User Documentation for NerdScore

The program is designed to calculate the nerd extent of various users. A formula built in NerdScore will calculate a score based on three scores. After a list of score is obtained to

1. **Menu**
2. General Use

The menu shows a welcome title for users when the app is run. A user will be required to choose from Fandom Score, Hobbies Score and Sports score.

For Hobbies Score and Sports Score, the input is required to be integers greater than zero. Any input inconsistent with the requirement will be reported and require a new input until correct values are put in.

The Hobbies Score is also required to be positive integers. Furthermore, this input is required to be a multiple of 4. Therefore, 0 is acceptable in this term.

There is also a loop to check if all three inputs are selected.

A standard output is shown below in Figure 1.1.

A screenshot of a social media post

Description automatically generatedFigure 1.1 Standard output of Menu

1. Common Mistakes
2. Inappropriate input

One of the common mistakes is a negative input or 0 in Fandom Score or Sports Score. For instance, if -1 is entered for Fandom Score, there will be a reminder and ask user to get a new input. The resulting output will be shown in Figure 1.2.a

A screenshot of a cell phone

Description automatically generatedFigure 1.2a Resulting output for negative output

Another possible case is the input for Hobbies Score is not input as a multiple of 4. In this situation, user will receive similar output but specify on new criteria, which is shown in 1.2b.

A screenshot of a social media post

Description automatically generated

Figure 1.2a Resulting output for an input not a multiple of 4 in Hobbies Score

1. Repeating input in one score

If a user forgets to give input to one of the three scores, a line will appear as a reminder. For example, if one type in Fandom Scores for twice but forgot to input in Sports score, the error will be reported like shown in Figure 1.2b

A screenshot of a social media post

Description automatically generated

Figure 1.2b Resulting output for repeating input

1. **NerdScore**
2. General Use

This program is written to make calculation based on given input of three different scores. To simplify this process, scores are assigned to simpler variables, where x is assigned to be FandomScore, y is assigned HobbiesScore, and z is assigned to be SportsNum.

All the variables are examined to be positive integers. And for Hobbies Score, the variable requires user input to be a multiple of 4, including 0. Result of inappropriate input will be talked in (2) Common Mistakes below.

Then the variables are proceeded into a complex formula Skill Equation to calculate a skill score. Skill Equation is shown in Figure 2.1 below

A picture containing object

Description automatically generated Figure2.1 Formula of Skill Equation

Finally, the numeric output of Skill Equation will be printed to users. Figure 2.2 is an example output with input of FandomScore= 1, HobbiesScore= 2, and SportsNum=3.

A screenshot of a cell phone

Description automatically generated Figure2.2 Sample output of Skill Equation

1. Common Mistakes
2. Negative input.

A set of if loops are set to check the validation of inputs. Any negative input will be reported error. And the user will be prompted to enter a positive integer with a new input for user provided below.

For example, if a Fandom Score is changed to -1, the user will receive a printed message like shown in Figure 2.2a.

A screenshot of a cell phone

Description automatically generatedFigure 2.2a Example of negative input and corresponding output

1. Float input

If a float is entered instead of an integer, the if loop can also filter the wrong input with isinstance. Once such error is detected, a line will be printed to the user. The line will tell the user which number is wrong and prompt the user to enter a valid input.

Example is shown in Figure 2.2b below.

A screenshot of a social media post

Description automatically generatedFigure 2.2b Sample output if a float is used for input

If the SportsNum is changed to a float of 1.3, the output will show the wrong input type and restate the requirement for SportsNum.

1. **NerdClass**
2. General Use

When a list of NerdScore is provided to this section, it will firstly check the given data and report errors if input is not provided in correct type or range. After this process, the program will allocate the scores into corresponding Nerd classes based on the specific range of score.

For a score of 0, the user will be classified as “Nerdlite”. Any score between 1 (include 1) to 10 belongs to “Nerdling”. Score equal to or above 10 but below 100 is classified as “Nerdlinger”. Starting from 100 to 500 (500 is not included) is the class of “Nerd”. Score above 500 but below 1000 is a class called “Nerdlington”. Class from 1000 to 2000 (2000 is not included) “Nerdrometa”. Any score equal to 2000 or above is classified as Nerd Supreme.

Meanwhile, the program will count the total number of users in different classes. Finally, the total number of users in each class will be exported as a list like Figure 3.1 shown below.

A screenshot of a social media post

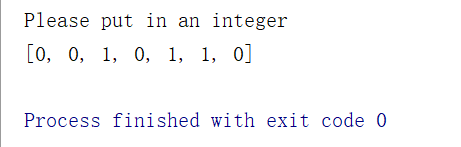
Description automatically generated

Figure 3.1 An example output of NerdClass

1. Common Mistakes

If any item of the list is out of range, for instance, a negative number exists in the input list, the program will respond with a line “Please put in an integer” and ignore that score when doing the classification and counting.

An example of mistake with input is shown in Figure 3.2 below. In this example, one of the inputs is changed to -1 while the others stay the same, which caused the reminder of wrong input. It can be also observed that one count is missing from the output list. The third number was 2 in Figure 3.1. With the changed input, however, the count of that class is revised to 1 in Figure 3.2.

Figure 3.2 Sample output for input with errors