

CSE 240 Spring 2021 Homework 9:

C++ Inheritance and Polymorphism (50 points)

Due Saturday, March 27, 2021 at 11:59PM, plus a 24-Hour grace period

Introduction

The aim of this assignment is to make sure that you understand and are familiar with the concepts covered in the lectures, including C++ inheritance, polymorphism, virtual function, containment, file operations, and exception handling. By the end of the assignment, you should have understood

- Class, data members, function members, virtual function, constructor, and destructor
- Accessing privileges among the classes
- Memory management and deletion of linked list
- Scope Resolution Operators
- Inheritance and containment
- Dynamic binding and polymorphism
- File operations
- Exception handling

Reading: Textbook Chapter 3, Sections 3.2 through 3.5. Read Section 3.2.1 on friend access. Read the program example in Section 3.4.2.

Preparation: Complete the multiple choice questions in the textbook exercise section. The answer keys can be found in the course Web site. These exercises can help you prepare for your weekly quiz and the exam. You are encouraged to read the other exercise questions and make sure that you understand these questions in the textbook exercise section, which can help you better understand what materials are expected to understand after the lectures and homework on each chapter.

You are expected to do the majority of the assignment outside the class meetings. Should you need assistance, or have questions about the assignment, please contact the instructor or the TA during their office hours.

You are encouraged to ask and answer questions on the course discussion board. However, **do not share your answers and code** in the course discussion board.

Programming Exercise (50 points)

You are given a partially completed project containing:

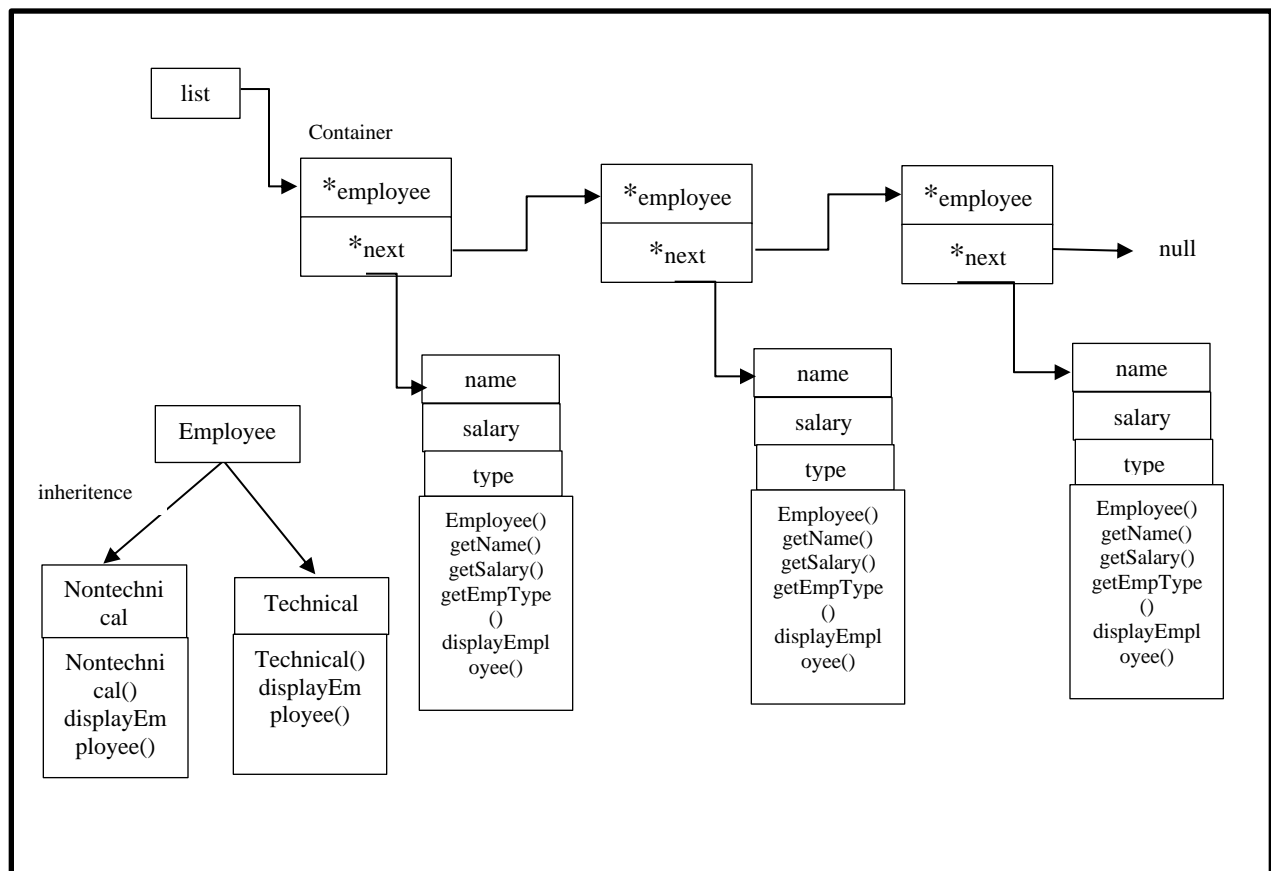
4 header files:

Container.h
employee.h
nontechnical.h
technical.h

5 C++ files:

Container.cpp
employee.cpp
nontechnical.cpp
technical.cpp
hw9.cpp

The diagram shows the inheritance and relationships between the classes and the linked list formed.



