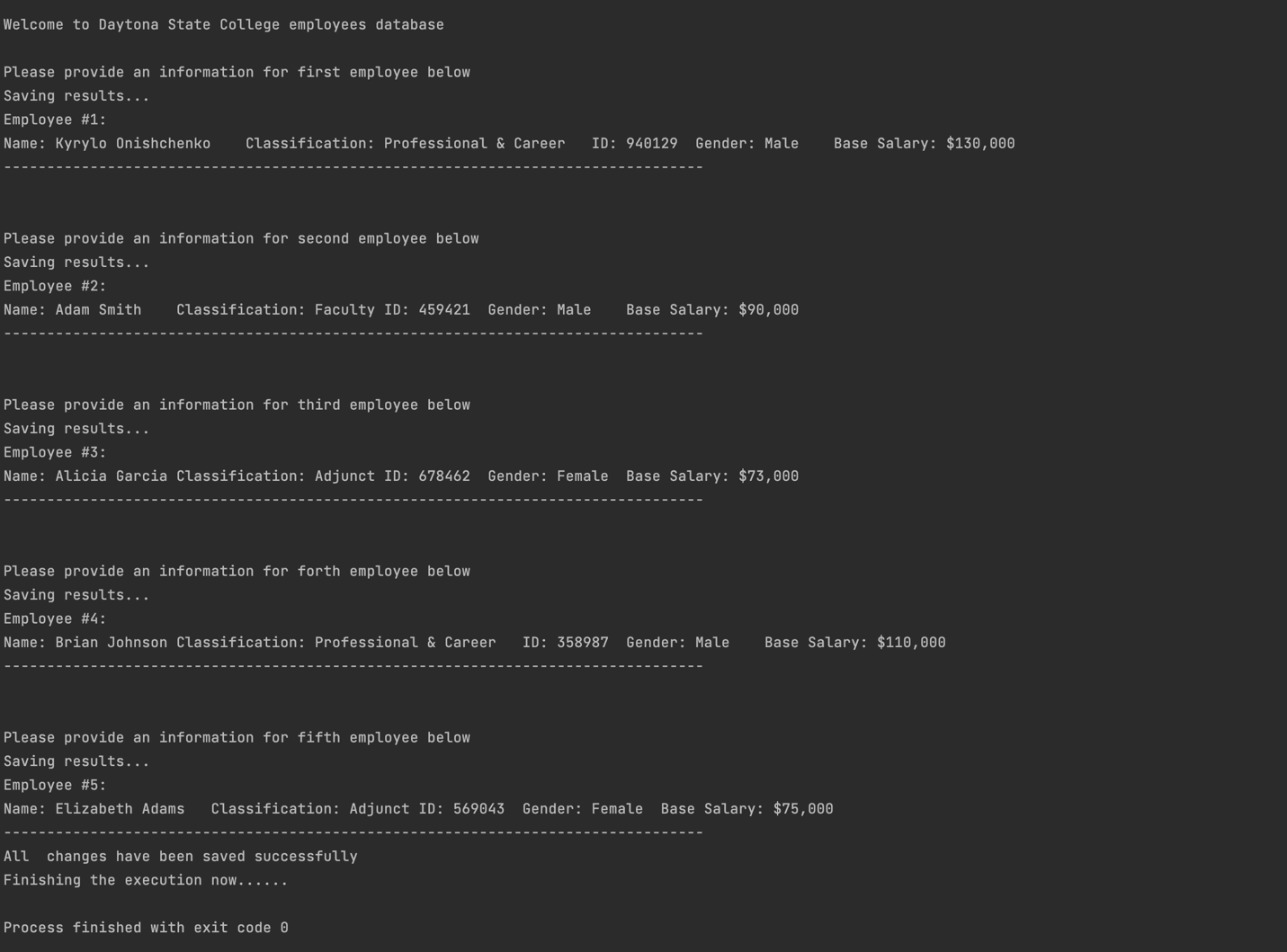
**Assignment 3.1 (a,b):**



In this assignment I have learned the basics of object-oriented programming which includes creating a class assigning private variables, creating a class’ constructor, one that does not take any parameters, as well as one that takes multiple variables as parameters which then gets assigned to those private variables at the beginning of the class itself. Furthermore, I have learned the difference between having ***void*** keywords in a method declaration (a.k.a the methods that do not return any value) as opposed to value returning method. Also, I have learned about keyword ***static*** in a method declaration, which allows calling a method without first creating an object.

**Assignment 3.2:**

1. **Write in your own word, your understanding of a “class” and an “object”. Give example.** 
   * As far as my own understanding goes, a “class” is a container that consists of a constructor, methods, and variables with assigned attributes, which can be used to establish an “object”. An object is an instance of a class on which you can invoke different methods established during class creation. For instance, let’s take “Employee” as an example. An employee may have multiple attributes, (specific and unique) traits/abilities such as name, classification, id, salary, and gender. Those attributes are unique to each specific newly established class instance (a.k.a object). In addition, a class may have multiple methods, those are specific blocks of instruction that may return, set, or compute any specific operation. In our example, methods, such as ***getEmployeeName()*** and ***setEmployeeName()*** return and assign the specific value to a name respectively.
2. **Why would you create a class and an object (what are their advantages)? Give example.** 
   * I prefer to think of classes as a composition of characteristics that are assigned to an object. Thus, it allows you to save time by not recreating the same or slightly different code for every object but rather assigning the same methods to different objects, therefore saving a lot of time and improving the maintainability of the program itself. Objects allow you to manipulate code in order to perform a specific set of functions that have been assigned within a class to that object. For example, let’s consider a basic wallet application that allows you to manually track your expenses by inputting every expense and calculating the total amount of money spent. Imagine writing the code for tracking the current amount and subtracting it every time the user inputs the number. But what if you want to track both income and expenses from multiple bank accounts? That is the problem if you do not use classes and assign objects to have access to the same methods for multiple accounts since it leaves you with the only option of writing repetitive code for every account tracking, then writing the same code twice for subtracting and adding amounts for both accounts. Thus, programming becomes a burden, and your program is unmaintainable. Object and classes both allow you to save production time, maintain your code better, and improve the scalability of the application by the ability to implement new features in the future.