Lab 3

Direction: Submit the typed source code or git url. All tasks must be completed. Each team member is required to do at least 1 task. And no group can exceed 3 members

Employee

For this lab, you will define the three classes **Name**, **Address** and **Employee**.
Your group will have to rewrite the given method so that they perform their descriptions correctly.

I. The header file "Name.h" contains the class **Name**, which consists of

- □ Private string field named *firstName*.
- □ Private string field named *lastName*.
- □ Public default constructor that assigns the empty string to both *firstName* and *lastName*.
- □ Public overloaded constructor that takes two strings as parameters named *firstName* and *lastName* respectively. If the parameters are valid names (they consist of only letters), they should be assigned to their respective fields (match the names); however, if any parameter is invalid, assign its respective field the empty string.
- □ Public copy constructor that performs a shallow copy.
- □ Public overloaded assignment operator that performs a shallow copy.
- □ Public get method for *firstName*.
- □ Public get method for *lastName*.
- □ Public set method for *firstName* that assigns the parameter to *firstName* only if it is a valid name.
- □ Public set method for *lastName* that assigns the parameter to *lastName* only if it is a valid name.
- □ Public string constant method named `ToString()` that takes no parameters. It returns a string with the format *firstName* followed by *lastName* with a space between them.
- □ Public overloaded ostream operator that the displays a Name object in the same format as `ToString()`.

II. The header file "Address.h" contains the class **Address**, which consists of

- □ Private string field named *street*.
- □ Private string field named *city*.
- □ Private string field named *state*.
- □ Private string field named *zipcode*.
- □ Public default constructor that assigns the empty string to both *street* and *city*,"NY" to *state* and "11111" to *zipcode*.
- □ Public overloaded constructor that takes four strings as parameters named *street*, *city*, *state* and *zipcode* respectively. It assigns *street* to the *street* field. It assigns *city* to the *city* field only if *city* is a valid name (consists of only letters); otherwise, it assigns the empty string to the *city* field. It assigns *state* to the *state* field only if *state* is a valid initial (consists of only two letters); otherwise, it assigns "NY" to the *state* field. And it assigns *zipcode* to the *zipcode* field only if *zipcode* is a valid zipcode (consists of exactly five digits); otherwise, it assigns "11111" to the *zipcode* field.
- □ Public copy constructor that performs a shallow copy.
- □ Public overloaded assignment operator that performs a shallow copy.
- □ Public get method for *street*.
- □ Public get method for *city*.
- □ Public get method for *state*.
- □ Public get method for *zipcode*.
- □ Public set method for *street*.

- □ Public set method for *city* that assigns the parameter to *city* only if it is a valid name.
- □ Public set method for *state* that assigns the parameter to *state* only if it is a valid initial.
- □ Public set method for *zipcode* that assigns the parameter to *zipcode* only if it is a valid zipcode.
- □ Public string constant method named `ToString()` that takes no parameters. It returns a string with the format *street* on its own line, followed by *city* comma *state* space *zipcode*.
- □ Public overloaded ostream operator that the displays a Name object in the same format as `ToString()`.

III. The header file "Employee.h" contains the class **Employee**, which consists of

- □ Private **Name** field named *name*.
- □ Private **Address** field named *address*.
- □ Private int field named *employeeId*.
- □ Private double field named *salary*.
- □ Private int static field named *nextId* which is initialized to 1.
- □ Public default constructor that assigns the default **Name** object to *name*, default **Address** object to *address*, *nextId* to *employeeId* and 5000 to *salary*. It also increments *nextId* by 1.
- □ Public overloaded constructor that takes a **Name**, **Address** and a double as parameters named *name*, *address* and *salary* respectively. It assigns *name* to the *name* field. It assigns *address* to the *address* field. It assigns *salary* to the *salary* field only if *salary* is at least 5000; otherwise, it assigns 5000 to the *salary* field. And it assigns *nextId* to *employeeId*. It also increments *nextId* by 1.
- □ Public copy constructor that performs a shallow copy.
- □ Public overloaded assignment operator that performs a shallow copy.
- □ Public get method for *name*.
- □ Public get method for *address*.
- □ Public get method for *employeeId*.
- □ Public get method for *salary*.
- □ Public static get method for *nextId*.
- □ Public set method for *name*.
- □ Public set method for *address*.
- □ Public set method for *salary* that assigns the parameter to *salary* only if it is at least 5000.
- □ Public string constant method named `ToString()` that takes no parameters. It returns a string with the format *name* space *employeeId* on their own line, followed by *address* on its own line, followed *salary*.
- □ Public overloaded ostream operator that the displays a Name object in the same format as `ToString()`.