CSE101

DUE APR. 03

23:59 KST

ASSIGNMENT I

PROFESSOR FRANCOIS RAMEAU

ALGEBRAIC OPERATIONS & FUNCTIONS

General instructions

- Please add your name and email in the dedicated location at the top of your code
- Do not use any external library, except when explicitly requested
- Try to follow the naming convention proposed by PEP-8 seen in class
- Use meaningful names for your variables and functions
- If you face a problem submitting your code via GitHub, please contact the professor and the TA by email
- Note that the received code will be tested on a classifier to detect potential usage of Large Language Model. We will also pay particular attention to plagiarism
- Leave comments in your code to explain your code and describe the difficulties you faced

INVITATION LINK

https://classroom.github.com/a/GKs_rjSP

Exercise 1: Library Book Management (4 points)

This summer, you're working in a library and notice that its borrowing system isn't automated. This leads to numerous issues and a high rate of unreturned books. Thus, you decide to draft some Python code to automatically calculate the fee a person will have to pay if they return their books late.

1. Requesting and Validating User Input

Create a function `request_dela_info` that asks for and returns the number of days a book has been borrowed. If the number of days entered is negative, display an error message to the user (use exception). Explore how the function can return a boolean indicating if the input is valid, alongside the number of days. Check in the lecture how it is possible to return two results from a function.

2. Calculating Late Fees

Implement a function `calculate_late_fees` that computes the late fee for a book based on the number of days it's overdue. The fee structure is simple: There is no late fee for the first five days. From day 6 to day 10, the late fee is $0.50 per day. For any day beyond the 10th, the late fee increases to $1 per day. The function should return the total late fee and also display this amount to the user.

Test this function using some manual input provided through the function `request_delay_info.`

3. Challenge your Code

It is time to test your code; luckily, someone returns a book the same day! However, the overdue is at least 119 years (see the news). For this special case, create a function `convert_years_to_days`, which will input the number of years and output the corresponding number of days.
In the main part of your code, use this function to convert a hypothetical overdue period expressed in years into days, and then apply your `calculate_late_fees` function to calculate the fee for this period.

Recap

1.	Create a function `request_delay_info`	1 point
2.	Create a function `calculate_late_fees`	1 point
3.	Create a function `convert_years_to_days`	1 point
4.	Call your functions in the main part of the code	1 point

Exercise 2: Area of the Entertainment Park (6 points)

After your experience at the library, you found yet another summer job at an entertainment park. You are in charge of hiring employees to maintain the infrastructure. It has been evaluated that the park necessitates one employee per 1000 square meters.

1. Calculate Area of a Circular Section

This park has the peculiar shape of a circle with a radius of 100 meters, to estimate the number of employees to hire, you first need to estimate the total area. Write a function `calculate_circle_area` that calculates the area of the park's circular section, given its radius. This function returns the calculated area.

Just as a reminder, the area A of a circle is defined by :

$$A = \pi r,$$

With r the radius of the circle expressed in meters.

Remember that you can import the math module to use an accurate approximation of π!

2. Estimating Maintenance Crew Size

Create a function `estimate_crew_size` that estimates the number of maintenance crew members needed based on the area. The park requires one crew member per 1000 square meters, rounded to the nearest whole number. Return and display the crew size (number of employees needed).

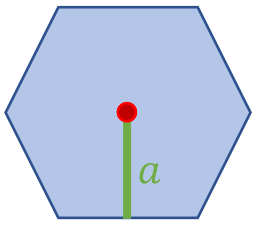
3. Area of a polygon

All of a sudden, the park's shape is changing due to a restructuring! It is not a circle anymore but a regular polygon formed by six sides of length 150m. A regular polygon is a geometric figure with all sides of equal length and all angles of equal measure.

While you are very familiar with calculating the area of a circle. The general formula to estimate the area A of a regular polygon is slightly more complex:

$$A = \frac{n \times s \times a}{2},$$

Where n is the number of sides, s is the length of each side, and a is the **apothem.** The apothem is the distance from the center of the polygon to the midpoint of a side.



The formula to calculate the apothem is the following:

$$a = \frac{s}{2\tan\left(\frac{\pi}{n}\right)}.$$

Write a function `calculate_poly_area` that calculates the park's area, given its number of sides, n, and their length equation. This function will necessitate the call of another function named `compute_apothem`, which returns the apothem a given n and s.

From the new estimated area, compute the needed number of crew members again.

Recap

1. Create a function `calculate_circle_area`	1 point
2. Create a function `estimate_crew_size`	1 point
3. Create a function `calculate_poly_area`	1 point
4. Create a function `compute_apothem`	1 point
5. Call these functions in the main part of the code	2 points