

SCALE FOR PROJECT PYTHON MODULE (/PROJECTS/PYTHON-MODULE-04)

You should evaluate 1 student in this team



Git repository

git@vogsphere-v2.42madrid.com:vogsphere/intra-uuid-f1abf1a9-78c

Introduction

- Remain polite, courteous, respectful and constructive throughout the evaluation process. The well-being of the community depends on it.

- Identify with the person (or the group) evaluated the eventual dysfunctions of the work. Take the time to discuss and debate the problems you have identified.

- You must consider that there might be some difference in how your peers might have understood the project's instructions and the scope of its functionalities. Always keep an open mind and grade him/her as honestly as possible. The pedagogy is valid only and only if peer-evaluation is conducted seriously.

Guidelines

- Only grade the work that is in the student or group's GiT repository.

- Double-check that the GiT repository belongs to the student or the group. Ensure that the work is for the relevant project and also check that "git clone" is used in an empty folder.

- Check carefully that no malicious aliases was used to fool you and make you evaluate something other than the content of the official repository.

- To avoid any surprises, carefully check that both the evaluating and the evaluated students have reviewed the possible scripts used to facilitate the grading.

- If the evaluating student has not completed that particular project yet, it is mandatory for this student to read the entire subject prior to starting the defence.

- Use the flags available on this scale to signal an empty repository, non-functioning program, a norm error, cheating etc. In these cases, the grading is over and the final grade is 0 (or -42 in case of cheating). However, with the exception of cheating, you are encouraged to continue to discuss your work (even if you have not finished it) in order to identify any issues that may have caused this failure and avoid repeating the same mistake in the future.



- Remember that for the duration of the defence, no segfault, no other unexpected, premature, uncontrolled or unexpected termination of the program, else the final grade is 0. Use the appropriate flag.

You should never have to edit any file except the configuration file if it exists. If you want to edit a file, take the time to explicit the reasons with the evaluated student and make sure both of you are okay with this.

- You must also verify the absence of memory leaks. Any memory allocated on the heap must

be properly freed before the end of execution.
You are allowed to use any of the different tools available on the computer, such as leaks, valgrind, or e_fence. In case of memory leaks, tick the appropriate flag.

Attachments

-  subject.pdf (<https://cdn.intra.42.fr/pdf/pdf/190953/en.subject.pdf>)
-  data-generator-tools.tar.gz (<https://cdn.intra.42.fr/document/document/43579/data-generator-to>)

Preliminaries

Basics

- Only grade the work that is in the learner's Git repository.
- Check that only the requested files are available in the git repository.
If not, the review stops here.
- Verify that the files are organized in the correct directory structure (ex0, ex1, ex2, ex3, ex4).

 Yes  No

Exercises

Exercise 0 - Ancient Text Recovery

Test the Ancient Text Recovery system:

- If needed, ask the learner to help you with the tests
- Create a test file with some content
- Run the program and verify it:
 - Opens the file using open()
 - Reads the entire content using read()
 - Displays the content correctly with print()
 - Closes the file properly using close()
- Test with different file contents
- Verify the program follows the basic file operation pattern: open, read, print, close
- Ask the learner to explain what happens if a file isn't closed properly

Answer "Yes" if all tests pass and the learner understands basic file operations, "No" if any test fails

 Yes  No

Exercise 1 - Archive Creation

Test the Archive Creation system:

- If needed, ask the learner to help you with the tests
- Run the program and verify it:
 - Opens a file for writing using open() with 'w' mode
 - Writes content to the file using write()
 - Closes the file properly using close()
 - Creates the file if it doesn't exist
- Check that the created file contains the expected content
- Test that the program overwrites existing files (if applicable)
- Ask the learner to explain the difference between 'r' and 'w' modes
- Verify the learner understands that 'w' mode can overwrite existing data

Answer "Yes" if all tests pass and the learner understands file writing, "No" if any test fails.

 Yes  No

Exercise 2 - Stream Management

Test the Stream Management system:

- If needed, ask the learner to help you with the tests
- Run the program and verify it:

Intra Projects Python Module 04 Edit

- Reads input from sys.stdin (or uses input())
- Writes normal output to sys.stdout (standard print())
- Writes alerts/errors to sys.stderr (using print(message, file=sys.stderr))
- Demonstrates understanding of the three streams
- Test input/output functionality
- Verify that different types of messages go to appropriate streams
- Ask the learner to explain why programs have separate streams for normal output and errors
- Check that the learner understands the purpose of each stream

Answer "Yes" if all tests pass and the learner understands stream management, "No" if any test fail

☒ Yes

☐ No

Exercise 3 - Vault Security

Test the Vault Security system:

- If needed, ask the learner to help you with the tests
- Verify the program uses the with statement for all file operations
- Test file reading with: with open(filename, 'r') as file:
- Test file writing with: with open(filename, 'w') as file:
- Verify that files are automatically closed when using with
- Create a scenario where an exception might occur and verify the file is still closed
- Ask the learner to explain how the with statement makes file operations safer
- Ask the learner to explain what RAII (Resource Acquisition Is Initialization) means
- Verify the learner understands context managers

Answer "Yes" if all tests pass and the learner understands safe file handling, "No" if any test fails.

☒ Yes

☐ No

Exercise 4 - Crisis Response

Test the Crisis Response system:

- If needed, ask the learner to help you with the tests
- Verify the program combines try/except blocks with the with statement
- Test FileNotFoundError handling:
 - Try to access a non-existent file
 - Verify the program catches the error gracefully
 - Check that an appropriate error message is displayed
- Test PermissionError handling (if possible):
 - Try to access a file without proper permissions
 - Verify the program handles the error without crashing
- Test successful file operations:
 - Verify normal file operations work correctly
 - Check that success cases are handled properly
- Ask the learner to explain the most common file operation errors
- Ask the learner to explain how proper error handling prevents data loss
- Verify the learner understands defensive programming principles

Answer "Yes" if all tests pass and the learner understands error handling, "No" if any test fails.

☒ Yes

☐ No

Knowledge Review

Knowledge Questions

Ask the learner to explain the following concepts and verify their understanding:

- "What happens to the storage system if connections aren't properly closed?" (from Exercise 1)
- "What's the critical difference between extraction mode ('r') and preservation mode ('w')?" (from Exercise 2)
- "Why do the Archives maintain separate channels for standard data and alerts?" (from Exercise 3)
- "How does the with protocol prevent data corruption? What is the RAII principle?" (from Exercise 4)
- "What are the most dangerous threats to digital archives?" (from Exercise 4)
- Ask them to demonstrate the basic file operation pattern: open, read/write, close
- Ask them to show how context managers work with the with statement

The learner should show solid understanding of file operations and data preservation concepts.

Answer "Yes" if explanations are satisfactory, "No" if lacking understanding.

Yes

No

Ratings

Don't forget to check the flag corresponding to the defense

Ok

★ Outstanding project

Empty work

Incomplete work

Invalid compilation

Norme

Cheat

▲ Concerning situation

Leaks

Forbidden function

Can't support / i

Conclusion

Leave a comment on this evaluation (2048 chars max)

Finish evaluation