

Exercises on Basic Python, Object-Oriented Programming, and Testing

Basic Python (Chapter 2)

1. Compute the result and explain how Python evaluates the expressions below:

- `5 + 3 * 2`
- `10 / 3`
- `10 // 3`
- `10 % 3`
- `5 ** 2`

2. What will be the output of the following Python program? Explain why.

```
x = 3.5
y = int(x)
print(y, type(y))
```

3. Lists:

- (a) Create a list with the first five prime numbers.
- (b) Write a function to remove all even numbers from a given list.
- (c) Implement a function that finds the second largest number in a list.
- (d) Write a function that flattens a nested list.
- (e) Write a function that rotates a list to the right by a given number of positions.
Input: `[1,2,3,4,5]`, Rotate by 2 Output: `[4,5,1,2,3]`
Analyze its time complexity.
- (f) Write a function to count the occurrences of each element in a list.
Input: `[1,1,2,3,3,3,4]` Output: `{1:2, 2:1, 3:3, 4:1}`
Analyze its efficiency.

4. Tuples:

- (a) Convert a list of tuples into a dictionary.
- (b) Write a function that swaps the first and last elements of a tuple.
- (c) Write a function that finds the maximum and minimum values in a tuple.
Input: `(4, 7, 1, 9)` Output: `(9,1)`
Discuss time complexity.
- (d) Write a function that converts a tuple of numbers into a single concatenated string.
Input: `(1,2,3,4)` Output: `"1234"`
Analyze its efficiency.

5. Dictionaries:

- (a) Create a dictionary that maps three cities to their country.
- (b) Write a function that merges two dictionaries, summing values of common keys.

- (c) Write a function that inverts a dictionary (keys become values and vice versa).
 Input: `{ 'a': 1, 'b': 2 }` Output: `{ 1: 'a', 2: 'b' }`
 Analyze time complexity.
- (d) Write a function to find the most frequently occurring value in a dictionary.
 Input: `{ 'a': 3, 'b': 2, 'c': 3 }` Output: `3`
 Analyze performance.
6. Sets:
- (a) Write a function that returns the union, intersection, and difference of two sets.
- (b) Write a function that finds the symmetric difference between two sets.
 Input: `{1,2,3}, {2,3,4}` Output: `{1,4}`
 Provide an asymptotic analysis.
- (c) Write a function to check if two sets are disjoint.
 Input: `{1,2,3}, {4,5,6}` Output: `True`
 Discuss its computational complexity.

Object-Oriented Programming (Chapter 3)

- Define a class `Rectangle` in Python with:
 - Attributes: `length` and `width`.
 - A method `area()` that returns the area of the rectangle.
 - A method `perimeter()` that returns the perimeter of the rectangle.
- Define a class `Circle` with an attribute `radius`. Include methods:
 - `area()` that calculates the area.
 - `circumference()` that calculates the circumference.

Example: `Circle(5).area()` returns approximately 78.54.
- Define a class `Student` with attributes `name`, `age`, and `grades`. Include methods:
 - `average()` returning the average of grades.
 - `is_passing()` returning `True` if the average grade is above a threshold (e.g., 60).

Example: `Student("Alice", 20, [80, 90]).average()` returns 85.
- Define a class `BankAccount` with attributes `balance` and methods `deposit()` and `withdraw()`.
Example: After depositing 50 into an account initialized with 100, the balance is 150.
- Implement inheritance by defining a superclass `Vehicle` with attributes `make` and `model`. Create subclasses `Car` and `Bike` with additional attributes `doors` for `Car` and `type` for `Bike`.
Example: `Car("Ford", "Mustang", 4)` creates a car object.

Testing (Chapter 4)

- Write a Python function `is_positive(n)` that returns `True` if `n` is positive and `False` otherwise. Use an `assert` statement to test your function.
Example: Input: 5, Output: `True`; Input: -3, Output: `False`
- Write unit tests for the `Rectangle` class using Python's `unittest` framework.
- Explain the concept of Test-Driven Development (TDD) and illustrate it by writing tests first for a simple function that calculates the factorial of a number.
- Write tests that specifically check edge cases, incorrect usage, and error handling for a function that divides two numbers.
- Explain why tests should be maintained even after they pass successfully.