Review Test Submission: Midterm Test Part 3 (Coding)

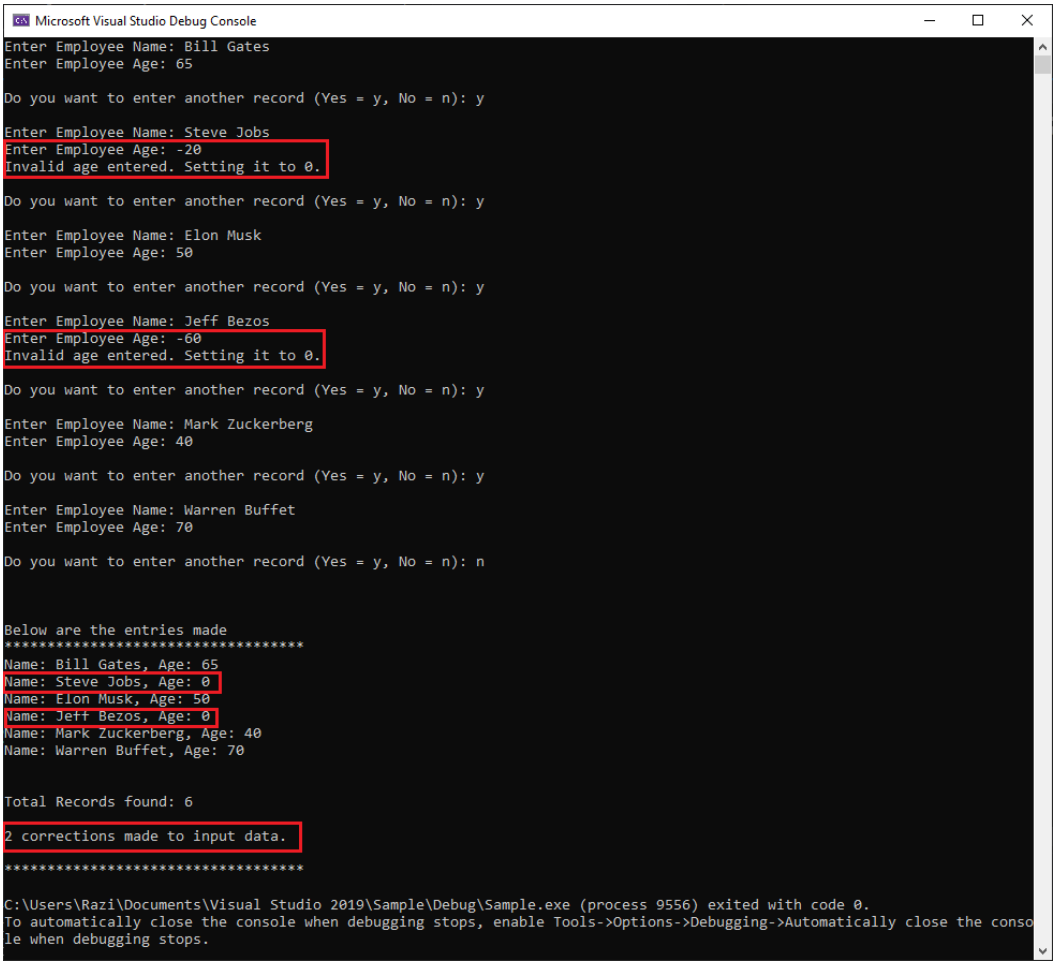
User	Arzu Gizem Kirici
Course	Introduction to Object Oriented Programming
Test	Midterm Test Part 3 (Coding)
Started	10/17/22 3:05 PM
Submitted	10/17/22 4:05 PM LATE
Due Date	10/17/22 4:05 PM
Status	Completed
Attempt Score	20 out of 40 points
Time Elapsed	59 minutes out of 1 hour
Results Displayed	Feedback

Question 1

20 out of 40 points

Write a C++ program that keeps taking input from the user about an Employee and stores them in the heap and then displays it. You are required to perform the following tasks:

a. Create an Employee class with the following data members:
   a. `char * name`
   b. `int age`
b. Make sure to have all the appropriate getters and setters along with a `void toString()` method that can display all the employee information in the form shown in the expected output screenshot.
c. At the start of the program, it should ask user to enter Employee Name and Age. When user is done entering the information, the program should ask whether user wants to enter more employee information or not? If user presses 'y' or 'Y', the program should ask user to enter Employee Name and age for another employee, otherwise it should quit while displaying showing the report.
d. Make sure to store the employee information in an array of employee and store it inside the heap memory on each employee entry from the user. You should start with the heap memory size of 0 and on each entry of employee name and age, an increment of heap memory size should happen so that on-demand heap memory is created for every new employee.
e. Your program should have a resize function `Employee* resize(Employee* arr, int* size)` to resize the employee array in the heap on every entry of employee information from the user.
f. Your program should have a display function `void display(Employee* arr, int size)` to display all the information by calling `toString()` method of employee class for all the objects in the employee array.
g. Make sure to validate the input for age. For example, if user enters an invalid age (e.g., < 0), then user program should default age to 0 and inform user that age is set to 0 because of invalid entry.
h. In display method in point (e), you should also be able to display, how many entries were corrected by the program due to invalid entries from the user. Please see highlighted parts of the output screenshot below for more information.
i. Follow the modular programming approach to organize your code and dont forget to indent and comment your code.
j. Below is the screenshot of the expected output of the program:



Response Feedback: You were required to use char * for name and not string.

resize is not accurate

← OK