This assignment consists of implementing the classes and the class relationships represented in a UML class diagram, instantiating objects from the classes, pushing data into the objects with constructors, and pulling the data back out using operators and other functions. This lab only explores inheritance (chap 9). The next lab introduces additional relationships, some of which rely on pointers and dynamic memory (chap 10).

Assignment

Write a C++ program consisting of the four related classes presented in the UML diagram below. Complete the program with the provided driver program (no modification required).

1. Make all member variables (i.e., attributes) private.
2. Make all member functions (i.e., behaviors, operations, or services) public.
3. Initialize all member variables via constructor calls.
4. Define each class in a separate .h file (remember the #ifndef and #endif preprocessor statements)

```
Employee
    -name : string
+Employee(in a_name : string)
+operator<<(in out : ostream&, in me : Employee&) : ostream&

WagedEmployee
    -wage : double
    -hours : double
+WagedEmployee(in name : string, in a_wage : double, in a_hours : double)
+operator<<(in out : ostream&, in me : WagedEmployee&) : ostream&

SalariedEmployee
    -salary : double
+SalariedEmployee(in name : string, in a_salary : double)
+operator<<(in out : ostream&, in me : SalariedEmployee&) : ostream&

SalesEmployee
    -commission : double
    -sales : double
+SalesEmployee(in name : string, in salary : double, in a_commission : double, in a_sales : double)
+operator<<(in out : ostream&, in me : SalesEmployee&) : ostream&
```
5. The SalesEmployee constructor
   a. uses a_commission and a_sales arguments to initialize the commission and sales member variables respectively
   b. calls the SalariedEmployee constructor and passes to it name and salary - remember that this is a constructor call that takes two arguments

6. The SalariedEmployee constructor
   a. uses a_salary to initialize the salary member variable
   b. calls the Employee constructor and passes to it name

7. The WagedEmployee constructor
   a. uses a_wage and a_hours to initialize the wage and hours member variables respectively
   b. calls the Employee constructor and passes to it name

8. The Employee constructor
   a. should initialize the name member variable

9. The SalesEmployee operator<<
   a. should call the SalariedEmployee operator<< to print the salary
   b. should print the commission and sales

10. The SalariedEmployee operator<<
    a. should call the Employee operator<< to print the name
    b. should print the salary

11. The WagedEmployee operator<<
    a. should call the Employee operator<< to print the name
    b. should print the wage and hours

12. The Employee operator<<
    a. should print the name

Program Submission

- Upload the four header files (employee.h, salariedemployee.h, wagedemployee.h, and salesemployee.h) to BlackBoard (make a new directory and upload the files to that directory)
- Make sure that the files are correctly named
- Remember to press the “Submit” button
- Please do not zip the files
- Retain a copy of your code for use in the next assignment