

Homework-1

Objective:

Write a Java program to store user input and then display it on the console.

Instructions:

1. Accept and store user input for the following fields:
 - First Name
 - Last Name
 - Age (in years) [eg: 18]
 - Phone (without any dashes/parenthesis) [eg: 8031234567]
 - Major
 - Year [eg: Freshman, Sophomore, Junior or Senior]
 - Today's Date (using the format: mm/dd/yyyy) [eg: 09/08/2024]
 - Include *at least 3 additional questions* of your choice. For example, musical instruments you play, your dream destination, your lucky number, student clubs, hobbies, volunteering etc. Be creative!
2. Compute the birth year of the user from the age provided using basic arithmetic operators
3. Convert the date entered by the user into European format of dd.mm.yyyy
4. Modify the phone number entered by the user using the format (xxx)xxx-xxxx [enclose area code in parentheses and hyphenate the 3 digits after]
5. Display an interesting intro about the user on the console (see the Hint section for more details)

Requirements:

- Functionality (85 pts)
 - o No Syntax, Major Run-Time, or Major Logic Errors. (85pts*)

- o Code that cannot be compiled due to syntax errors is nonfunctional code and will receive no points for this entire section.
 - Code that cannot be executed or tested due to major run-time or logic errors is nonfunctional code and will receive no points for this entire section.
- o Clear and Easy-To-Use Interface (5 pts)
 - Users should easily understand what the program does and how to use it.
 - Users should be prompted for input and should be able to enter data easily.
 - Users should be presented with output after major functions, operations, or calculations.
 - All the above must apply for full credit.
- o Users must be able to enter input values (15 pts)
 - No error checking is needed here and you may assume that users will only enter valid values
- o Store, modify operations and display user provided input (65 pts)
 - See the Instructions section for more details
- o *You may NOT use methods such as `replace()`, `parseInt()`, `getNumericValue()`, `digit()`, regular expressions or classes such as `DecimalFormat`, `DateFormat`, `SimpleDateFormat`, `Character` or any automatic converter (80pts*).
 - *The use of built-in methods or parser classes like `replace()`, `parseInt()`, `getNumericValue()`, `digit()`, `DateFormat`, `SimpleDateFormat`, `Character`, `regex` or any automatic converter will result in a 0 for this entire section.
 - You may use other String methods
- Coding Style (5 pts)
 - o Readable Code
 - Meaningful identifiers for data and methods.

- Proper indentation that clearly identifies statements within the body of a class, a method, a branching statement, a loop statement, etc.
- All the above must apply for full credit.
- Comments (10 pts)
 - o Your name at the beginning of the file as a single-line comment. (5 pts)
 - o At least 5 meaningful comments in addition to your name. These must describe the function of the code it is near. (5 pts)

Hint:

Here's an quick intro for <first name> <last name>:

"You were born in the year <birth year>. In your free time, you love to watch <sport> and catch up on movies, with '<favorite movie>' being one of your favorites. Your lucky number is <lucky number>. You are a <year> at USC majoring in <major name>. You can be reached at <formatted phone number>. You are now officially a Gamecock!"

Last Updated: <date in European format>

Example:

Here's an quick intro for Nicholas Brown:

"You were born in the year 2006. In your free time, you love to watch football and catch up on movies, with 'Avengers End Game' being one of your favorites. Your lucky number is 9. You are a Freshman at USC majoring in Computer Science. You can be reached at (803)123-4567. You are now officially a Gamecock!"

Last Updated: 08.09.2024

Submission:

- Submit the `.java` file on Dropbox