Python L2 Assignments

Python Developer – Level 2 Skills Test

Duration: 90 minutes
Instructions:

- Use Python 3.x
- You may use only the Python Standard Library (unless stated otherwise).
- Write clean, readable code with comments where helpful.
- Show sample runs with inputs and outputs for each problem.
- Handle edge cases where applicable.

Q1. String Manipulation & Data Parsing

You are given a text file where each line contains a name and a score separated by a comma, for example:

```
Alice, 90
Bob, 78
Charlie, 85
```

Task:
Write a function that reads the file and returns a dictionary where keys are names (as strings) and values are scores (as integers).

Example:

```
# Expected:
{'Alice': 90, 'Bob': 78, 'Charlie': 85}
```

Q2. Dictionary & List Comprehensions

You are given a list of dictionaries:

```
data = [
    {"name": "John", "age": 25},
    {"name": "Jane", "age": 30},
    {"name": "Bob", "age": 22}
]
```

Task:
Create a list of names of all people older than 24 using **list comprehension**.

Expected Output:

```
['John', 'Jane']
```

Q3. Error Handling

Write a function safe_divide(a, b) that:

- Returns the division of a / b if possible.
- If b is zero, return the string "Infinity".
- If either input is not a number, return the string "Invalid input".

Example:

```
safe_divide(10, 2)   # 5.0
safe_divide(10, 0)   # "Infinity"
```

```
safe_divide(10, "x") # "Invalid input"
```

Q4. Debugging Logic

You are given a list of integers. Write a function that returns a list containing **only the even numbers** from the list.
Example:

```
get_even_numbers([1, 2, 3, 4, 5])
# Expected: [2, 4]
```

Q5. File I/O & Data Aggregation

You have a CSV file in the format:

```
date,category,amount
2025-01-01,Food,10
2025-01-02,Transport,5
2025-01-03,Food,15
```

Task:
Read the CSV and return a dictionary showing **total amount spent per category**.

Expected Output:

```
{'Food': 25.0, 'Transport': 5.0}
```

Q6. Working with Dates

Write a function that takes a date string in YYYY-MM-DD format and returns the **day of the week**.

Example:

```
get_day_of_week("2025-08-13")
# Expected: "Wednesday"
```

Q7. API Request & JSON Processing

Use the free API endpoint:
https://jsonplaceholder.typicode.com/posts

Task:
Fetch all posts and return a list of titles where userId is 1.

Example Output (first 3):

```
['Title 1', 'Title 2', 'Title 3', ...]
```

Q8. Recursion

Write a recursive function factorial(n) that returns the factorial of a number n.

- If n is negative, raise a ValueError.
- Factorial of 0 or 1 should return 1.

Example:

```
factorial(5)  # Expected: 120
```

Q9. Class Implementation

Create a BankAccount class with:

- Attributes: owner, balance (default 0)
- Methods: deposit(amount), withdraw(amount), __str__()
- Withdrawal should return "Insufficient funds" if the balance is not enough.

Example:

```
acc = BankAccount("Alice", 100)
acc.deposit(50)
acc.withdraw(30)
print(acc)
# Expected: Owner: Alice, Balance: 120
```

Q10. Algorithm & Optimization

Write a function second_largest(nums) that returns the **second largest unique number** in the list.

- Do not use sorted() or max() more than once.
- The solution should run in a single pass.

Example:

```
second_largest([10, 20, 5, 8, 20])  # Expected: 10
```