Employee class is defined in employee.h and member functions are defined in employee.cpp.

Q1: Create Nontechnical and Technical child classes of Employee class (Inheritance) (10 points)

Files: nontechnical.h , technical.h

Q2: Define displayEmployee() for child classes (Polymorphism) (10 points)

Files: nontechnical.cpp , technical.cpp

Q3: Add Friend function changeSalary() (5 points)

Q3.a in employee.h

Q3.b in hw9.cpp

Q3.c in hw9.cpp. Case 'c' of executeAction()

Q4 - Q7 are used to implement a menu-driven program. Its implementation is in hw9.cpp. The menu gives the following options to user:

- a) Add a new employee to the list. You do not need to check if the employee exists in the list because that is already implemented in executeAction(). You may add the new employee to head or tail of the list. (sample solution adds it to the tail)
- b) Display the list of employees along with their details (name, salary, type).
- c) Change the salary of an employee. This option uses the friend function to access private data member 'salary' of the employee and change it to the one entered by the user.

You need to implement save() that writes the list of books to a file 'list.txt' while exiting the program and load() that reads the file to form the list at the beginning of the program. During first execution of the program, there will be no 'list.txt', so load() would not form a list. Once list.txt is saved, load() can read it the next time you run the program. You may remove the list.txt from your project directory if you do not want to load that list (maybe while testing your code).

The following is one way to store the list in list.txt file. The format here is:

```
<no. of employees>
<Employee1 name>
<Employee1 salary >
<Employee1 type>
<Employee2 name>
<Employee2 salary >
<Employee2 type>
and so on..
```

You may store it in another format if you wish. Notice that the '3' at the beginning is the number of employees stored in the file.

 list.txt - Notepad

File Edit Format View Help

```
2
a
1000
0
b
3000
1
```

Expected output:

addEmployee():

```
CSE240 HW9
Please enter your selection:
    a: add a new employee
    d: display employee list
    c: change salary of an employee
    q: quit
a

Enter employee name: a
Enter new salary: 1000
Enter employee type:
0. Nontechnical
1. Technical
0

Employee added to list!
```

displayList():

```
CSE240 HW9
Please enter your selection:
    a: add a new employee
    d: display employee list
    c: change salary of an employee
    q: quit

d

Employee name: a
Salary: 1000
Employee Type: Nontechnical

Employee name: b
Salary: 2000
Employee Type: Technical
```

changeSalary():

```
CSE240 HW9
Please enter your selection:
    a: add a new employee
    d: display employee list
    c: change salary of an employee
    q: quit

c

Enter employee name: b

Enter new salary: 3000

Salary changed!
```

Verify by using option d.

Grading of Programming Assignment

The TA will grade your program following these steps:

(1) Compile the code. If it does not compile, 20% of the points given will be deducted. For example, if there are 20 points possible, you will earn 16 points if the program fails to compile.

(2) There is a 10% deduction for not commenting your code. As a thumb rule, there should be at least 2 useful comments in each function.

(3) The grader will read your program and give points based on the points allocated to each component, the readability of your code (organization of the code and comments), logic, inclusion of the required functions, and correctness of the implementation of each function.

What to Submit?

You are required to submit your solutions in a compressed format (.zip). Zip all files into a single zip file. Make sure your compressed file is labeled correctly: lastname_firstname9.zip.

The compressed file MUST contain the following:

4 header files:

Container.h	(unchanged)
employee.h	(completed code)
nontechnical.h	(completed code)
technical.h	(completed code)

5 C++ files:

Container.cpp	(unchanged)
employee.cpp	(unchanged)
nontechnical.cpp	(completed code)
technical.cpp	(completed code)
hw9.cpp	(completed code)

No other files should be in the compressed folder.

After submitting, download your own submission to verify if the files are correct.

If multiple submissions are made, the most recent submission will be graded, even if the assignment is submitted late.

Where to Submit?

All submissions must be electronically submitted to the respected homework link in the course web page where you downloaded the assignment.

Late submission deduction policy

- No penalty for late submissions that are received within 24 hours after the deadline;
- 10% grade deduction for every day it is late after the grace period;
- No late submission after Tuesday at 11:59PM.

Academic Integrity and Honor Code.

You are encouraged to cooperate in study group on learning the course materials. However, you may not cooperate on preparing the individual assignments. Anything that you turn in must be your own work: You must write up your own solution with your own understanding. If you use an idea that is found in a book or from other sources, or that was developed by someone else or jointly with some group, make sure you acknowledge the source and/or the names of the persons in the write-up for each problem. When you help your peers, you should never show your work to them. All assignment questions must be asked in the course discussion board. Asking assignment questions or making your assignment available in the public websites before the assignment due will be considered cheating.

The instructor and the TA will CAREFULLY check any possible proliferation or plagiarism. We will use the document/program comparison tools like MOSS (Measure Of Software Similarity: <http://moss.stanford.edu/>) to check any assignment that you submitted for grading. The Ira A. Fulton Schools of Engineering expect all students to adhere to ASU's policy on Academic Dishonesty. These policies can be found in the Code of Student Conduct:

http://www.asu.edu/studentaffairs/studentlife/judicial/academic_integrity.htm

ALL cases of cheating or plagiarism will be handed to the Dean's office. Penalties include a failing grade in the class, a note on your official transcript that shows you were punished for cheating, suspension, expulsion and revocation of already awarded degrees.