This assignment involves writing a program that models information for a company’s employees. It will utilize the following classes: an Employee class, a Name class, an Address class, and a Date class. These classes should model the following information:

Date Class

* Should model a date as an integer month, integer day, and integer year. You may use the Date class from one of the lectures or define your own.

Name Class

* Should model a person’s name using a String for first name and a String for last name.

Address Class

* Should model an address as a street (e.g., 123 Main Street), a city (e.g., San Francisco), a state (e.g., New York or simply a state code like NY), and a zip code (e.g., 11550). These attributes can all be String types.

Employee Class

* Employee attributes include an integer employee id number, an employee name (a Name object), an employee address (an Address object), and an employee hire date (a Date object).

You must provide appropriate class constructors, any appropriate getter, setter methods, and other methods that you think are appropriate.

Program Behavior

Write a program that does the following:

* Prompts the user to enter the number of employees to handle.
* Prompts the user to enter the above employee information.
* Displays the employee information.
* A sample program run is provided below. Your program dialogue doesn’t have to look exactly the same, but it should look similar.

Design Constraints & Hints

* Your program must store the employee data in an array of type Employee. (Hint: we discussed String arrays in an earlier lecture. Arrays of programmer-defined objects work the same way.)
* You must use a static method called from main() to display the employee data. The method shall accept an array of type Employee as its argument.
* All user input must be prompted from main()…not from any of the application classes.
* You must use a Scanner to get user input. Use nextInt() to input integer types and nextLine() to enter String types. When Scanner is used to get keyboard input consisting of numeric and string types, things get a little tricky. If you prompt the user to input an employee ID (an integer), and then prompt for first name, last name, etc. (all String types), you have to clear the input buffer before a user enters the first String object (I’ll discuss why in class). This is done by using a statement like input.nextLine().
* The code for each class must be compiled in separate source files…one file per class.

Assignment Deliverables

* Source code (i.e., .java files) for each class used in your solution. Do not include any IDE-specific project files.
* A screen shot of your program input/output similar to the one below.
* Submit source files and screen shot in a single zip file.

Sample Screen Shot

A computer screen shot of a black and white computer screen

AI-generated content may be incorrect.