



The last position in the array is no longer needed. The last position of the array is released (using `unset`). This will inform the operating system that the space in memory is no longer needed. The operating system will call its garbage collector to free up the memory location. We will discover an easier way to accomplish this task when we discuss associative arrays in a later chapter. Once the array has been reconfigured, the `saveChanges` method is called to replace the records in the log file. The code shown is very similar to previous examples in this chapter, with one exception. The `fopen` method uses the parameter “w”. The “w” parameter will erase anything that is already in the file and replace it with what is currently being written to the file. In this example the file will be updated (replaced) with the new set of records that excludes the record that was deleted. The `displayRecords` method is called anytime the program is called (with or without a record being deleted). This method displays the contents of the log file.

*Programming note—Programs that retrieve data that will be used throughout the program usually retrieve the information in the initial stages of the code and place it in a data structure (array, list, or dataset). As the data is updated in the program, the information in the data structure is updated. When the data processing has been completed, the information is then returned to the original location (text file or database). Updating the text file or database is usually completed as one of the final stages of the program. The process of a user logging out of a program would provide an event to indicate that the updated data should be saved.*