Python Assignment 2

1. Loops

Questions

1. Write a program that prints all even numbers from 1 to 100 using a for loop.
2. Using a while loop, ask the user to enter numbers until they type 0. Print the sum of all entered numbers (excluding 0).
3. Print the multiplication table for a number n (from 1 to 10), where n is input by the user.

2. Functions

Questions

1. Write a function `is_prime(n)` that returns True if n is prime and False otherwise. Use it to print all primes from 2 to 100.
2. Write a function `count_occurrences(items, target)` that returns how many times target appears in the list items.
3. Write a function `normalize_scores(scores)` that takes a list of numbers and returns a new list scaled to 0–100 (divide by max and multiply by 100). Handle an empty list by returning an empty list.

3. Map

Questions

1. Given a list of names like `["ali", "Sara", "omid"]`, use map to return a new list with each name capitalized (first letter uppercase, rest lowercase).
2. Given a list of Celsius temperatures, use map to convert them to Fahrenheit using the formula `F = C * 9/5 + 32`.

3. Use `map` to transform a list of strings `["1","2","3","-4"]` into integers. Filter out any values that cannot be converted (hint: use a helper function that returns None on failure and then remove None later).

4. Filter

Questions

1. Given a list of integers, use filter to keep only the positive numbers.
2. From a list of words, use filter to keep only those with length ≥ 5.
3. From a list of email strings, use filter to keep only those that contain exactly one '@' and at least one dot after the '@'.

5. Lambda Functions

Questions

1. Sort a list of tuples [(name, age)] by age using sorted with a lambda key.
2. Given prices = [120, 55, 300, 90], use a lambda with map to apply a 10% discount to all prices and return the new list.
3. Given a list of strings, sort them by their last character using a lambda with sorted.

6. Reduce

Questions

1. Using `functools.reduce`, compute the product of all numbers in a list (e.g., [2,3,4] -> 24).
2. Using reduce, find the longest string in a list of strings.
3. Using reduce, concatenate a list of words into a sentence separated by spaces (avoid a leading/trailing space).

7. Try/Except Exception

Questions

1. Write a function `safe_divide_list(nums, d)` that divides each number in nums by d. Use a loop and try/except to handle division by zero and non-numeric values; return a new list where invalid divisions are replaced with None.
2. Write a function `read_int_until_valid(prompt)` that loops asking the user for an integer and uses try/except to keep asking until a valid integer is entered. Return the integer.
3. Implement `sum_valid_numbers(text_values)` that loops over a list like `["10", "abc", "3.5", "7"]` and sums only the valid numeric entries (ints or floats). Use try/except.

8. Files in Python

Questions

1. Ask the user for a filename, then append a new line containing the current date and time to that file. If the file doesn't exist, create it.
2. Given a text file of numbers (one per line), read the file and compute the average. Handle empty files and invalid lines gracefully.
3. Write a program that copies the contents of input.txt to output.txt, but only lines that are not empty and don't start with #.

9. Classes & Objects

Problem

Create a class `BankAccount` with attributes owner and balance (default 0). Implement methods:

- `deposit(amount)` (adds to balance; reject negative amounts),
- `withdraw(amount)` (subtracts if sufficient funds; otherwise print an error),

- `__str__` to display "owner: balance".

Write a short script that creates two accounts, performs a few operations, and prints results.

10. Encapsulation

Problem

Refactor BankAccount to make balance private (name-mangled), and provide:

- `get_balance()` read-only accessor,
- `validation` inside deposit/withdraw,
- a property `is_overdrawn` returning True when balance < 0 (should never happen if validations are correct; include a unit-style check).

11. Inheritance

Problem

- Create a base class Shape with method `area()` that raises `NotImplementedError`.
- Implement `Rectangle(width, height)` and `Circle(radius)` subclasses that override `area()`.
- Create a list of mixed shapes and print each area (rounded to 2 decimals).

12. Polymorphism

Problem

- Define an interface-like base class `Notifier` with method send(message).
- Implement `EmailNotifier(address)` and `SMSNotifier(number)` that both implement send.
- Write a function `broadcast(notifiers, message)` that calls send on any Notifier passed in. Demonstrate with at least two different notifiers.