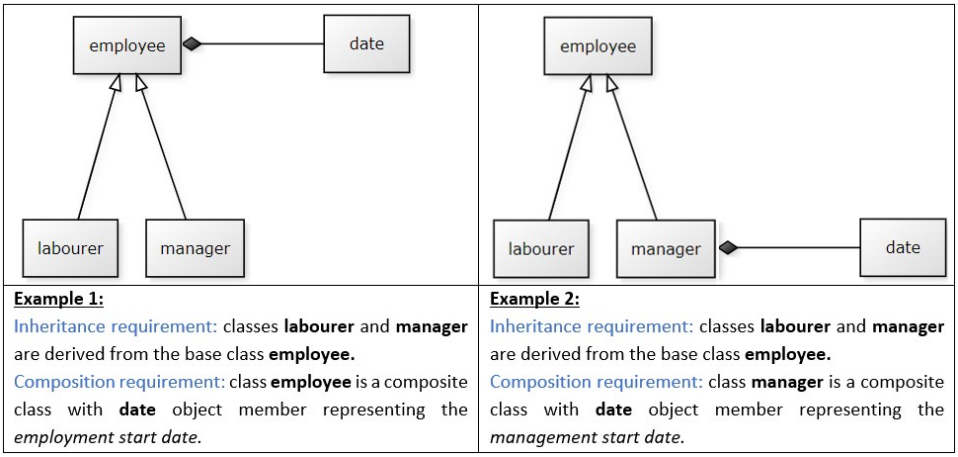
Data Structures and Algorithms Project Proposal

Course projects are necessary to ensure that students have a better understanding of the material presented in the course.

In this project, which is to be completed in **teams of three students**, the students will select a specific concept to implement in C++. This implementation will cover the application of various data structures taught in the course as well as searching and sorting algorithms.

The project is split into 3 parts, as described below:

1. **Project Proposal**: A 1–2-page document that describes the concept the teams have selected, as well as a brief break down of components they wish to implement displayed in a UML. The system must include at least 4 classes which are related using composition and inheritance. Here are two examples using the classes you have seen in this course:



Example 1:	**Example 2:**
Inheritance requirement: classes **labourer** and **manager** are derived from the base class **employee.** Composition requirement: class **employee** is a composite class with **date** object member representing the *employment start date.*	Inheritance requirement: classes **labourer** and **manager** are derived from the base class **employee.** Composition requirement: class **manager** is a composite class with **date** object member representing the *management start date.*

2. **Implementation**: The data generated are to be stores using `linked lists`. The essential functionalities of the implementation are:

   - `Store:` to add items to the data structure,
   - `Retrieve:` to search for a specific item in the data structure. A searching algorithm is required for efficient searching and retrieving the data records,
   - `Sort:` to sort the existing items in the data structure.
   - `Delete:` to remove a record from the data structure,
   - `Count:` to return the number of records in the data structure,
   - `Empty:` to check for empty data structure,
   - `Clear:` to empty the data structure.

3. **Project Report**: A 4-5-page document describing the design process of the project. Ideally one should be able to rebuild the implementation by reading through the documentation of the design. The teams could use UML or flow diagram to detail their designs. The report should also document tests cases completed to prove that the system utilizes the desired functionalities. Lastly, the report should also comment on memory leaks in the system or special data input cases leading to the system failure, if any.

Project Dates:

Project proposal (max 2 pages): Mar 05, 2024
Project presentation: Apr 05, 2024 (tentative)
Project report (max 5 pages): Apr 10, 2024 (tentative)

Project Grading Scheme:

20% project proposal + 50% project presentation + 30% project report