

# CSE 1300 - Assignment 3 Summer 2024

## Description:

For this assignment you'll be writing a Python program to practice with Conditional and Repetition program statements along with the List and Set data structures. You should only need content from Modules 3, 4 & 5 for this assignment. You can use content from later modules, but it is not required.

You will need to use [Python's built in Random Library](#) to randomly determine some values.

Games are a fun way to learn programming concepts, so you are going to write a simple game:

The Oregon Trail was 2,170-mile route from Independence, Missouri, to Oregon City, Oregon, that was used by hundreds of thousands of American pioneers in the mid-1800s to emigrate west. The trail was arduous and snaked through Missouri and present-day Kansas, Nebraska, Wyoming, Idaho and finally into Oregon. ([History.com](#) and [Wikipedia](#)).

You are to create a simple game to replicate the trail and the dangers the travelers faced on their journey.

- 1) Your program must ask how many are in their party and ask the user to enter the names of the settlers. The names should be stored using an appropriate data structure/data type. There should be a minimum of two settlers and a maximum of 10. (*Hint: Which data structure allows you to add and remove items from it?*)
- 2) Your program should include at least five possible deaths that can occur on the trail to the settlers using an appropriate data structure/data type (*Hint: do the possible ways to die change during the game?*). See articles above for ideas.
- 3) Variable to track mileage that the party has traveled. Start at zero (0).
- 4) Variable to track food – ask party how much food they purchase at the start. *Do not inform the player of this information: average human consumes ~ 5 pounds of food per day so each day will require 5 \* number of settlers.*
- 5) Variable to track number of days taken to complete the trail. Start at one (1).

Program should display what day it is and ask the user they want to do that day (shown in **Bold** below), only the choices should be displayed (*note: do not display the travel distances, food usage or chance of a settler dying to the user*):

- 1) **Continue journey.** *Party travels 5 miles with a 10% chance of a random settler dying from a random cause of death and uses 5 pounds of food per settler for the day.*
- 2) **Increase speed of travel.** *Increases chances of settler death to 20% for this day only and increases distance traveled this day to 10 miles and uses 5 pounds of food per settler for the day.*

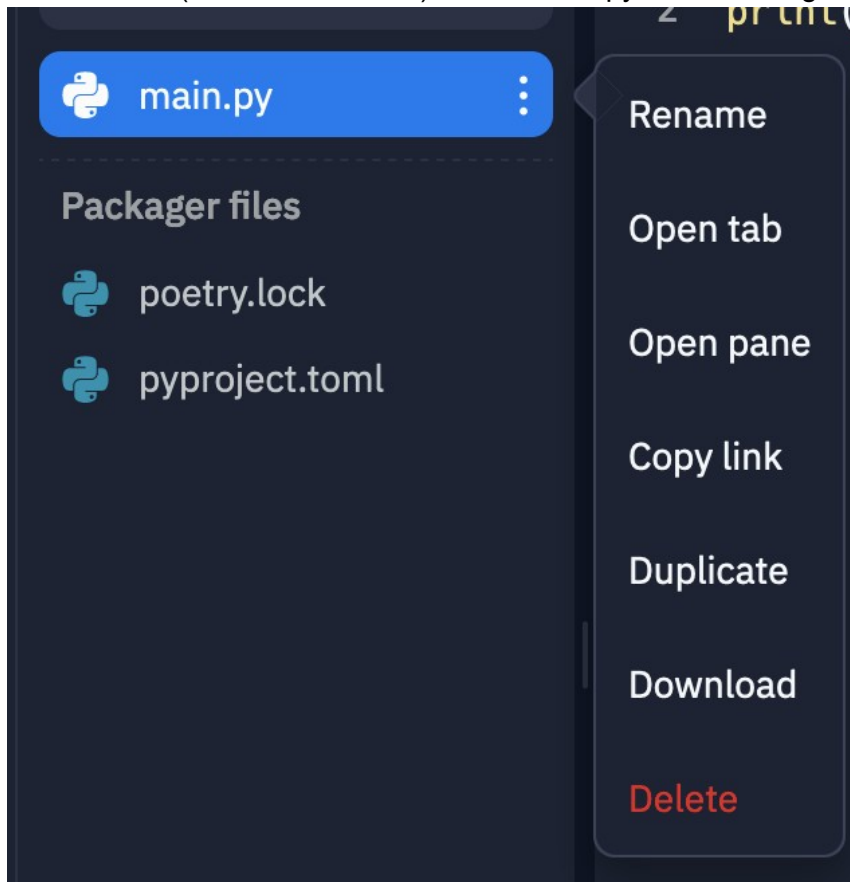
- 3) **Decrease speed of travel.** *Decreases chances of settler death to 1% for this day only and decreases distance traveled this day to 2.5 miles and uses 5 pounds of food per settler for the day.*
- 4) **Increase rations.** *Increases amount of food used per person to 7.5 pounds for this day and decreases the chance of accidental death to 5% for this day also with the party traveling 5 miles this day.*
- 5) **Decrease rations.** *Decreases amount of food used per person to 2.5 pounds for this day and increases the chance of accidental death to 15% for this day also with the party traveling 5 miles this day.*
- 6) **Abandon journey.** *Settlers give up and game ends.*

Use a **function/method** to print the menu of choices and **return** the chosen value to determine the outcome.

Game should end automatically if all party members die, no food is left, player abandons the journey, or the party travels the full distance. Once the game ends, display the remaining party member names, the distance traveled, how much food is left (if any) and how much further they had to travel to reach their destination.

## Submission Instructions:

1. All your code should be in a single Python (.py) file. The file name does not matter, the example below assumes you used Replit to create your program.
2. Once you are happy with your program, download your main.py by clicking the Menu button (three vertical dots) next to main.py and selecting Download:



3. Go to [Gradescope](#), find Assignment 3 under “Assignments”, and submit the main.py file.
4. After submitting, click Code in the upper right to verify that your file was uploaded correctly and that it is the correct Python file and file type (.py).



5. Be sure to submit before the deadline.
6. Do not submit multiple or zipped files, just submit the single Python (.py) file.