Last homework: Inheritance and polymorphism

Objective: The main goal of this last homework is to implement a simple inheritance hierarchy between the Person and Employee class.

Special instruction: This homework will be credited for a total of 2 bonus points. The graders will post your grades on a scale of 100 points, which will be scaled down to 2 bonus points when calculating your final letter grade.

Description. Declare and implement the Employee class as a derived class of the Person class, which is already implemented in the demo code posted under this week (week #15). Note that your project doesn't need to include the Student class. Please also feel free to modify the Person class. For instance, you can replace the data type for birthday to string.

Specifically, your project will include Person as the base class, and Employee as the derived class of Person. Your Employee class will include the following data members:

year of employment (e.g., 2012)

base salary

a dynamic array to keep track of the annual salary increase rate since the year of employment. For instance, an employee joined the company in year 2013, his salary increase rates in 2014, 2015, and 2016 were 10%, 7%, and 11%, respectively. This dynamic array will then hold the following three numbers: 0.1, 0.07, and 0.11. Furthermore, let bSal be the base salary, the current salary of the above employee should be equal to: bSal\*1.10\*1.07\*1.11. Note that although the size of this array is derivable from the year of employment, you are recommended to include two additional data members to manage the capacity and size of this array. (This is similar to what you did in the previous homework.)

At the minimum, your Employee class needs to include the following function members:

a default constructor

the big-3: destructor, copy constructor and overloaded assignment operator

an accessor to return the current salary of an employee

an accessor and mutator for each of the above data members. It's sufficient to return the salary increase rate in a specific year.

an overloaded put operator (<<) to print out all the information in an Employee object

override the output() function in the base class, as this function is virtual in Person.

Please provide a simple character-based interface to allow the graders to test each of the above member functions. Make sure you test the output() function using Person pointers and observe polymorphism in action.

Submission requirements: Please use separate compilation and submit your source code in one zip files. Your project will include two header files (one for the Person class, the other for the Employee class) and three .cpp files.