Assignment #3: Due 5:00 PM on Wednesday, March 9 at 5:00 PM, via email

A Basic Inheritance Hierarchy with Polymorphism (from the CSC 123 textbook, pages 621-622)

Define an Employee class. An Employee has a name, an ID number, an age, a salary, a title, and a department name. The methods of Employee should:

a. Print an employee record that includes all of the above information
b. Change a salary, changeSalary(), and
c. return the salary, getSalary().

The method changeSalary() accepts a parameter, *increase*, of type int or double. If increase is an int, then the salary should be increased by that amount. If increase is double, then the new salary should be (increase + 1) times the salary. For example, if the increase is 0.10, the salary is multiplied by 1.10, yielding an increase of 10%. The value of the (double) increase should be between 0.0 and 1.0.

Define a class Manager that Extends Employee. A manager is an employee who supervises other employees. A Manager object should include all data of the Employee object plus the list of the employee ID numbers of those employees under his/her supervision as well as all the other relevant data.

Define a class Executive that extends Manager. An Executive is a Manager who receives a bonus a the end of each year equal to percentage of his/her regular salary. Each Executive has his/her own bonus rate. You will need to redefine getSalary() to include the bonus. You will also need to add a setter (mutator) method, setBonus(), to set the percentage of the executive's bonus. The default bonus rate should be 10%.

Implement a test class that demonstrates the facilities of the Employee, Manager, and Executive classes. Your test class should accept employee information for an arbitrary number of employees. Your program should ask whether or not the employee is a manager or an executive, and prompt for all relevant information.

After all data are entered, print an error message if there are any inconsistencies. In particular, a manager cannot manage a nonexistent employee. Also, every employee who is not an executive is supervised by some manager or executive.

Your program should provide the user with the following options:

- Change the salary of an employee
- Adjust the bonus of an executive
- Add or delete an employee from a manager's list of employees
- Print an employee's data

If any change causes an inconsistency in the data, your program should print an error and not allow the change.

Your program should access an employee via the employee ID number. ~~Use a binary search to find an employee's record.~~ Just store in an array; the binary search aspect will be used as part of a future exercise.