Homework 10 - OOP

CS 1301 - Intro to Computing - Fall 2022

Important

- Due Date: **Tuesday, November 22nd, 11:59 PM**.
- This is an individual assignment. High-level collaboration is encouraged, **but your submission must be uniquely yours.**
- Resources:
  - TA Helpdesk
  - Email TA's or use class Piazza
  - How to Think Like a Computer Scientist
  - CS 1301 YouTube Channel
- Comment out or delete all function calls. Only import statements, global variables, and comments are okay to be outside of your functions.
- **Read the entire document before starting this assignment.**

For this assignment, you will be building classes based on Thanksgiving. Thanksgiving is national holiday celebrated on the fourth Thursday of November each year in the United States where people across the country give thanks for their family, friends, and more. There will be three classes working together: a People class, a Food class, and an Activities class.

Each of these classes will have attributes and methods, as described below. You have been provided with a file that has the beginning of these classes. You are responsible for filling in the rest, and the methods you need will be clearly listed out in the grading rubric. **Please read the entire assignment before writing your code in order to understand how all of the classes interact with each other!**

Hidden Test Cases: In an effort to encourage debugging and writing robust code, we will be including hidden test cases on Gradescope for some functions. You will not be able to see the input or output to these cases. Below is an example output from a failed hidden test case:

```
Test failed: False is not true
```

Note: The given `HW10.py` has __repr__ methods, as well as some other useful methods that we have already defined for you. Do not delete or change these, as these are needed for testing.

Written by Paige Holland (pholland7@gatech.edu), Craig Nyazema (cnyazema3@gatech.edu), & Lorette Leone (lleone3@gatech.edu)

People

Attributes:

- name (`str`): the name of the person
- age (`int`): the age of the person
- isAsleep (`bool`): states whether the person is asleep or not
- canCook (`bool`): states whether or not the person can cook.

Methods:

- __init __
  - Initializes the following attributes:
    - name (`str`)
    - age (`int`)
    - isAsleep (`bool`)
    - canCook (`bool`) - should default to False.
    - friends(`list`) - should always start off as an empty list.
- wakeUp
  - This method does not take in any arguments.
  - In this method, check whether or not a person is asleep by referring to the isAsleep attribute. If they are, wake them up and change their isAsleep attribute to False.
  - If the person is already awake, return a string in the following format: `"{name} is already awake."`
- invite
  - This method takes in one additional argument; a People object.
  - In this method, add the object to the friends list.
- __str __
  - This method does not take in any additional arguments and creates a string representation of the `People` object.
  - The representation of the `People` object should include whether or not the person can cook and should be in the following format if they can: `"My name is {name} and I can cook."`

    - If they can't cook, the string representation should be in the following format: `"My name is {name} and I can't cook."`.

- __lt__
    - One `People` object is considered less than another if their age is less than the other's.

Food

Attributes:

- name (`str`): name of the food
- ingredients (`list`): list of tuples where the first element is the name of the ingredient and second element is the servings requried for that ingredient
- prepTime (`int`): time for meal preperation

Methods:

- __ init __
    - Intializes the following attributes:
        - name (`str`)
        - ingredients (`list`)
        - prepTime (`int`)
- __ str __
    - This method does not take in any additional arguments and creates a string representation of the People object.
    - The representation of the People object should be in the following format: `"{Food name} takes {Food prep time} to make."`

Activities

Attributes:

- ingredientsDict (`dict`): a dictionary of ingredients (`str`) mapped to their current quantity (`float`)

Methods:

- __ init __
    - Intializes the following attribute:

        - ingredientsDict (`dict`) - If no value is provided, the dictionary should be empty

- cook
    - In this method, a `Food` object is being cooked.
    - This method should take in one additional parameter:
        - food (`Food`)
    - This method should first check if there are enough ingredients to cook the food. If cooking the food results in a negative updated value in ingredientsDict, then we cannot cook it.
        - If we do have enough ingredients:
            - Update ingredientsDict by subtracting the servings required (derived from `Food` 's ingredients list) from the current value in the dictionary
            - Return `"We made {Food name}!"`
        - If we do not have enough ingredients:
            - Return `"We did not have enough ingredients to make {Food name}."`
- kidsTable
    - This method will determine the four youngest `People` objects that will be sitting at the Kids Table.
    - This method should take in one additional parameter:
        - guests (`list`) -> this list only contains `People` objects
    - In this method, you should compare each of the `People` objects in the guest list based on their ages.
    - Return a list of the names of the four youngest guests (`list`).
- buyIngredients
    - In this method, an ingredient will be added and/or updated in ingredientsDict.
    - This method takes in two additional parameters:
        - ingredientName (`str`)
        - ingredientAmount (`int`)
    - If the given ingredientName is not already in the dictionary, then create a new key-value pair. Otherwise, just update the current quantity by adding ingredientAmount to the existing value.

Grading Rubric

Class: Method	Points
People: __ init __	6
People: wakeUp	12
People: invite	12
People: __ str __	6
People: __ lt __	6
Food: __ init __	6
Food: __ str __	6
Activities: __ init __	6
Activities: cook	14
Activities: kidsTable	14
Activities: buyIngredients	12
Total	**100**

Provided

The `HW10.py` skeleton file has been provided to you. This is the file you will edit and implement. All instructions for what the functions should do are in this skeleton and this document.

Submission Process

For this homework, we will be using Gradescope for submissions and automatic grading. When you submit your `HW10.py` file to the appropriate assignment on Gradescope, the autograder will run automatically. The grade you see on Gradescope will be the grade you get, unless your grading TA sees signs of you trying to defeat the system in your code. You can re-submit this assignment an unlimited number of times until the deadline; just click the “Resubmit” button at the lower right-hand corner of Gradescope. You do not need to submit your `HW10.py` on Canvas.