Lab 1

Total Points: 100

Criteria	Exceeds Expectations ✅ *(Full Points)*	Meets Expectations ◯ *(Partial Points)*	Needs Improvement ⚠ *(Minimal Points)*	Points
Part 1: Draw a Diamond *(30 pts)*	Diamond is perfectly symmetric, accepts only valid odd input, and handles errors gracefully.	Diamond prints mostly correctly but has minor alignment issues or missing input validation.	Diamond fails to display properly or crashes due to missing logic.	/30
Part 2: Txt analysis *(30 pts)*	Correctly counts **letters, words, and sentences** for any input, ignores non-letter characters, and formats output cleanly.	Counts are mostly correct but mishandle edge cases (e.g., punctuation, extra spaces).	Incorrect or missing counts, incomplete logic, or code crashes.	/30
Part 3: Caesar Cipher *(30 pts)*	Correctly **encrypts & decrypts** using shift values, handles wrap-around (a→z, z→a), accepts upper/lowercase, and validates inputs.	Works for basic cases but fails for some edge cases like negative shifts, capitalization, or invalid inputs.	Cipher logic missing or incorrect, program does not produce valid results.	/30
Code Quality & Best Practices *(10 pts)*	Code is clean, readable, and well-commented. Uses **meaningful variable names** and follows **PEP8** formatting.	Code mostly readable but lacks comments or uses inconsistent formatting.	Code is hard to follow, poorly formatted, or missing comments.	/10
Total				**/100**

Notes

- Allow **partial credit** for partially working solutions.
- If students write custom helper functions, give extra points **only if the implementation improves readability**.
- Deduct up to **5 points** for missing or broken input validation.
- Deduct **5–10 points** for not committing and syncing correctly in Codespaces