CSE 3302: Programming Languages

Lab 01 – Recursive File Space

100 points

INSTRUCTIONS

1. **Do NOT plagiarize.**
2. **No group work. All work should be your own.**
3. **Do not discuss your work with other students in the class.**
4. **You CANNOT borrow code from online sources.**
5. **Turn in your program using Canvas. Do not email your program to the TA or the instructor.**
6. **Name your source file as netid_lab01.extension where netid is your UTA netid and the extension is whatever is appropriate for that language (ksf_lab01.c or ksf_lab01.py, etc). If you do not know your netid, check what it is by using NetID Self Service. Your 1000 number is NOT your netid. If your file name is wrong, your assignment will not be graded.**
7. **All code should be your own. You may not copy code from the slides, book, others, or the internet unless specified.**
8. **The programs will be tested against a directory with multiple levels of subdirectories.**
9. **Write an explanation of your code using comments. If the explanation is not clear, you will NOT receive full credit.**
10. **The code should have your name, 1000 number, lang ver, and OS used as the first 4 lines of the source.**
11. **Submit a single ZIP file containing all your source code files. The filename will by netid_lab01.zip where 'netid' is replaced by your netid just as in #6 above.**
12. **NOTE: Your code must include a recursive function written by the student; this means the library function 'os.walk()' is not allowed when using Python.**

Objective: Write the same program in 3 different languages.

Description:

Write a program to calculate the total size of all files in the current directory / folder and all sub-folders.
The code should be runnable on the Omega server(netid@omega.uta.edu)without any configuration.

Languages to choose from:

- C
- C++

- Java
- Python
- Perl
- Other (get GTA approval beforehand)

Answer the following questions in comments in one of the source files or in the submission text area on Canvas:

1) Was one language easier or faster to write the code for this? If so, describe in detail why, as in what about the language made that the case.
2) Even though a language may not (e.g. FORTRAN) does not support recursion, describe how you could write a program to produce the same results without using recursion. Would that approach have any limitations and if so, what would they be?

Note: Do NOT contact the TA or instructor regarding the test cases.