two **data structures**
- Demonstrate the **application of algorithms** learnt (e.g. searching)

The data structures and algorithms chosen should be appropriate for your application.

The suitability and complexity of the implemented data structures and algorithms and the user friendliness of the application will determine how well you score in this section.

4. ADDITIONAL FEATURES

In addition to the basic requirements outlined above, you may implement **additional** features.

Examples of additional features:

- Allow for capturing of actor rating and movie rating.
- Allow recommendations based on rating of actor/movie.
- Data structures that can help improve performance and/or storage.
- Any other useful enhancements.

Again, the suitability and complexity of your chosen data structures and algorithms and the user friendliness in the implementation of your additional features will determine how well you score in this section.

Only 2 features are necessary for maximum marks.

5. DELIVERABLES

- a. A **report** in Microsoft Word document format that clearly indicates:
- i. Team members' names, student IDs and group name
 - ii. A brief description of the application
 - iii. Roles and contributions of each member in the group
 - iv. **Instructions** on how to run your application
 - v. Class Diagram
 - vi. Description of data structures and algorithms implemented
 - vii. Detailed explanation of why the data structures and algorithms are selected.
 - viii. All relevant appendices (diagrams, screenshots, user guides, etc.) wherever appropriate
 - ix. References for any non-DSA materials used in the report and/or application
- b. Folder containing all the necessary files (solution, .h, .cpp, .csv) to run your application.

Note 2: You are required to upload a zipped file to Brightspace: Assessment>Assignment>. Leaders are to upload the file to their respective folder. E.g. Name1_Name2.zip

Note 3: A penalty of 10 marks per day after **9 February 2025, 2359 hrs** will be applied for late submission.

- c. A 20-minute presentation that has the following:
- Introduction
 - Description of application and its features
 - Demo of Application
 - Explanation of selection, design and implementation of data structures and algorithms used
 - Q&A session

6. MARKING SCHEME

Assignment (40% of module):

Report (20 marks)

- Clarity and comprehensiveness of the document
- Explain and analyse clearly and comprehensively the selection and implementation of the data structures and algorithms

Program (60 marks)

- Basic Requirements (70%)
 - Section 3 of this document
 - Proper documentation and programming style
 - A blocked comment at the top of the class stating Team members' name, group, student ID and any features that you would like to highlight specifically.
 - A blocked comment at the top of each function, giving brief description of what the function does, including input parameters and return value, if any.
 - Good programming practices (indentation, good function/variable/attribute naming, etc.).
- Additional Features (30%)
 - Section 4 of this document
 - Documentation and programming style as above

Note 4: Unfamiliarity with any part of the program will cause penalties that will compromise the overall assignment grade for a student.

Presentation (20 marks)

- Give good introduction of the application.
- Give detailed description of the various functionalities of the application.
- Explain clearly and comprehensively the choice of design and implementation of the data structures and algorithms.
- Able to provide good answers in response to tutor's questions.

The student must be able to demonstrate understanding of the concepts of data structures and algorithms learnt, apply the understanding in the implementation of the application. Any demonstration of poor understanding during presentation will cause penalty to the program equivalent.

7. PERFORMANCE CRITERIA

Performance Criteria for grading the assignment is as described below. Marks awarded will be based on the **program** as well as student's degree of understanding of work done and DSA concepts as assessed during the **demo**.

A Grade

- Program implements the Basic Requirements successfully
- Program implements the Advanced Requirements successfully
- Program demonstrates very good understanding and application of data structures and algorithms.
- Program demonstrates very good design and documentation.
- Program has been tested well.
- Report has good clarity and comprehensiveness, able to describe and explain well the overall design, analysis and application of data structures and algorithms with good understanding of efficiency considerations.
- Very good demonstration of program and showing excellent understanding of work done during the demo

B Grade

- Program implements the Basic Requirements successfully
- Program implements some of the Advanced Requirements successfully
- Program demonstrates good understanding and application of data structures and algorithms.
- Program demonstrates good design and documentation.
- Program has been tested adequately.
- Report has relatively satisfactory amount of clarity and comprehensiveness, able to describe and explain satisfactorily the overall design, analysis and application of data structures and algorithms with some understanding of efficiency considerations.
- Good demonstration of program and showing good understanding of work done during the demo

C Grade

- Program implements the Basic Requirements successfully
 - Program demonstrates some understanding and application of data structures and algorithms.
 - Program demonstrates some amount of design and documentation.
 - Program has been tested adequately.
 - Report describes with some explanation, the overall design, analysis and application of data structures and algorithms with some understanding of efficiency considerations.
 - Some amount of demonstration of the program and showing some understanding of work done during the demo
-

D Grade

- ◆ Program implements some of the Basic Requirements successfully
- ◆ Program with some design and documentation.
- ◆ Program has been tested to some degree
- ◆ Report describes with some explanation on the design, application of data structures and algorithms,
- ◆ Able to answer some questions during the demo

-- End of Document --