Space Travel Simulator Programming Project [2020-21]

Important Information:

All measurements should use SI units.

- Distance is measured in Kilometers (Km).
- All distances should be the AVERAGE DISTANCE from the Earth.
- Time is measured using seconds [s].
- Speed is measured using kilometers per second [Km / s].

Travel Time = Distance of Travel / Speed of Travel

Content Rubric

- Each Version Number listed below should be submitted to GitHub as a separate commit. If you forget one or two it is not a major issue but if I see huge chunks of code submitted at once, especially close together in time, it indicates that you are copy-pasting and not actually writing it yourself.

- MAKE SURE TO PUT YOUR CLASS PERIOD AT THE START OF EACH SUMMARY IN GITHUB.

    - **For Example:**[3A] Version 0.25

- **Version 0.0**: [5 Points]

    - Include Line 1 Comment with: Program Name, Author Name, Time/Date, and Version Number
    - On Line 2 write the following code: import time
        - import should turn green-yellow depending on your color scheme in IDLE.
    - Skip two lines by pressing enter.

- **Version 0.1** [5 Points]

    - Include COMMENT with definition of vocabulary term **ALGORITHM**
    - Single line COMMENTS breaking down the algorithm for the Space Travel Simulator.
        - Print Instructions
        - Get user selection for object in space.
        - Determine distance to object (stored as variables).
        - Get user input for speed of travel (in Km / s).
        - Calculate TIME of trip (Distance / Speed)
        - Check IF trip is > 3 years (seconds)
        - If TIME > 3 years, print() warning message indicating mission will most likely fail.
        - If Time <= 3 years, print() warning message indicating mission should succeed.
        - Print a thank you / good bye message.
    - Skip two lines by pressing enter.

- **Version 0.15** [5 Points]

    - Use print() method to print a STRING that introduces the program and explains its usage.
    - Comment explaining that methods such as a print() are pre-built functions in Python.
    - Skip two lines by pressing enter.

- **Version 0.2** [5 Points]

- COMMENT line that says "DECLARING VARIABLES".
- COMMENT that says, "Variables store different types of data."
- COMMENT that says, "Most Common Data Types are: INTEGERS, FLOATS, STRINGS, and BOOLS."
- COMMENT that says, "Integers are positive or negative whole numbers, including zero. Abbreviated as int."
- COMMENT that says, "Floats (floating point) are positive or negative numbers that have a decimal. Abbreviated as float."
- COMMENT that says, "Strings are lines of text including letters and characters. Abbreviated as str."
- Comment that says, "Bools (Boolean) are True of False values. Abbreviated as bool."
- Skip two lines by pressing enter.

- **Version 0.25** [5 Points]

  - COMMENT that says, "NAMING VARIABLES".
  - COMMENT that says, "Variables should ALWAYS be descriptive. Should be able to identify what type of data is stored in the variable."
  - COMMENT that says, "Examples include: num_eggs, amnt_gas, high_score, or player_name."
  - COMMENT that says, "Variables in Python can start with _ or a letter but NOT a number."
  - COMMENT that says, "Variables can use camelCase style or snake_case style but use the SAME style in your code."
  - Skip two lines by pressing enter.

- **Version 0.3** [5 Points]

  - COMMENT that says "DECLARING and INITIALIZING VARIABLES".
  - COMMENT that says, "Declaring a variable means to tell Python the name of the variable."
  - COMMENT that says, "Generally variables will be declared on the first column of a line."
  - COMMENT that says, "INITIALIZING a variable means to assign it a starting value using the = symbol."
  - COMMENT that says, "In Python, the = symbols means to assign the variable on the left side the value of the statement or expression on the right side."
  - COMMENT that says, "my_score = 5x + 12"
  - COMMENT that says, "The line above DECLARES a variable called my_score and INITIALIZES it to the value 5x + 12"
  - Skip two lines by pressing enter.

- **Version 0.35** [5 Points]

  - DECLARE a variable named *million* and INITIALIZE it to 1000000.
  - DECLARE a variable named *billion* and INITALIZE it to 1000000000.
  - DECLARE a variable name *trillion* and INITALIZE it to 1000000000000.
  - Skip two lines by pressing enter.

- **Version 0.4** [5 Points]

  - COMMENT that says "Objects in our Solar System"

Space Travel Simulator Programming Project [2020-21]

- DECLARE variables for THREE objects found inside our Solar system, such as the Sun, other planets, moons, or the asteroid belt. Each variable should be on its own line.
- INITIALIZE each variable using the distance from the Earth in kilometers and multiplying it by *million*, *billion, or trillion* as necessary. See the next bullet for an example.
  - dist_sun = 150.01 * million
- Skip two lines by pressing enter.

- **Version 0.45** [5 Points]

  - COMMENT that says, "Objects Outside of the Solar System".
  - COMMENT that says, "Most objects outside of the Solar System are so far away they are measured in LIGHT YEARS."
  - DECLARE a variable named *light_year* and INITIALIZE it to 9.4607 * *trillion*.
    - light_year = 9.46073 * trillion
  - DECLARE variables for THREE objects found outside our Solar system such as other stars, nebulae, black holes, dust clouds, or super novas. Each variable should be on its own line.
  - INITIALIZE each variable with the distance from the Earth. Most likely all of these distances will be given in light years. See the next bullet for an example:
    - dist_alpha_centauri = 4.37 * light_year
  - Skip two lines by pressing enter.

- **Version 0.5** [5 Points]

  - COMMENT that says, "Ask the user what their name is."
  - DECLARE a variable called *user_*name and INITIALIZE it using the input() method to have the user input their name. Make sure to print instructions using the input() method.
  - Use the print() method to print a STRING that substitutes their user name into the STRING.
  - Use the time.sleep() method to pause for 3-5 seconds after print().
  - Skip two lines by pressing enter.

- **Version 0.6 [10 Points]**

  - Use multiple print() statements to print a "menu" on the screen that lists all of the objects the user can pick.
  - Include instructions that explain HOW to pick an object from the menu. Examples:
    - "Type the number of the object on the list and then press ENTER."
    - "Type the FIRST LETTER of the object on the list and press ENTER."
  - Use the time.sleep() method to pause for 3-5 seconds after print() displays the menu.
  - Skip two lines by pressing enter.

- **Version 0.65** [5 Points]

  - DECLARE a variable called *user_choice* and INITIALIZE it using the input() method.
    - Make sure to print instructions with the input() method.
    - If you are having the user enter a number, make sure to use int() to convert to an integer value.
    - If you are having the user enter a letter, make sure to use the .lower() method to make sure the letter is lowercase.
  - DECLARE a variable called *distance* and INITIALIZE it to 0.

- Skip two lines by pressing enter.

Version 0.7 continues on the next page.

- **Version 0.7 [10 Points]**
  - Use an if/elif/else statement to assign a value to *distance* based on the user's choice from the menu. Use print() method to print a STRING that shows the name of the object and the *distance* from Earth.
    - If they DO NOT pick an item from the menu (else:), use print() method to instruct them to restart the program and make a choice from the menu, then use the exit() method to close the program.
  - Use time.sleep() to pause 3-5 seconds after the if/elif/else statement executes.
  - Skip two lines by pressing enter.

- **Version 0.8 [5 Points]**
  - DECLARE a variable named *light_speed* and INITIALIZE it to the value of light speed as measured in Kilometers / second. (Use your Internet search skills to find this number. DO NOT ROUND IT, USE THE EXACT NUMBER.)
  - DECLARE a variable named *user_speed* and INITIALIZE it using the input() method.
    - Make sure to include instructions to enter the speed as a WHOLE NUMBER in Km / s.
  - Use an if/elif/else statement to check that the *user_speed* is LESS THAN OR EQUAL TO *light_speed.*
    - If it IS use print() to let the user know the speed is acceptable.
    - If it is NOT use input() to give them a chance to enter a correct speed.
      - **Check once more if the speed is acceptable.**
        - **If it IS, print() that it is and move on.**
        - **If it is NOT, print() that they need to restart and then exit().**
  - Use time.sleep() to pause for 3-5 seconds.
  - Skip two lines by pressing enter.

- **Version 0.9 [5 Points]**
  - DECLARE a variable named *trip_time* and INITIALIZE it to *distance / user_speed.*
    - Since our distance is measured in Km and speed is Km / s, after dividing the *trip_time* variable will be measured in seconds.
  - Use a print() method to print a STRING that contains the *trip_time* variable.
  - DECLARE a variable named *secs_per_year* and INITIALIZE it to 3.154e7
    - Python supports scientific notation, which you have normally seen as 3.154×10^7.
  - DECLARE a variable named *max_time* and INITIALIZE it to three times *seconds_per_year*.
  - Use time.sleep() to pause for 3-5 seconds.
  - Skip two lines by pressing enter.

- **Version 0.95 [5 Points]**
  - Use an if/else statement to check whether *trip_time* is LESS THAN OR EQUAL TO *max_time*.
    - If it IS, print a message that the mission can proceed safely.
    - If it IS NOT, print a message of warning that mission might faily.

- Use time.sleep() to pause for 3-5 seconds.
- Skip two lines by pressing enter.

[Continued on Next Page]

- **Version 1.0 [5 Points]**
  - Use a print() method to print a Thank You! Message on the screen.
  - Use time.sleep() to pause for 3-5 seconds.
  - Use the exit() method to end the program.

Process Rubric

- **Code must execute correctly without runtime errors. [10 Points]**
- File Name: lastname_firstname_space_travel [5 Points]
- File Format: Python (.py) [5 Points]
- **Due Date:** Submitted to GitHub by 10/09/2020 at 3:00PM. [25 Points]